ISSN 0555-2656

БИОНИКА ИНТЕЛЛЕКТА

ИНФОРМАЦИЯ, ЯЗЫК, ИНТЕЛЛЕКТ

№ 3 (77) **2011**

НАУЧНО-ТЕХНИЧЕСКИЙ ЖУРНАЛ

Основан в октябре 1967 г.

Учредитель и издатель
Харьковский национальный университет радиоэлектроники

Периодичность издания – *3 раза в год*

Харьков • ХНУРЭ • 2011

СОДЕРЖАНИЕ

УДК 519.7

М.Ф. Бондаренко, Ю.П. Шабанов-Кушнаренко
ХНУРЭ, г. Харьков, Украина

ОБ АЛГЕБРЕ КОНЕЧНЫХ ПРЕДИКАТОВ

Рассмотрены проблемы построения эффективного математического языка описания структур и функций систем естественного интеллекта. В качестве формального языка, на котором можно было бы математически описывать структуры и функции естественного интеллекта, используется алгебра конечных предикатов.

ТЕОРИЯ ИНТЕЛЛЕКТА, АЛФАВИТНЫЙ ОПЕРАТОР, ПРЕДИКАТ, АЛГЕБРА КОНЕЧНЫХ ПРЕДИКАТОВ

Введение

Теория интеллекта обслуживает *технику интеллекта* (то есть компьютеризацию и информатизацию), является ее математическим и физическим фундаментом. В соответствии с этим она делится на *математику* и *физику интеллекта*. Иногда говорят о теории искусственного интеллекта. Нам представляется это не вполне оправданным. Хотя интеллект невозможен без материального носителя, однако он может быть реализован как на естественной (человек), так и на искусственной (машина) физической основе. Механизмы разума можно изучать с общих позиций в рамках единой теории интеллекта, не привязываясь при этом к конкретному способу материальной реализации интеллектуальной системы.

Компьютер – машина математическая, он может воспроизвести лишь те структуры и функции, которые описаны в строго формализованном виде. Чтобы иметь возможность получать такие описания, необходимы математический язык описания и неформализованный материал о строении и функциях систем естественного интеллекта, прежде всего интеллекта человеческого. Теория интеллекта имеет многовековую предысторию. Существенный вклад в нее внесли многие выдающиеся мыслители. Особо следует отметить вклад Пифагора, Парменида, Платона, Аристотеля, Декарта, Лейбница, Ньютона, Буля, Фреге, Гильберта и Рассела.

Цифровая вычислительная техника в настоящее время стала одним из главных рычагов дальнейшего научно-технического прогресса, основой автоматизации процессов управления экономикой, производственных процессов, проектно-конструкторских и научно-исследовательских работ, важным источником повышения производительности труда и роста благосостояния народа, необходимым звеном в системе обороны страны. Электронные цифровые вычислительные машины представляют собой универсальное средство переработки информации, с их помощью, в принципе, можно автоматизировать любые виды умственного и физического труда [1]. Однако огромные потенциальные возможности вычислительных машин фактически используются далеко не в полной мере. Многие виды работ пока не поддаются автоматизации, а это серьезно сдерживает рост производительности труда, темпы прогресса. В чем причина такого парадоксального положения? Очевидно, в том, что многие работы, успешно выполняемые людьми, пока не под силу цифровой вычислительной машине: слишком еще слаб ее интеллект.

Как же повысить уровень машинного интеллекта? Для облегчения решения этой проблемы имеет смысл попытаться получить подсказку у природы, то есть пойти по бионическому [2] пути и обратиться к изучению человеческого интеллекта: ведь он способен выполнять информационные работы, недоступные пока вычислительной машине. Можно изучать две стороны человеческого интеллекта: 1) материальную сторону интеллекта – мозг, нервную систему, организм человека; 2) деятельность интеллекта, его функции, выражающиеся в поведении и действиях человека. Научная область, нацеленная на разработку описаний структуры и функций человеческого интеллекта, которые можно было бы использовать в деле совершенствования цифровых вычислительных машин, называется *теорией интеллекта*. При таком определении теории интеллекта ее успехи будут оцениваться не столько тем, как далеко эта теория продвинулась в познании интеллекта человеческого, но, главным образом, тем, какой уровень совершенства интеллекта машинного она смогла обеспечить.

Какие структуры и функции человеческого интеллекта должна изучать теория интеллекта? Очевидно, те и только те, которые, в принципе, доступны интеллекту машинному. Машинный же интеллект, то есть цифровая вычислительная машина, может действовать только механически, он способен воспроизводить лишь детерминированные, дискретные и конечные информационные процессы. *Детерминированные процессы* – это процессы с однозначным исходом, в них отсутствует фактор случайности. *Дискретные процессы* – это процессы, в которых информация имеет вид отдельных порций или квантов – цифр, букв, слов,

формул и т.д., в них отсутствует фактор непрерывности. *Конечные процессы* – это такие процессы, в которых может участвовать лишь конечное число единиц информации, в них отсутствует фактор бесконечности. Таким образом, *теория интеллекта* представляет собой науку о математическом описании детерминированных, дискретных и конечных интеллектуальных процессов, воспроизводимых человеческим разумом, и структур, обеспечивающих реализацию таких процессов, которая ориентирована на совершенствование цифровой вычислительной техники и ее практическое использование.

1. Алфавитные операторы

Для того чтобы иметь возможность развивать теорию интеллекта, прежде всего необходимо располагать формальным языком, на котором можно было бы математически описывать интересующие нас структуры и функции человеческого интеллекта. В качестве такого языка мы используем *алгебру конечных предикатов*, описание которой начнем с введения понятия конечного алфавитного оператора. К этому понятию естественным образом приводит ознакомление с принципом действия цифровой вычислительной машины. Любая вычислительная система, будь то машина в целом или ее отдельный блок, представляет собой устройство, осуществляющее преобразование информации. Эта система имеет *вход*, через который в нее поступает информация, подлежащая обработке, и *выход*, через который выдается выходная информация, сформированная вычислительной системой в ответ на поступившую в нее входную информацию. Как входная, так и выходная информация имеет вид знаковой последовательности, называемой *словом*. Знаки, из которых составлено слово, называются *буквами*.

Любая вычислительная система подвержена следующим ограничениям: 1) алфавит букв, из которых строятся слова для любой конкретной вычислительной системы, всегда конечен; 2) длина слов, которые способна воспринимать и формировать эта система, ограничена некоторым конечным наперед заданным числом букв, определяемым конструкцией системы, ее быстродействием и сроком службы; 3) реакции вычислительной системы строго детерминированы: повторное предъявление входного слова всегда приводит к формированию системой того же самого выходного слова.

Принципиально важно то, что человеческий интеллект в функциональном отношении весьма сходен с цифровой вычислительной машиной и подвержен тем же ограничениям, что и любая вычислительная система. Так же, как и вычислительная машина, человеческий интеллект способен воспринимать, преобразовывать и формировать информацию. Информация, которой оперирует человеческий интеллект, имеет вид слов, в роли которых выступают тексты, звучащая речь, чувственные образы предметов. Роль входа информации у человека выполняют органы чувств, роль выхода информации – органы движения и речи. Эти органы обладают конечной чувствительностью и конечной разрешающей способностью, поэтому в каждый момент времени человек может воспринимать сигналы (буквы) только из конечного множества (алфавита). Органы чувств, движения и речи обладают конечной полосой пропускания [1], поэтому они могут передать к интеллекту и от него лишь конечное число букв в единицу времени. Срок человеческой жизни ограничен, поэтому ограничена длина слов, которые могут обрабатываться интеллектом человека. Весьма вероятно, что реакции человека, формируемые его органами движения и речи, всецело определяются внешними воздействиями на этого человека, имевшими место в течение его предшествующей жизни. К числу этих воздействий на человека следует относить не только информацию, поступившую через органы чувств, но также и генетическую информацию, переданную его родителями. Важно отметить, что генетическая информация дискретна и конечна.

Рассмотрим конечное множество $A=\{a_1, a_2,..., a_k\}$, называемое *алфавитом*. Элементы $a_1, a_2,..., a_k$ этого множества называются *буквами* алфавита A. Число букв k в алфавите может быть произвольным. Примерами алфавитов могут служить: 1) русский алфавит, состоящий из 33 букв а, б,..., ю, я; 2) множество {0, 1,..., 9} всех десятичных цифр; 3) множество всех различных символов, использованных в этой статье (русские, латинские и греческие буквы различных шрифтов, знаки препинания, интервал между словами, математические знаки и т. п.); 4) множество символов, имеющихся на клавиатуре какой-либо вычислительной машины.

Любая последовательность $\sigma_1\sigma_2...\sigma_m$ букв σ_1, σ_2,..., σ_m алфавита A называется *словом* в алфавите A. Число m букв в слове называется *длиной слова*. Длину слова ничем не ограничивают, поэтому m может быть любым натуральным числом. Заметим, что буквы в слове могут повторяться, так что на различных местах в слове могут встречаться одинаковые буквы. Последовательности, имеющие в своем составе только одну букву, и последовательности, не содержащие ни одной буквы, также считаются словами. В последнем случае говорят, что слово имеет *нулевую длину*. Такое слово называется *пустым* и обозначается нами символом *. Хотя введение пустого слова, быть может, выглядит неестественным, однако оно необходимо для логической завершенности понятия слова. Рассмотрим примеры слов: 1) русские слова *мальчик*, *лампа* и т. п. могут рассматриваться как слова в русском

алфавите (в только что определенном смысле); 2) десятичная запись любого натурального числа есть слово в алфавите {0, 1,..., 9}; 3) любое предложение этой статьи есть слово в алфавите, состоящим из всех различных символов, встречающихся в этой статье; 4) весь текст данной статьи можно считать словом в том же алфавите.

Пусть A – произвольно выбранный непустой алфавит. Рассмотрим множество M всех слов, составленных из букв алфавита A. Поскольку в этом множестве встречаются слова сколь угодно большой длины, то общее число слов в нем бесконечно. Формально множество M может быть представлено как объединение всех декартовых степеней алфавита A, таким образом, $M=A^0\cup A^1\cup A^2\cup...$. Любое подмножество [3] L множества M всех слов алфавита A называется *языком* над алфавитом A. Язык L может быть как *конечным*, так и *бесконечным*. Примерами языков могут служить: 1) множество всех русских слов, представляющее собой конечный язык над русским алфавитом; 2) множество десятичных кодов всех натуральных чисел, представляющее собой бесконечный язык над алфавитом десятичных цифр; 3) совокупность всевозможных предложений, которые можно построить из знаков, фигурирующих в этой статье. Поскольку длина предложений, используемых в естественном языке, ограничена, то в последнем случае мы имеем дело с конечным языком.

Пусть A – произвольно выбранный непустой алфавит, а M – множество всех слов этого алфавита. Любая функция $Y=FX$, отображающая слова X из множества M в слова Y множества M, называется *алфавитным оператором*, заданным в множестве M. Иными словами, алфавитным оператором называется всякое однозначное соответствие, которое сопоставляет словам какого-либо алфавита слова того же алфавита. Совокупность L_1 всех тех слов, для каждого из которых алфавитный оператор F ставит в соответствие некоторое вполне определенное слово, называется *областью определения* или *входным языком* алфавитного оператора F. Может случиться, что алфавитный оператор F некоторым словам множества M не ставит в соответствие никакого слова, так что входной язык не обязательно совпадает с множеством M. Множество L_2 всех слов, являющихся значениями алфавитного оператора F, называется его *областью значений* или *выходным языком*.

Слова входного языка называются *входными словами* алфавитного оператора, слова выходного языка – его *выходными словами*. Алфавитный оператор, входной язык которого совпадает с множеством M, называется *всюду определенным*; если же входной язык является лишь частью множества M, то алфавитный оператор называется *частичным*. Примером всюду определенного алфавитного оператора может служить оператор, заданный на множестве десятичных кодов всех натуральных чисел, который десятичному коду каждого натурального числа ставит в соответствие десятичный код квадрата этого числа. Примером частичного алфавитного оператора может служить оператор, заданный на том же множестве, который десятичному коду натурального числа ставит в соответствие десятичный код квадратного корня этого числа, но только при условии, что этот корень оказывается натуральным числом.

Описанное выше понятие алфавитного оператора неплохо подходит для того, чтобы служить средством математического описания деятельности интеллекта, то есть объективно наблюдаемых реакций интеллекта на внешние воздействия. Информация, воспринимаемая интеллектом, соответствует входным словам некоторого алфавита; информация формируемая интеллектом, соответствует выходным словам. Закономерности преобразования информации интеллектом, то есть *функции интеллекта*, соответствуют тем или иным алфавитным операторам, преобразующим входные слова в выходные. Задача математического описания функций интеллекта заключается в том, чтобы указать соответствующие этим функциям алфавитные операторы.

2. Конечные алфавитные операторы

И все же понятие алфавитного оператора обладает одним существенным недостатком: его задают на бесконечном множестве слов, содержащем слова сколь угодно большой длины. Это обстоятельство приводит к потенциальному присутствию бесконечности в понятии алфавитного оператора [4], что создает определенные неудобства при математическом описании функций интеллекта и практическом их применении. Этих неудобств можно избежать, заменив понятие алфавитного оператора близким ему понятием *конечного алфавитного оператора*. Отличие конечного алфавитного оператора от только что описанного алфавитного оператора состоит лишь в том, что он задается не на бесконечном множестве $A^0\cup A^1\cup A^2\cup...$ слов различной длины, а на конечном множестве A^m слов одинаковой длины m, составленных из букв алфавита A.

Понятие конечного алфавитного оператора очень удобно для того, чтобы служить средством математического описания функций интеллекта. Ограничение, наложенное на длину слов, не может служить препятствием для его использования в теории интеллекта, поскольку, как указано выше, интеллект человека подвержен такому же ограничению. В качестве числа m должно быть выбрано число, не меньшее максимальной длины слов, с которыми может оперировать изучаемый интел-

лект. Возникает, правда, затруднение, вызванное тем, что интеллект обычно оперирует словами различной длины, тогда как в понятии конечного алфавитного оператора фигурируют слова только одинаковой длины.

Затруднение это, однако, легко преодолевается за счет введения в алфавит A дополнительной буквы #, называемой *знаком пробела* или просто *пробелом*. Слово, имеющее длину, меньшую, чем m, при его математическом описании заменяется словом длины m, левая часть которого совпадает с исходным словом, а правая часть представляет собой последовательность пробелов (с тем же успехом можно было бы сделать и наоборот, поменяв местами левую и правую части). При чтении формальной записи слова знаки пробела, стоящие в слове правее всех остальных букв, не должны приниматься во внимание. Слово, составленное из одних пробелов, должно интерпретироваться как пустое слово. Например, если выбрано m=6, то слово *лист* будет формально представлено в виде слова *лист*##. Положение здесь такое же, как при машинной записи числовых кодов: длина всех кодов принята одинаковой, однако нули, стоящие в левой части кода, при его чтении во внимание не принимаются. Если бы мы определили понятие конечного алфавитного оператора таким образом, чтобы в его входной и выходной языки вошли и слова меньшей длины, чем m, то это существенно усложнило бы математический язык для записи таких операторов.

Множество M, на котором задан конечный алфавитный оператор, содержит $c=k^m$ слов, то есть ровно столько, сколько имеется всех различных m-разрядных k-ичных числовых кодов. Всего существует c^c различных конечных алфавитных операторов, заданных на множестве M. Для каждой изучаемой функции интеллекта, в принципе, можно выбрать число k букв, алфавит A и предельную длину слов m такие, что в множестве всевозможных конечных алфавитных операторов, заданных на A^m, всегда найдется такой конечный алфавитный оператор, который может быть принят в качестве адекватного математического описания этой функции интеллекта. Задача состоит в том, чтобы, сообразуясь с фактическими свойствами изучаемой функции интеллекта, суметь выделить этот оператор и записать его в виде формулы на некотором математическом языке. Реализуя полученную формулу средствами цифровой вычислительной техники, можно будет изученную функцию интеллекта затем искусственно воспроизвести.

3. Конечные предикаты

Для того чтобы иметь возможность математически описывать функции интеллекта, нам необходим формальный язык, на котором можно было бы вести такое описание. Формальный язык должен быть выбран с таким расчетом, чтобы на нем можно было в удобной форме записать любой конечный алфавитный оператор. Такой язык дает нам описываемая ниже алгебра конечных предикатов. Введем понятие *конечного предиката*. Пусть A – конечный алфавит, состоящий из k букв $a_1, a_2,..., a_k$, Σ – множество, состоящее из двух элементов, обозначаемых символами 0, 1 и называемых соответственно *ложью* и *истиной*. Переменную, заданную на множестве A, будем называть *буквенной*, переменную, заданную на множестве Σ, – *логической*. *Конечным n-местным предикатом* над алфавитом A называется любая функция $f(x_1, x_2,..., x_n)=t$ от n буквенных аргументов $x_1, x_2,..., x_n$, заданных на множестве A, и принимающая логические значения t. Иногда будем называть конечный предикат f *k-ичным*, подчеркивая тем самым, что его алфавит A состоит из k букв.

Любой конечный предикат f можно, в принципе, задать с помощью таблицы его значений. В этой таблице каждому набору значений аргументов (x_1, $x_2,..., x_n$) ставится в соответствие значение t предиката f. В виде примера таблицей 1 задан двуместный предикат $t=f(x_1, x_2)$, определенный над алфавитом A={а, м, п}. По таблице находим, что, например, на наборе (x_1, x_2) значений аргументов x_1, x_2, равном (а, м), предикат f принимает значение 0, на наборе (п, м) – значение 1. Аналогично находим значения предиката и для остальных наборов. Расставив в последней строке таблицы логические значения каким-либо иным способом, получим другой двуместный предикат над тем же алфавитом.

Таблица 1

x_1	а	а	а	м	м	м	п	п	п
x_2	а	м	п	а	м	п	а	м	п
t	0	0	1	1	1	0	0	1	1

Для упрощения дальнейшего изложения сейчас нам будет полезно познакомиться с некоторыми сведениями об *n-разрядных k-ичных числовых кодах*. При построении k-ичных кодов используют k знаков 0, 1,..., k-1, называемых *k-ичными цифрами*. Число k называется *основанием* k-ичной *системы счисления*. Цифры в k-ичном коде располагаются в определенном порядке, значение каждой цифры определяется ее положением в коде. Так, например, в десятичной системе счисления в записи 179 цифра 1 обозначает число 100 или $1\cdot10^2$, цифра 7 – число 70 или $7\cdot10^1$, цифра 9 – число 9 или $9\cdot10^0$. Сумма этих чисел дает значение кода:

$$179=1\cdot10^2+7\cdot10^1+9\cdot10^0.$$

В общем виде десятичный код $a_1a_2...a_i...a_{n-1}a_n$, где $a_1, a_2,..., a_i,..., a_{n-1}, a_n$ – десятичные цифры; n – число разрядов в коде; i – номер разряда кода, обозначает число:

$$a_1a_2...a_i...a_{n-1}a_n=a_1 10^{n-1}+a_2 10^{n-2}+...+a_i 10^{n-i}+...+$$
$$+a_{n-1}10^1+a_n10^0.$$

Как узнать, какое число представляет собой тот или иной k-ичный код? Для этого достаточно его перевести в привычный нам десятичный код. Этот перевод можно осуществить, воспользовавшись определением k-ичного кода. Пусть $b_1b_2...b_i...b_{n-1}b_n$ – k-ичный код, где b_1, b_2,..., b_i,..., b_{n-1}, b_n – какие-нибудь произвольно выбранные k-ичные цифры. По определению этот код представляет собой следующее число:

$$b_1b_2...b_i...b_{n-1}b_n=b_1k^{n-1}+b_2k^{n-2}+...$$
$$...+b_ik^{n-i}+...+b_{n-1}k^1+b_nk^0.$$

В качестве примера переведем двоичный код 10110011 в десятичный:

$$10110011_2=$$
$$=1\cdot2^7+0\cdot2^6+1\cdot2^5+1\cdot2^4+0\cdot2^3+0\cdot2^2+1\cdot2^1+1\cdot2^0=179_{10}.$$

При одновременном рассмотрении кодов с различными основаниями, во избежание путаницы, справа внизу у кода будем указывать его основание.

Рассмотрим теперь способ обратного перевода десятичного кода числа в k-ичный. Пусть задано некоторое число N в десятичном коде, которое мы хотим перевести в k-ичный код. Предположим, что $b_1b_2...b_n$ есть k-ичный код числа N. Тогда:

$$N=b_1k^{n-1}+b_2k^{n-2}+...+b_{n-1}k^1+b_nk^0.$$

Разделим число N на k:

$$\frac{N}{k}=b_1k^{n-2}+b_1k^{n-3}+...+b_{n-2}k^0+\frac{b_n}{k}.$$

Целая часть частного N_1 этого деления равна:

$$N_1=b_1k^{n-2}+b_2k^{n-3}+...+b_{n-2}k^1+b_{n-1}k^0.$$

В остатке получаем b_n. Итак, в результате деления числа N на k в остатке получаем последний разряд k-ичного кода этого числа. Выполним второе деление, разделив число N_1 на k. В целой части частного получаем число

$$N_2=b_1k^{n-3}+b_2k^{n-4}+...+b_{n-3}k^1+b_{n-2}k^0,$$

а в остатке – b_{n-1}, то есть предпоследний разряд k-ичного кода числа N. При третьем делении в остатке получаем b_{n-2} и т. д. При n-ном делении в остатке получаем b_1, то есть первый разряд k-ичного кода, а в целой части частного – нуль.

В итоге приходим к следующему правилу перевода десятичного кода в k-ичный. Для перевода десятичного кода в k-ичный код следует разделить его на k, полученную целую часть частного снова разделить на k и т. д. Деление продолжать до тех пор, пока в целой части частного получится нуль. Считывая остатки этих делений в обратном порядке, получаем k-ичный код числа. В качестве примера переведем число 169 в двоичный код. Выполняем последовательные деления заданного числа на 2, в скобках записываем остатки делений: 169:2=84 (1), 84:2=42 (0), 42:2=21 (0), 21:2=10 (1), 10:2=5 (0), 5:2=2 (1), 2:2=1 (0), 1:2=0 (1). Считывая остатки, получаем двоичный код числа: $169_{10}=10101001_2$.

Введем число $L(n, k)$ всех различных n-разрядных k-ичных кодов. Оно выражается формулой $L(n, k)=k^n$. Для примера подсчитаем число всех трехразрядных восьмиричных кодов: $L(3, 8)=8^3=512$. Наборы значений аргументов $(x_1, x_2,..., x_n)$ n-местных k-ичных предикатов удобно интерпретировать как n-разрядные k-ичные числовые коды. При этом буквы a_1, a_2,..., a_k алфавита A интерпретируем как k-ичные цифры соответственно 0, 1,..., k-1. В таблице значений предиката наборы значений аргументов будем располагать в порядке возрастания представляемых ими чисел. Именно в таком порядке расположены, к примеру, двухразрядные троичные коды в табл. 1. В этом случае буквы а, м, п интерпретируем соответственно как троичные цифры 0, 1, 2. Введем число $M(n, k)$ всех различных наборов значений аргументов n-местного k-ичного предиката. Очевидно, что $M(n, k)=L(n, k)=k^n$. Число столбцов в таблице значений предиката, в которых располагаются наборы значений аргументов, определяется величиной $M(n, k)$. Например, число столбцов табл. 1 равно $M(2, 3)=3^2=9$. Каждому набору значений конечного предиката присвоим свой номер, в качестве которого примем число, соответствующее этому набору при его интерпретации в виде числового кода.

Пронумеруем все k-ичные n-местные предикаты. С этой целью будем интерпретировать последовательность значений предиката, расположенную в нижней строке его таблицы, как kn-разрядный двоичный код. При этом символ 0 интерпретируем как цифру нуль, а символ 1 – как цифру один. Число, соответствующее этому коду, примем в качестве номера конечного предиката. К примеру, предикату, представленному табл. 1, присваиваем номер 001110011, что соответствует числу 115 в десятичной записи. Пусть $N(n, k)$ — число всех различных n-местных k-ичных предикатов. Очевидно, оно совпадает с числом всех k^n-разрядных двоичных кодов, поэтому $N(n, k)=L(k^n, 2)=2^{(k^n)}$. В виде примера определим число всех различных двуместных троичных предикатов: $N(2,3)=2^{(3^2)}=2^9=512$.

4. Применения конечных предикатов

Каждому конечному алфавитному оператору можно поставить в соответствие некоторый свой конечный предикат. Сделать это можно следующим образом. Пусть

$$F(x_1x_2...x_m)=y_1y_2...y_m$$

– произвольно выбранный конечный алфавитный оператор, преобразующий входные слова $x_1x_2...x_m$ длины m в выходные слова $y_1y_2...y_m$ той же длины, составленные из букв алфавита A. Построим

$2m$-местный конечный предикат $f(x_1, x_2,..., x_m, y_1, y_2,..., y_m)$ над алфавитом A, руководствуясь следующим правилом:

$$f(x_1, x_2,..., x_m, y_1, y_2,..., y_m) = \begin{cases} 1, \text{ если } F(x_1x_2...x_m) = y_1y_2...y_m, \\ 0, \text{ если } F(x_1x_2...x_m) \neq y_1y_2...y_m. \end{cases} \quad (1)$$

Запишем уравнение:

$$f(x_1, x_2,..., x_m, y_1, y_2,..., y_m)=1. \quad (2)$$

Это уравнение связывает переменные $x_1, x_2,..., x_m, y_1, y_2,..., y_m$ некоторым отношением [5]. Подставляя в (2) буквы $x_1, x_2,..., x_m$ входного слова $x_1x_2...x_m$ алфавитного оператора F, получим в результате решения этого уравнения буквы $y_1, y_2,..., y_m$ выходного слова $y_1y_2...y_m$. Таким образом, предикат f, построенный указанным способом, содержит в себе всю информацию об интересующем нас алфавитном операторе F. С его помощью можно определить выходное слово алфавитного оператора, представляемого этим предикатом, для любого входного слова.

Важно отметить, что точно таким же способом можно задать с помощью конечных предикатов не только всюду определенные, но также и любые частичные конечные алфавитные операторы. Если для входного слова $x_1x_2...x_m$ алфавитный оператор F не ставит в соответствие никакого выходного слова, это значит, что уравнение (2) для заданного набора значений аргументов $(x_1, x_2,..., x_m)$ не имеет ни одного решения относительно набора переменных $(y_1, y_2,..., y_m)$, то есть что решения этого уравнения не существует. Важно, что с помощью одного конечного предиката можно задать целое семейство конечных алфавитных операторов. Это достигается введением в предикат дополнительной переменной z, значениями которой служат номера задаваемых алфавитных операторов. Пусть, к примеру, нам нужно представить в виде конечного предиката семейство, состоящее из трех алфавитных операторов:

$$F_1(x_1x_2...x_m)=y_1y_2...y_m,$$
$$F_2(x_1x_2...x_m)=y_1y_2...y_m,$$
$$F_3(x_1x_2...x_m)=y_1y_2...y_m.$$

Строим $2m+1$-местный предикат $f(x_1, x_2,..., x_m, y_1, y_2,..., y_m, z)$, полагая, что при $z=1$ он соответствует алфавитному оператору F_1, при $z=2$ – оператору F_2, при $z=3$ – оператору F_3.

С помощью конечных предикатов можно представить не только интересующие нас конечные алфавитные операторы с однозначными значениями, но и так называемые многозначные операторы. *Многозначным конечным алфавитным оператором* называется такое соответствие F, которое связывает каждое входное слово из A^m с некоторым семейством входных слов из того же множества. Если входному слову $x_1x_2...x_m$ алфавитный оператор F ставит в соответствие несколько входных слов, значит, уравнение (2) для заданного набора значений аргументов $(x_1, x_2,..., x_m)$ имеет несколько решений относительно набора переменных $(y_1, y_2,..., y_m)$.

Многозначный оператор можно рассматривать как обобщение понятия однозначного конечного алфавитного оператора. *Однозначный конечный алфавитный оператор* – это такой многозначный конечный оператор, который каждому входному слову ставит в соответствие некоторое множество выходных слов, состоящее не более чем из одного слова. Если каждому входному слову ставится в соответствие множество, состоящее не менее чем из одного слова, то такой конечный алфавитный оператор называется *всюду определенным*, если же некоторым из входных слов ставится в соответствие пустое множество слов, то – *частичным*.

Поначалу, быть может, несколько обескураживает то, что конечные предикаты дают нам больше, чем мы от них ожидали. В самом деле, введя конечные предикаты, мы хотели получить средство представления только для однозначных конечных алфавитных операторов, фактически же получили в виде «бесплатного приложения» возможность представления и многозначных операторов. Хорошо это или плохо? Если бы оказалось, что многозначные алфавитные операторы бесполезны для теории интеллекта, то это, конечно, было бы плохо, так как в теории интеллекта появились бы неинтерпретируемые математические структуры. К счастью, однако, обнаруживается, что многозначные алфавитные операторы (а это покажет дальнейшее изложение) весьма удобны для математического описания высказываний или суждений интеллекта – важных объектов теории интеллекта. Однозначным же алфавитным операторам отводится роль средства математического описания деятельности интеллекта.

Читатель, вероятно, заметил, что при переходе от конечных алфавитных операторов к соответствующим им конечным предикатам мы одновременно совершили переход от переменных X и Y, значениями которых были слова, к переменным $x_1, x_2,..., x_m, y_1, y_2,..., y_m$, пробегающим буквенные значения. Переход к буквенным переменным очень важен, он дает нам в руки удобный математический язык для описания функций интеллекта. Важно понять, что буквенные переменные было бы невозможно использовать, если б мы в свое время не перешли к конечным алфавитным операторам, а продолжали базироваться на понятии алфавитного оператора, заданного на бесконечном множестве слов произвольной длины.

При попытке представления алфавитных операторов с помощью предикатов с буквенными

переменными нам пришлось бы столкнуться с необходимостью введения бесконечного числа буквенных переменных. Если б мы, уже после введения конечного множества слов, сохранили в данном множестве слова различной длины, то это также помешало бы прийти к буквенным переменным, так как потребовалось бы ввести предикаты от непостижимого переменного числа переменных. Конечные предикаты с буквенными переменными – это та награда, которую мы получаем в обмен на отказ от традиционного понятия алфавитного оператора, базирующегося на абстракции потенциальной бесконечности.

Рассмотрим пример представления конечного алфавитного оператора с помощью конечного предиката. Пусть нам дан оператор $FX=Y$, преобразующий двухбуквенные слова $X=x_1x_2$ русского алфавита в однобуквенные слова $Y=y_1$ того же алфавита. Вид преобразования указан в табл. 2. Этому алфавитному оператору ставим в соответствие предикат $t=f(x_1, x_2, y_1)$, описываемый табл. 3. В таблице указаны не все столбцы, а только те из них, у которых в последней строке стоит значение 1. Полагаем, что во всех отсутствующих столбцах предикат принимает значение 0.

Таблица 2

X	до	ре	ми	фа	ля	си
Y	а	о	у	и	е	я

Таблица 3

x_1	д	р	м	ф	л	с
x_2	о	е	и	а	я	и
y_1	а	о	у	и	е	я
t	1	1	1	1	1	1

Поскольку в русском алфавите 33 буквы, то общее число столбцов таблицы должно было бы составлять $33^3 \approx 30$ тысяч. В связи со столь резким увеличением размера таблицы у читателя может сложиться впечатление, что представление операторов в виде конечных предикатов не дает никаких преимуществ и даже усложняет дело, однако это не так. Дело в том, что конечные предикаты, в отличие от конечных алфавитных операторов, которые пришлось бы описывать средствами многозначной логики, допускают весьма удобную аналитическую запись, сходную с формулами алгебры логики [6]. К разработке этой аналитической записи мы сейчас и приступаем.

5. Формулы алгебры конечных предикатов

Для того чтобы получить возможность записывать конечные предикаты в виде формул, мы введем специальную алгебраическую систему, которая называется *алгеброй конечных предикатов*. Точнее, будет введена не одна алгебра, а целое семейство таких алгебр. Каждая алгебра конечных предикатов полностью характеризуется *алфавитом букв* A, состоящим из k символов $a_1, a_2, ..., a_k$, и *алфавитом переменных* B, состоящим из n символов $x_1, x_2,..., x_n$. Средствами алгебры конечных предикатов с алфавитом букв $A=\{a_1, a_2,..., a_k\}$ и алфавитом переменных $B=\{x_1, x_2,..., x_n\}$ может быть записан любой n-местный k-ичный предикат $f(x_1, x_2,..., x_n)$, заданный над алфавитом A. Мы будем считать, что буквы и переменные алфавитов A и B пронумерованы.

Введем понятие *формулы алгебры конечных предикатов* с алфавитом букв $\{a_1, a_2, ..., a_k\}$ и алфавитом переменных $\{x_1, x_2,..., x_n\}$. Формулы будем строить из следующих символов: букв $a_1, a_2,..., a_k$, переменных $x_1, x_2,..., x_n$, знаков дизъюнкции $\vee$ и конъюнкции $\wedge$, открывающей и закрывающей скобок (,), *логических констант* 0 и 1, называемых соответственно *ложью* и *истиной*. Понятие формулы алгебры конечных предикатов определим индуктивно с помощью следующей системы из четырех правил: 1) символы 0 и 1 называем формулами; 2) все выражения вида $a_i(x_j)$, где индекс i находится в пределах от 1 до k, а индекс j – в пределах от 1 до n называем формулами; 3) если выражения A и B являются формулами, то выражение $(A\vee B)$ называем формулой; 4) если выражения A и B – формулы, то выражение $(A\wedge B)$ – тоже формула.

Каждую формулу будем рассматривать как обозначение некоторого предиката $f(x_1, x_2,..., x_n)$, заданного над алфавитом $\{a_1, a_2,..., a_k\}$. Закон соответствия между формулами и обозначаемыми ими предикатами определяем индуктивно следующими правилами: 1) формула 0 обозначает *тождественно ложный предикат*, то есть предикат, равный 0 для всех наборов значений аргументов; 2) формула 1 обозначает *тождественно истинный предикат*, то есть предикат, равный 1 для всех наборов значений аргументов; 3) формула $a_i(x_j)$ обозначает предикат, равный 1 для всех тех наборов значений аргументов, у которых $x_j=a_i$, и равный 0 – для всех остальных наборов. Пусть формула A обозначает предикат f, а формула B – предикат g; тогда: 4) формула $(A\vee B)$ обозначает предикат, равный 0 для всех тех наборов значений аргументов, при которых $f=0$ и $g=0$, и равный 1 – для остальных наборов; 5) формула $(A\wedge B)$ обозначает предикат, равный 1 для всех тех наборов значений аргументов, при которых $f=1$ и $g=1$, и равный 0 – для остальных наборов.

Предикат $(A\vee B)$ называется *дизъюнкцией* или *логической суммой*, а предикат $(A\wedge B)$ – *конъюнкцией* или *логическим произведением предикатов* A и B. Функция, которая ставит в соответствие любым предикатам A и B предикат $(A\vee B)$, называется *операцией дизъюнкции* или *логического сложения предикатов*. Аналогично функцию, сопоставляющую произвольным предикатам A и B предикат $(A\wedge B)$, назовем *операцией конъюнкции* или *логиче-*

ского умножения предикатов. Операции дизъюнкции и конъюнкции будем называть *элементарными операциями алгебры конечных предикатов*. Пусть x, y – логические переменные. Функция $x \vee y$ со значениями $0 \vee 0=0$, $0 \vee 1=1$, $1 \vee 0=1$, $1 \vee 1=1$ называется *функцией дизъюнкции*, а функция $x \wedge y$ со значениями $0 \wedge 0=0$, $0 \wedge 1=0$, $1 \wedge 0=0$, $1 \wedge 1=1$ – *функцией конъюнкции*. В ряде случаев формулы алгебры конечных предикатов удобно рассматривать как значения соответствующих предикатов, то есть как логические константы. При такой интерпретации записи вида $(A \vee B)$ и $(A \wedge B)$ можно рассматривать как функции дизъюнкции и конъюнкции, а входящие в их состав выражения A и B – как логические переменные.

Каждую формулу вида $a_i(x_j)$ удобно рассматривать как одноместный предикат, зависящий только от переменной x_j и определяемый следующим образом:

$$a_i(x_j)=\begin{cases}1, \text{ если } x_j = a_i,\\ 0, \text{ если } x_j \neq a_i.\end{cases} \qquad (3)$$

Предикат $a_i(x_j)$ как бы "узнаёт" одну единственную букву a_i среди возможных букв алфавита A, поэтому назовем его *предикатом узнавания буквы* a_i или просто *узнаванием буквы* a_i, зависящим от переменной x_j. Всего имеется kn различных узнаваний букв. Действуя на различные узнавания букв операциями дизъюнкции и конъюнкции, многократно и в различном порядке, мы получаем различные предикаты. Узнавания букв будем считать *элементарными предикатами алгебры конечных предикатов*. Совокупность элементарных операций и элементарных предикатов образует *базис алгебры конечных предикатов*.

Рассмотрим пример формулы алгебры конечных предикатов. Пусть $k=3$, $n=4$, a_1=а, a_2=м, a_3=п, $x_1=x$, $x_2=y$, $x_3=z$, $x_4=t$. Выражение

$$((\text{а}(y) \wedge \text{а}(t)) \wedge ((\text{м}(x) \wedge \text{м}(z)) \vee (\text{п}(x) \wedge \text{п}(z)))) \qquad (\text{а})$$

есть формула алгебры конечных предикатов. Действительно, согласно второму правилу, выражения а(y), а(t), м(x), м(z), п(x), п(z) – формулы. С помощью четвертого правила из имеющихся уже формул строим следующие формулы: (а(y)$\wedge$а(t)), (м(x)$\wedge$м(z)), (п(x)$\wedge$п(z)). Из двух последних формул по третьему правилу образуем формулу ((м(x)$\wedge$м(z))$\vee$(п(x)$\wedge$п(z))). Наконец, применяя четвертое правило к формулам (а(y)$\wedge$а(t)), ((м(x)$\wedge$м(z))$\vee$(п(x)$\wedge$п(x))), получаем формулу (а).

Недостатком введенных формул является то, что они выглядят неоправданно громоздкими. Это хорошо видно на приведенном примере. Этот недостаток мы компенсируем тем, что будем пользоваться сокращенной записью формул. При переходе от полной записи формулы к сокращенной будем опускать внешние скобки, то есть формулы $(A \vee B)$ и $(A \wedge B)$ мы будем писать в виде $A \vee B$ и $A \wedge B$. Далее, узнавания букв $a_i(x_j)$ будем записывать в более краткой и удобной форме $x_j^{a_i}$, называя букву a_i в этой записи *показателем узнавания буквы*. Наконец, знак конъюнкции в сокращенных записях формул будем заменять точкой или же вовсе опускать $A \wedge B = A \cdot B = AB$.

Кроме того, в записях формул будем опускать все те скобки, наличие которых не обязательно для правильного понимания смысла формулы. Если скобки, регулирующие порядок выполнения действий в сокращенной записи формулы не указаны, то, по принимаемому нами соглашению о старшинстве операций, вначале будем выполнять операции конъюнкции и лишь затем – операции дизъюнкции, например запись $A \vee B \wedge C$ следует понимать как формулу $A \vee (B \wedge C)$. Этим соглашением операция конъюнкции принимается *старшей* по отношению к операции дизъюнкции. При отсутствии скобок в формуле условимся первой из однотипных операций выполнять ту, знак которой стоит в формуле левее, например, $A \vee B \vee C = (A \vee B) \vee C$, $A \wedge B \wedge C = (A \wedge B) \wedge C$. Принятые нами правила сокращения записи формул таковы, что при необходимости всегда можно от сокращенной записи формулы перейти к самой формуле. Применяя, к примеру, только что рассмотренные способы сокращенной записи, формулу (а) можно представить в более компактном и удобном для чтения виде:

$$y^{\text{а}}t^{\text{а}}(x^{\text{м}}z^{\text{м}} \vee x^{\text{п}}z^{\text{п}}). \qquad (\text{б})$$

От формулы всегда можно перейти к таблице обозначаемого ею предиката. Для этого нужно по формуле вычислить значения предиката для всевозможных наборов значений аргументов. Вычислим, к примеру, значения предиката $f(x, y, z, t)$, представленного формулой (б) для наборов значений аргументов (м, а, м, а) и (м, а, п, а):

$$f(\text{м, а, м, а}) = \text{а}^{\text{а}}\text{а}^{\text{а}}(\text{м}^{\text{м}}\text{м}^{\text{м}} \vee \text{м}^{\text{п}}\text{м}^{\text{п}}) =$$
$$= 1 \cdot 1 \cdot (1 \cdot 1 \vee 0 \cdot 0) = 1 \cdot (1 \vee 0) = 1 \cdot 1 = 1,$$
$$f(\text{м, а, п, а}) = \text{а}^{\text{а}}\text{а}^{\text{а}}(\text{м}^{\text{м}}\text{п}^{\text{м}} \vee \text{м}^{\text{п}}\text{п}^{\text{п}}) =$$
$$= 1 \cdot 1 \cdot (1 \cdot 0 \vee 0 \cdot 1) = 1 \cdot (0 \vee 0) = 1 \cdot 0 = 0.$$

Аналогичные вычисления для всевозможных четырехбуквенных наборов, составленных из букв а, м, п (их всего $3^4=81$), показывают, что предикат f «узнаёт» два слова *мама* и *папа*: он ставит им в соответствие значение 1, то есть истину. Остальным словам предикат f ставит в соответствие значение 0, то есть ложь.

Возникает вопрос, *полна* ли введенная нами алгебра конечных предикатов, то есть для каждого ли конечного предиката найдется обозначающая его формула? Оказывается, что алгебра конечных предикатов полна. Действительно, тождественно ложный предикат может быть записан в виде формулы 0.

Предикат, принимающий значение 1 только на одном наборе значений аргументов (σ_1, σ_2,..., σ_n), можно представить формулой $x_1^{\sigma_1}x_2^{\sigma_2}...x_n^{\sigma_n}$. Предикат, принимающий значение 1 на произвольных наборах (σ_{11}, σ_{12},..., σ_{1n}), (σ_{21}, σ_{22},..., σ_{2n}),..., (σ_{m1}, σ_{m2},..., σ_{mn}), может быть представлен формулой:

$$x_1^{\sigma_{11}}x_2^{\sigma_{12}}...x_n^{\sigma_{1n}} \vee x_1^{\sigma_{21}}x_2^{\sigma_{22}}...x_n^{\sigma_{2n}} \vee ... \vee x_1^{\sigma_{m1}}x_2^{\sigma_{m2}}...x_n^{\sigma_{mn}}.$$

Таким образом, любой конечный предикат может быть записан на языке алгебры конечных предикатов.

Алгебра конечных предикатов не только полна, но даже в некотором смысле *избыточна*. Так, мы видим, что введенная в ней первым правилом формула 1 нам не понадобилась для записи произвольных предикатов. Обозначаемый формулой 1 тождественно истинный предикат может быть выражен с помощью других средств алгебры конечных предикатов, например, в форме дизъюнкции всевозможных конъюнкций вида $x_1^{\sigma_1}x_2^{\sigma_2}...x_n^{\sigma_n}$. В случае, когда $k \geq 2$, то есть когда алфавит A содержит по крайней мере две буквы a_1 и a_2, можно обойтись и без формулы 0. Обозначаемый этой формулой тождественно ложный предикат может быть записан, например, в виде формулы $x_1^{a_1}x_1^{a_2}$.

Исключим из этого определения формулы алгебры конечных предикатов первое правило, вводящее формулы 0 и 1; кроме того, предположим, что $k \geq 2$, и рассмотрим вопрос, можно ли в этих условиях еще более упростить определение формулы при сохранении полноты алгебры. Оказывается, что при принятых предположениях алгебра конечных предикатов *несократима* в том смысле, что любые дальнейшие сокращения в оставшихся правилах образования ее формул ведут к неполноте данной алгебры. Действительно, исключим из числа формул одно из выражений: $a_i(x_j)$, вводимое вторым правилом. Тогда мы не сможем образовать формулу для обозначения предиката, обращающегося в 1 на всех тех наборах значений аргументов, у которых $x_j = a_i$, и в 0 – на всех остальных наборах. Если же мы исключим третье правило, тогда станет невозможным представление в виде формулы тождественно истинного предиката. Исключая четвертое правило, мы не сможем представить в виде формулы тождественно ложный предикат.

Таким образом, формулы 0 и 1 не обязательны для алгебры конечных предикатов. Их введение не расширяет выразительные возможности этой алгебры, поскольку и без этих формул алгебра конечных предикатов полна. Символы 0 и 1 включены в число формул лишь потому, что они обеспечивают дополнительные удобства при пользовании алгеброй конечных предикатов. Заметим, что формула 0 для полноты алгебр, у которых $k=1$, необходима. Алгебры этого типа не представляют серьезного интереса для теории интеллекта ввиду их вырожденности. Алфавит A таких алгебр состоит из одной буквы a_1. Предикаты, охватываемые этими алгебрами, принимают значения лишь на одном наборе значений аргументов (a_1, a_1,..., a_1), в котором буква a_1 встречается n раз. Алгеброй охватывается всего два предиката 0 и 1.

6. Тождества алгебры конечных предикатов

Приступим к изучению свойств алгебры конечных предикатов. Сначала рассмотрим ее основные тождества. Назовем *тождественными* формулы, выражающие один и тот же предикат. Факт тождественности формул будем обозначать знаком $\equiv$. Для равенства значений предикатов будем использовать знак $=$. Пусть A, B, C – произвольные формулы алгебры конечных предикатов. Справедливы следующие тождества: *законы коммутативности*

$$A \vee B \equiv B \vee A, \quad (4)$$

$$AB \equiv BA; \quad (5)$$

законы ассоциативности

$$(A \vee B) \vee C \equiv A \vee (B \vee C), \quad (6)$$

$$(AB)C \equiv A(BC); \quad (7)$$

законы дистрибутивности

$$(A \vee B)C \equiv AC \vee BC, \quad (8)$$

$$AB \vee C \equiv (A \vee C)(B \vee C); \quad (9)$$

законы идемпотентности

$$A \vee A \equiv A, \quad (10)$$

$$AA \equiv A; \quad (11)$$

законы элиминации (поглощения)

$$A \vee AB \equiv A, \quad (12)$$

$$A(A \vee B) \equiv A. \quad (13)$$

Названия для приведенных тождеств заимствованы из алгебры логики, где рассматриваются совпадающие с ними по форме законы. Однако в алгебре логики формулы обозначают не конечные предикаты, а булевы функции [7]. Справедливость тождеств легко проверяется перебором всевозможных значений предикатов A, B, C. Например, докажем первый закон коммутативности: $0 \vee 0 \equiv 0 \vee 0$, $0 \vee 1 \equiv 1 \vee 0$, $1 \vee 0 \equiv 0 \vee 1$, $1 \vee 1 \equiv 1 \vee 1$. Справедлив, кроме того, ряд тождеств, в которых участвуют логические константы:

$$A \vee 1 \equiv 1, \quad (14)$$

$$A \cdot 0 \equiv 0, \quad (15)$$

$$A \cdot 1 \equiv A, \quad (16)$$

$$A \vee 0 \equiv A. \quad (17)$$

Пусть x – произвольная буквенная переменная. Имеет место следующее тождество:

$$x^{a_1} \vee x^{a_2} \vee ... \vee x^{a_k} \equiv 1. \quad (18)$$

Действительно, какую бы букву алфавита A мы ни подставили вместо x в левую часть равенства (18), всегда один из дизъюнктивных членов, а вместе с ним и вся дизъюнкция, обращается в 1. Тождество (18) назовем *законом истинности*. Пусть a и b – произвольные, но отличающиеся друг от друга ($a \neq b$) буквы алфавита A. Имеют место следующие тождества:

$$x^a x^b \equiv 0. \qquad (19)$$

Действительно, какую бы букву мы ни подставили вместо x в левую часть равенства (19), всегда хотя бы один из конъюнктивных членов обратится в 0, а вместе с ним обратится в 0 и вся левая часть равенства (19). Тождество (19) назовем *законом ложности*. Закон истинности в некотором смысле можно рассматривать как аналог закона исключенного третьего алгебры логики, а закон ложности – как аналог закона противоречия [1].

С абстрактной точки зрения введенная нами алгебра конечных предикатов есть дистрибутивная решетка с нулем и единицей [8]. А именно – это есть множество всех n-местных k-ичных предикатов с заданными на нем бинарными операциями дизъюнкции и конъюнкции, для которых выполняются аксиомы (4) – (17) дистрибутивной решетки с нулем и единицей. Роль нуля в алгебре конечных предикатов выполняет тождественно ложный предикат 0, роль единицы – тождественно истинный предикат 1. Важно заметить, что в алгебре конечных предикатов выполняются сверх этого еще и тождества (18) и (19). Как будет показано ниже, наличие этих тождеств позволяет ввести в алгебре конечных предикатов третью, унарную операцию – отрицание и рассматривать эту алгебру как булеву алгебру [7]. В силу наличия тождеств (18) и (19) алгебра конечных предикатов оказывается более богатой свойствами, чем булева алгебра, взятая в чистом виде.

Интересно выяснить вопрос: *полна* ли система тождеств, введенных нами в алгебре конечных предикатов. Иными словами, можно ли с помощью этих тождеств доказать тождественность любых двух формул алгебры конечных предикатов, обозначающих один и тот же предикат? Несколько позже будет доказано, что система тождеств (4) – (19) в указанном смысле полна. Эта система тождеств разбивает все множество формул алгебры конечных предикатов на классы эквивалентности [7] таким образом, что каждому из этих классов может быть взаимно однозначно сопоставлен свой конечный предикат.

Не все из перечисленных тождеств, однако, необходимы для достижения полноты. Поэтому число тождеств в системе можно сократить. Например, справедливость тождеств (14) и (15) может быть доказана на основе остальных тождеств:

$$A \vee 1 \equiv 1 \vee A \equiv 1 \vee A \cdot 1 \equiv 1 \vee 1 \cdot A \equiv 1;\ A \cdot 0 \equiv 0 \cdot A \equiv 0 \cdot (0 \vee A) \equiv 0.$$

Из совокупности остальных тождеств выводится второй закон дистрибутивности (9):

$$AB \vee C \equiv AB \vee C \vee CB \equiv AB \vee C \vee CA \vee CB \equiv$$
$$\equiv AB \vee CC \vee CA \vee CB \equiv BA \vee CA \vee BC \vee CC \equiv$$
$$\equiv (B \vee C) \wedge A \vee (B \vee C)\, C \equiv$$
$$\equiv A(B \vee C) \vee C(B \vee C) \equiv (A \vee C)(B \vee C).$$

Было бы интересно указать какую-нибудь несократимую полную систему тождеств алгебры конечных предикатов, удобную в роли системы аксиом для этой алгебры.

Алгебру конечных предикатов можно определить *абстрактно* как некую алгебраическую систему [8]. Рассмотрим множество M, в котором содержатся, по крайней мере, два элемента 0 и 1 и $k \cdot n$ элементов a_{ij} ($1 \le i \le k$, $1 \le j \le n$). Пусть на множестве M заданы две операции $\circ$ и Δ, удовлетворяющие следующим аксиомам:

$$a \circ b \equiv b \circ a,\ a \Delta b \equiv b \Delta a, \qquad (20)$$

$$(a \circ b) \circ c \equiv a \circ (b \circ c),\ (a \Delta b) \Delta c \equiv a \Delta (b \Delta c), \qquad (21)$$

$$(a \circ b) \Delta c \equiv (a \Delta c) \circ (b \Delta c),\ (a \Delta b) \circ c \equiv (a \circ c) \Delta (b \circ c), \qquad (22)$$

$$a \circ a \equiv a,\ a \Delta a \equiv a, \qquad (23)$$

$$a \circ (a \Delta b) \equiv a,\ a \Delta (a \circ b) \equiv a, \qquad (24)$$

$$a \circ 1 \equiv 1,\ a \Delta 0 \equiv 0,\ a \Delta 1 \equiv a,\ a \circ 0 \equiv a, \qquad (25)$$

$$a_{1j} \circ a_{2j} \circ \ldots \circ a_{kj} \equiv 1,\ (1 \le j \le n), \qquad (26)$$

$$a_{i_1 j} \Delta\, a_{i_2 j} \equiv 0,\ (i_1 \neq i_2,\ 1 \le i_1,\ i_2 \le k,\ 1 \le j \le n). \qquad (27)$$

Тождества (20) – (27) назовем *аксиомами абстрактной алгебры конечных предикатов*. Любая алгебраическая система, удовлетворяющая перечисленным требованиям, изоморфна [7] описанной ранее алгебре конечных предикатов. Чтобы перейти от абстрактного определения алгебры конечных предикатов к прежнему ее определению, следует множество M проинтерпретировать как множество всевозможных k-ичных n-местных предикатов, элементы 0 и 1 – как тождественно ложный и тождественно истинный предикаты, элементы a_{ij} – как предикаты $x_j^{a_i}$, операции $\circ$ и Δ – как дизъюнкцию и конъюнкцию предикатов.

Заметим, что только что приведенное определение абстрактной алгебры конечных предикатов допускает еще одну, двойственную первой, интерпретацию. А именно, множество M, как и прежде, интерпретируем как множество всевозможных k-ичных n-местных предикатов, элементы 0 и 1 – как тождественно истинный и тождественно ложный предикаты, элементы a_{ij} как предикаты вида:

$$P_i(x_j) = \begin{cases} 0, \text{ если } x_j = a_i, \\ 1, \text{ если } x_j \neq a_i. \end{cases} \qquad (28)$$

Операции $\circ$ и Δ интерпретируем как конъюнкцию и дизъюнкцию предикатов. Интересно отме-

тить, что в пределах каждой из этих двух интерпретаций абстрактной алгебры конечных предикатов при $k>2$ двойственности не наблюдается (такая двойственность имеет место в алгебре логики и в алгебре конечных предикатов при $k=2$). Хотя при замене знака $\circ$ на Δ и знака 0 на 1, и наоборот, набор тождеств (20) – (25) остается тем же самым, однако тождества (26) и (27) не переходят друг в друга.

Выводы

Может показаться, что алгебра конечных предикатов есть обобщение алгебры логики. Так, мы видим, что обе алгебры относятся к классу булевых алгебр. Кроме того, в частном случае, когда $k=2$ и $A=\{0,1\}$, тождества (18) и (19) превращаются соответственно в закон исключенного третьего $x\vee\bar{x}=1$ и закон противоречия $x\bar{x}=0$ алгебры логики. Здесь обозначено $x^1=x$, $x^0=\bar{x}$. Все тождества (4) – (19) алгебры конечных предикатов в этом случае по форме совпадают с тождествами алгебры логики. И все же неверно было бы утверждать, что алгебра логики – это частный случай алгебры конечных предикатов.

Дело в том, что смысл, вкладываемый в операцию отрицания $\bar{x}$ в алгебре логики и в предикат узнавания нуля x^0 в алгебре конечных предикатов при $k=2$, $A=\{0, 1\}$, различен. В то время как в алгебре логики операцией отрицания можно действовать на любую булеву функцию, в алгебре конечных предикатов при $k=2$, $A=\{0, 1\}$ положение иное: узнаванием нуля разрешается действовать лишь на буквенные аргументы $x_1, x_2,..., x_n$, на предикаты же действовать узнаванием нуля не разрешается. В алгебре конечных предикатов выражения вида A^0, где A – произвольная формула, не являются формулами. А в алгебре логики выражения типа $\bar{A}$ – это формулы.

Таким образом, алгебра логики не является частным случаем алгебры конечных предикатов. В этом обстоятельстве кроется объяснение того факта, что, как было отмечено выше, алгебра конечных предикатов (без 0 и 1 и при $k\geq 2$) несократима. В алгебре же логики набор ее базисных (элементарных) операций (отрицание, дизъюнкция и конъюнкция) можно сократить без потери данной алгеброй свойства полноты (например, можно исключить операцию дизъюнкции), чего не могло бы случиться, если бы алгебра логики была частным случаем алгебры конечных предикатов. Алгебра конечных предикатов при $k=2$ и $A=\{0, 1\}$, как алгебраическая система, насколько нам известно, в литературе специально не рассматривалась. Однако она неявно присутствует во многих руководствах по алгебре логики. Так, например, в литературе можно встретить термин *формулы с тесными отрицаниями*, обозначающий формулы алгебры логики, у которых отрицания могут стоять только непосредственно над независимыми переменными [8]. К такого рода формулам алгебры логики относятся, в частности, дизъюнктивные и конъюнктивные нормальные формы, а также скобочные формы. Эти формы реализуют в виде двухступенчатых или многоступенчатых комбинационных схем, на входы которых поступают двоичные сигналы вместе с их отрицаниями [1].

Список литературы: 1. *Глушков, В.М.* Введение в кибернетику. [Текст] / В.М. Глушков – К.: Изд. АН УССР, 1964. – 324 с. **2.** Энциклопедия кибернетики [Текст] / Т. 1. – К.: Изд. АН УССР, 1974. – 608 с. **3.** *Глушков, В.М.* Алгебра, языки, программирование [Текст] / В.М. Глушков, Г.Е. Цейтлин, Е.Л. Ющенко – К.: Наук. думка, 1974. – 318 с. **4.** *Новиков, П.С.* Элементы математической логики [Текст] / П.С. Новиков – М.: Наука, 1973. – 400 с. **5.** *Шрейдер, Ю.А.* Равенство, сходство, порядок [Текст] / Ю.А. Шрейдер – М.: Наука, 1971. – 256 с. **6.** Дискретная математика и математические вопросы кибернетики [Текст] / Под ред. С.В. Яблонского и О.Б. Лупанова. – М.: Наука, 1974. – Т. 1. – 311 с. **7.** *Мальцев, А.И.* Алгебраические системы [Текст] / А.И. Мальцев – М.: Наука. – 1970. – 394 с. **8.** *Лавров, И.А.* Задачи по теории множеств, математической логике и теории алгоритмов [Текст] / *И.А. Лавров, Л.Л. Максимова* – М.: Наука, 1975. – 247 с.

Поступила в редколлегию 25.05.2011

УДК 519.7

Про алгебру скінченних предикатів / М.Ф. Бондаренко, Ю.П. Шабанов-Кушнаренко // Біоніка інтелекту: наук.-техн. журнал. – 2011. – № 3 (77). – С. 3-13.

Розробляється математична мова опису структур і функцій систем природного інтелекту. Введені поняття скінченного алфавітного оператора, алгебри скінченних предикатів і формул алгебри скінченних предикатів.

Табл. 3. Бібліогр.: 8 найм.

UDC 519.7

About algebra of final predicates / M.F. Bondarenko, Yu.P. Shabanov-Kushnarenko // Bionics of Intelligense: Sci. Mag. – 2011. – № 3 (77). –P. 3-13.

The mathematical language of structures and functions of the natural intellect systems description is developed. The concepts of finite alphabetical operator, algebras of finite predicates and formulas of finite predicates algebra, are entered.

Tab. 3. Ref.: 8 items.

УДК 519.7

М.Ф. Бондаренко, Ю.П. Шабанов-Кушнаренко
ХНУРЭ, г. Харьков, Украина

НОРМАЛЬНЫЕ ФОРМЫ ФОРМУЛ АЛГЕБРЫ КОНЕЧНЫХ ПРЕДИКАТОВ

Рассмотрены задачи построения нормальных форм формул алгебры конечных предикатов, показано, что между конечными предикатами и совершенными дизъюнктивными нормальными формами существует взаимно однозначное соответствие. Сформулирована теорема о конъюнктивном разложении, рассмотрена задача канонической минимизации формул алгебры конечных предикатов, методы дизъюнктивной и конъюнктивной минимизации.

ТЕОРИЯ ИНТЕЛЛЕКТА, ПРЕДИКАТ, АЛГЕБРА КОНЕЧНЫХ ПРЕДИКАТОВ, ДНФ, СДНФ, МИНИМИЗАЦИЯ ФОРМУЛ

Введение

Наличие тождеств в алгебре конечных предикатов свидетельствует о том, что один и тот же конечный предикат можно записать в виде различных формул. Значит, формул алгебры конечных предикатов больше, чем обозначаемых ими предикатов. Возникает задача: из множества всех формул выделить класс формул, называемых нормальными формами, чтобы в нем каждому предикату соответствовала только одна формула алгебры конечных предикатов. Выделим два таких класса формул: класс *совершенных дизъюнктивных нормальных форм* (*СДНФ*) и класс *совершенных конъюнктивных нормальных форм* (*СКНФ*).

1. Совершенная дизъюнктивная нормальная форма

Вначале введем понятие совершенной дизъюнктивной нормальной формы. *Элементарной конъюнкцией* назовем любую конъюнкцию узнаваний различных буквенных переменных, взятых с произвольными фиксированными показателями. Наибольшее число множителей в элементарной конъюнкции равно n, наименьшее – нулю. В качестве элементарной конъюнкции, не имеющей в своем составе ни одного узнавания буквы, примем формулу 1. Примерами элементарных конъюнкций в алгебре конечных предикатов с алфавитом букв $A=\{a, b, c\}$ и алфавитом переменных $B=\{x_1, x_2, x_3\}$ могут служить формулы $x_1^a x_2^b$, $x_2^c x_3^b$, $x_1^b x_2^c x_3^c$. Элементарные конъюнкции, отличающиеся между собой только порядком конъюнктивных членов, будем считать одинаковыми, например, $x_1^b x_2^c x_3^c$ и $x_2^c x_1^b x_3^c$. Любую дизъюнкцию произвольного числа различных элементарных конъюнкций назовем *дизъюнктивной нормальной формой* (*ДНФ*). В качестве ДНФ, не содержащей ни одного дизъюнктивного члена, примем формулу 0. Примером ДНФ в той же алгебре может служить формула $x_1^a x_2^b \vee x_2^c x_3^c \vee x_1^b x_2^c x_3^c$.

Любая элементарная конъюнкция, в которой встречаются все переменные алгебры конечных предикатов, называется *конституэнтой единицы*. Условимся все узнавания в конституэнте единицы располагать в порядке возрастания номеров переменных. Примеры конституэнт единицы (в той же алгебре): $x_1^a x_2^a x_3^b$, $x_1^b x_2^c x_3^b$. В общем виде конституэнта единицы запишется следующим образом:

$$x_1^{\sigma_1} x_2^{\sigma_2} \ldots x_n^{\sigma_n} \equiv \bigwedge_{i=1}^{n} x_i^{\sigma_i} .$$

Здесь $\sigma_1, \sigma_2, \ldots, \sigma_n$ – некоторые фиксированные буквы алфавита A, знак $\bigwedge_{i=1}^{n}$ обозначает операцию логического перемножения n членов, i – индекс, изменяющийся в пределах от 1 до n, по которому ведется перемножение.

Присвоим каждой конституэнте единицы $x_1^{\sigma_1} x_2^{\sigma_2} \ldots x_n^{\sigma_n}$ свой *номер*. Для этого составим n-разрядный k-ичный код $\sigma_1\sigma_2\ldots\sigma_n$ из показателей ее узнаваний, рассматривая буквы a_1, a_2,..., a_k алфавита A как k-ичные цифры 0, 1, ..., k-1. Число, соответствующее коду $\sigma_1\sigma_2\ldots\sigma_n$, будем считать номером данной конституэнты единицы. Например, номером конституэнты единицы $x_1^b x_2^a x_3^c$ служит число 11 (в десятичной записи), соответствующее троичному коду 102. Всего имеется k^n различных конституэнт единицы.

Любая дизъюнкция произвольного числа n различных конституэнт единицы называется *совершенной дизъюнктивной нормальной формой* (*СДНФ*). Мы будем отождествлять между собой все СДНФ, отличающиеся только порядком расположения конституэнт единицы. Конституэнты единицы условимся располагать в порядке возрастания их номеров. Пример СДНФ (в той же алгебре): $x_1^a x_2^a x_3^c \vee x_1^b x_2^b x_3^a \vee x_1^c x_2^c x_3^b$. Общий вид СДНФ следующий:

$$x_1^{\sigma_{11}} x_2^{\sigma_{12}} \ldots x_n^{\sigma_{1n}} \vee x_1^{\sigma_{21}} x_2^{\sigma_{22}} \ldots x_n^{\sigma_{2n}} \vee \ldots \vee x_1^{\sigma_{m1}} x_2^{\sigma_{m2}} \ldots x_n^{\sigma_{mn}} \equiv$$
$$\equiv \bigvee_{i=1}^{m} x_1^{\sigma_{i1}} x_2^{\sigma_{i2}} \ldots x_n^{\sigma_{in}} \equiv \bigvee_{i=1}^{m} \bigwedge_{j=1}^{n} x_j^{\sigma_{ij}} .$$

Здесь σ_{ij} – некоторые фиксированные буквы алфавита A; знак $\bigvee_{i=1}^{m}$ обозначает операцию логического суммирования m членов; m – число конституэнт

единицы в СДНФ; i, j – индексы, изменяющиеся в пределах соответственно от 1 до m и от 1 до n, по которым ведутся логическое суммирование и логическое перемножение.

Присвоим каждой СДНФ свой номер. Для этого каждой СДНФ поставим в соответствие некоторый двоичный код длины k^n. Длина кода совпадает с числом всех различных конституэнт единицы. Если в рассматриваемой СДНФ конституэнта единицы с номером i отсутствует, то в i-том разряде ее двоичного кода записываем 0, если присутствует, то записываем 1. Нумерацию разрядов двоичного кода в данном случае начинаем с нуля и ведем слева направо. Число, соответствующее полученному таким способом двоичному коду, будем считать номером СДНФ. В качестве СДНФ с номером 0 принимаем формулу 0. Например, номер СДНФ $x_1^a x_2^b \vee x_1^b x_2^c$ в алгебре с алфавитами $A=\{a, b, c\}$, $B=\{x_1, x_2\}$ будет число 136_{10} , соответствующее двоичному коду 010001000. Всего имеется 2^{kn} различных СДНФ, то есть ровно столько, сколько существует всех различных конечных предикатов. Вместе с тем, каждой формуле соответствует единственный конечный предикат, каждому же конечному предикату соответствует некоторая своя СДНФ. Следовательно, между конечными предикатами и совершенными дизъюнктивными нормальными формами существует взаимно однозначное соответствие.

Важно выяснить вопрос: можно ли с помощью приведенных выше тождеств преобразовать произвольную формулу алгебры конечных предикатов к СДНФ. Если да, то отсюда будет следовать, что система этих тождеств *полна*. Действительно, сравнивая между собой СДНФ двух формул, всегда можно, в силу единственности представления любого конечного предиката в виде СДНФ, решить вопрос о тождестве этих формул. Если СДНФ совпадают, то исходные формулы тождественны, если не совпадают, то они соответствуют различным предикатам. Такое преобразование существует. Следовательно, система тождеств (4) – (19) [1] алгебры конечных предикатов полна.

Ниже приводится описание алгоритма преобразования произвольной формулы к СДНФ. Алгоритм сопровождается примером: $A=\{a, b\}$, $B=\{x_1, x_2, x_3\}$, требуется преобразовать к СДНФ формулу

$$f\equiv(x_1^a\vee x_2^b)(x_1^a\vee x_1^b x_3^a)\vee(x_1^a\vee x_2^a)(x_2^b x_3^b\vee x_2^a x_3^a).$$

1) Пользуясь тождествами (5), (7) и (8) [1], раскрываем в формуле все скобки:

$$f\equiv x_1^a(x_1^a\vee x_1^b x_3^a)\vee x_2^b(x_1^a\vee x_1^b x_3^a)\vee$$
$$\vee x_1^a(x_2^b x_3^b\vee x_2^a x_3^a)\vee x_2^a(x_2^b x_3^b\vee x_2^a x_3^a)\equiv$$
$$\equiv(x_1^a\vee x_1^b x_3^a)x_1^a\vee(x_1^a\vee x_1^b x_3^a)x_2^b\vee(x_2^b x_3^b\vee x_2^a x_3^a)x_1^a\vee$$
$$\vee(x_2^b x_3^b\vee x_2^a x_3^a)x_2^a\equiv x_1^a x_1^a\vee x_1^b x_3^a x_1^a\vee x_1^a x_2^b\vee x_1^b x_3^a x_2^b\vee$$
$$\vee x_2^b x_3^b x_1^a\vee x_2^a x_3^a x_1^a\vee x_2^b x_3^b x_2^a\vee x_2^a x_3^a x_2^a.$$

В результате получаем некоторую дизъюнкцию конъюнкций узнаваний предмета.

2) Пользуясь тождествами (4)-(7), (10), (11), (15), (17) и (19) [1], производим упрощения в формуле:

$$f\equiv x_1^a\vee 0\cdot x_3^a\vee x_1^a x_2^b\vee x_1^b x_2^b x_3^a\vee x_1^a x_2^b x_3^b\vee x_1^a x_2^a x_3^a\vee$$
$$\vee 0\cdot x_3^b\vee x_2^a x_3^a\equiv x_1^a\vee 0\vee x_1^a x_2^b\vee x_1^b x_2^b x_3^a\vee x_1^a x_2^b x_3^b\vee$$
$$\vee x_1^a x_2^a x_3^a\vee 0\vee x_2^a x_3^a\equiv x_1^a\vee x_1^a x_2^b\vee$$
$$\vee x_1^b x_2^b x_3^a\vee x_1^a x_2^b x_3^b\vee x_1^a x_2^a x_3^a\vee x_2^a x_3^a.$$

В результате получаем некоторую дизъюнктивную нормальную формулу предиката.

3) Пользуясь тождествами (16) и (18) [1], во все конъюнкции вводим недостающие переменные:

$$f\equiv x_1^a(x_2^a\vee x_2^b)(x_3^a\vee x_3^b)\vee x_1^a x_2^b(x_3^a\vee x_3^b)\vee x_1^b x_2^b x_3^a\vee$$
$$\vee x_1^a x_2^b x_3^b\vee x_1^a x_2^a x_3^a\vee(x_1^a\vee x_1^b)x_2^a x_3^a.$$

4) Пользуясь тождествами (4)-(8) и (10) [1], снова раскрываем скобки и производим упрощения:

$$f\equiv x_1^a x_2^a x_3^a\vee x_1^a x_2^a x_3^b\vee x_1^a x_2^b x_3^a\vee x_1^a x_2^b x_3^b\vee x_1^a x_2^b x_3^a\vee$$
$$\vee x_1^a x_2^b x_3^b\vee x_1^b x_2^b x_3^a\vee x_1^b x_2^b x_3^a\vee x_1^a x_2^b x_3^b\vee x_1^a x_2^a x_3^a\vee$$
$$\vee x_1^a x_2^a x_3^a\vee x_1^b x_2^a x_3^a\equiv x_1^a x_2^a x_3^a\vee x_1^a x_2^a x_3^b\vee x_1^a x_2^b x_3^a\vee$$
$$\vee x_1^a x_2^b x_3^b\vee x_1^b x_2^a x_3^a\vee x_1^b x_2^b x_3^a.$$

В результате получаем искомую СДНФ.

От совершенной дизъюнктивной нормальной формы нетрудно перейти к таблице обозначаемого ею конечного предиката. Для этого нужно выделить в таблице все наборы значений аргументов, совпадающие с наборами показателей узнаваний букв конституэнт единицы, фигурирующих в СДНФ. Против выделенных наборов надо выписать значение предиката 1, против остальных наборов – значение 0. Рассмотрим пример. Пусть $A=\{a, b, c\}$, $B=\{x_1, x_2\}$. Предикат задан совершенной дизъюнктивной нормальной формой $t\equiv x_1^a x_2^b\vee x_1^a x_2^c\vee x_1^b x_2^a$. Этому предикату соответствует табл. 1.

Таблица 1

x_1	a	a	a	b	b	b	c	c	c
x_2	a	b	c	a	b	c	a	b	c
t	0	1	1	1	0	0	0	0	0

Для обратного перехода от таблицы значений конечного предиката к его СДНФ выделяем в таблице все те наборы аргументов, на которых этот предикат обращается в 1. Далее, для каждого такого набора выписываем соответствующую конституэнту единицы, принимая в ней показатели узнаваний букв в соответствии с этим набором. Наконец, все полученные таким способом конституэнты единицы соединяем знаками дизъюнкции. Рассмотрим пример. Предикат задан табл. 2.

Таблица 2

x_1	0	0	0	1	1	1	2	2	2
x_2	0	1	2	0	1	2	0	1	2
t	1	0	0	0	1	0	0	1	0

СДНФ этого предиката имеет вид:

$$x_1^{\,0}x_2^{\,0}\vee x_1^{\,1}x_2^{\,1}\vee x_1^{\,2}x_2^{\,1}\equiv t.$$

Заметим, что таблицу значений предиката можно легко построить также и по его произвольной дизъюнктивной нормальной форме. Для этого нужно выделить в таблице те наборы значений аргументов, в состав которых входят наборы показателей узнаваний элементарных конъюнкций ДНФ. Против выделенных наборов проставляем значения 1, против остальных наборов – значение 0. Рассмотрим пример. Пусть $A=\{\alpha, \beta\}$, $B=\{x, y, z\}$. Предикат задан следующей ДНФ:

$$t\equiv x^{\alpha}y^{\beta}\vee x^{\beta}z^{\alpha}.$$

Ему соответствует табл. 3.

Таблица 3

x	α	α	α	α	β	β	β	β
y	α	α	β	β	α	α	β	β
z	α	β	α	β	α	β	α	β
t	0	0	1	1	1	0	1	0

В ряде случаев бывает полезно, отправляясь от задания конечного предиката в виде произвольной формулы алгебры конечных предикатов, получить какую-нибудь по возможности *экономную* ДНФ. С этой целью можно воспользоваться первыми двумя шагами описанного выше алгоритма приведения к СДНФ. Полученную ДНФ следует попытаться упростить, применяя первый закон поглощения (12) и первый закон идемпотентности (10) [1]. Подчеркнем, что по данному методу мы находим формулу, не обязательно простейшую среди всевозможных ДНФ. Однако она, как правило, оказывается не намного сложнее самой простой ДНФ. На отыскание такой экономной формулы затрачивается гораздо меньше усилий. Для примера произведем упрощение ДНФ, полученной на втором шаге в примере только что упомянутого алгоритма преобразования к СДНФ:

$$x_1^{\,a}\vee x_1^{\,a}x_2^{\,b}\vee x_1^{\,b}x_2^{\,b}x_3^{\,a}\vee x_1^{\,a}x_2^{\,b}x_3^{\,b}\vee x_1^{\,a}x_2^{\,a}x_3^{\,a}\vee x_2^{\,a}x_3^{\,a}=$$
$$=x_1^{\,a}\vee x_1^{\,b}x_2^{\,b}x_3^{\,a}\vee x_2^{\,a}x_3^{\,a}.$$

В алгебре конечных предикатов имеет место важное утверждение, которое назовем *теоремой о разложении*. Можно сформулировать два варианта этой теоремы: *теорему о дизъюнктивном разложении* и *теорему о конъюнктивном разложении*. Сформулируем и докажем теорему о дизъюнктивном разложении. Теорему о конъюнктивном разложении рассмотрим несколько позже. Формулируется теорема о дизъюнктивном разложении следующим образом.

Любой конечный предикат $f(x_1, x_2,..., x_n)$ *может быть представлен в виде*

$$f(x_1, x_2,..., x_l, x_{l+1},..., x_n)\equiv$$
$$\equiv\bigvee_{(\sigma_1,\sigma_2,...,\sigma_l)} x_1^{\sigma_1}x_2^{\sigma_2}...x_l^{\sigma_l}f(s_1, s_2,..., s_l, x_{l+1},..., x_n). \qquad (1)$$

Представление предиката $f(x_1, x_2,..., x_n)$ в виде правой части тождества (1) назовем его *дизъюнктивным разложением* по переменным $x_1, x_2,..., x_l$. Буквенные переменные $\sigma_1, \sigma_2,..., \sigma_l$ играют роль индексов при образовании многократной дизъюнкции. Запись $(\sigma_1, \sigma_2,..., \sigma_l)$ под знаком дизъюнкции означает, что логическая сумма берется по всевозможным наборам индексов $\sigma_1, \sigma_2,..., \sigma_l$. Таким образом, в правой части тождества (1) присутствуют k^l дизъюнктивных членов.

Докажем теорему о дизъюнктивном разложении. Доказательство будем вести индукцией по k, l и n.

1. При $k=l=n=1$ равенство (1) принимает вид:

$$f(x_1)\equiv\bigvee_{(\sigma_1)} x_1^{\sigma_1}f(\sigma_1).$$

Оно является тождеством. Действительно, переменная x_1 принимает единственное возможное значение a_1, поэтому $f(x_1)\equiv f(a_1)$ и

$$\bigvee_{(\sigma_1)} x_1^{\sigma_1}f(\sigma_1)\equiv x_1^{\,a_1}f(a_1)\equiv 1\cdot f(a_1).$$

Таким образом, $k=l=n=1$, теорема о разложении справедлива.

2. Пусть $l=n=1$. Предположим, что при $k=t$ равенство (1) является тождеством и выведем отсюда, что при $k=t+1$ равенство (1) тоже будет тождеством. По индуктивному предположению имеем тождество

$$f(x_1)\equiv\bigvee_{(\sigma_1)} x_1^{\sigma_1}f(\sigma_1)\equiv$$
$$x_1^{\,a_1}f(a_1)\vee x_1^{\,a_2}f(a_2)\vee...\vee x_1^{\,a_t}f(a_t), \qquad (а)$$

которое является частным случаем тождества (1) при $l=n=1$ и $k=t$. Для каждого предиката $f(x_1)$ заданного на множестве $A_t=\{a_1, a_2,..., a_t\}$, построим два предиката: $f'(x_1)$ и $f''(x_1)$, определенных на множестве $A_{t+1}=\{a_1, a_2,..., a_t, a_{t+1}\}$, задавая их следующими условиями:

$$f'(x_1)=\begin{cases}0, \text{если } x_1=a_{t+1},\\ f(x_1), \text{если } x_1\neq a_{t+1},\end{cases}$$

$$f''(x_1)=\begin{cases}1, \text{если } x_1=a_{t+1},\\ f(x_1), \text{если } x_1\neq a_{t+1}.\end{cases}$$

Докажем, что для указанных предикатов справедливо тождество (1), то есть:

$$f'(x_1)\equiv\bigvee_{(\sigma_1)} x_1^{\sigma_1}f'(\sigma_1)\equiv x_1^{\,a_1}f'(a_1)\vee x_1^{\,a_2}f'(a_2)\vee...$$

$$...\vee x_1^{\,a_t}f'(a_t)\vee x_1^{\,a_{t+1}}f'(a_{t+1}),$$

$$f''(x_1)\equiv\bigvee_{(\sigma_1)} x_1^{\sigma_1}f''(\sigma_1)\equiv x_1^{\,a_1}f''(a_1)\vee x_1^{\,a_2}f''(a_2)\vee...$$

$$...\vee x_1^{\,a_t}f''(a_t)\vee x_1^{\,a_{t+1}}f''(a_{t+1}). \qquad (б)$$

Действительно, если $x_1\neq a_{t+1}$, то $f'(x_1)\equiv f''(x_1)\equiv f(x_1)$ и $x_1^{\,a_{t+1}}\equiv 0$. В этом случае, согласно (а), равенства (б) обращаются в тождества. Если же $x_1=a_{t+1}$, то $f'(x_1)\equiv 0$, $f''(x_1)\equiv 1$, $x_1^{\,a_1}\equiv x_1^{\,a_2}\equiv...\equiv x_1^{\,a_t}\equiv 0$, $x_1^{\,a_{t+1}}\equiv 1$, $f'(a_{t+1})\equiv 0$,

$f''(a_{t+1})\equiv 1$, поэтому равенства (б) также обращаются в тождества типа $0\equiv 0$ или $1\equiv 1$. Заметим, что предикатами $f'(x_1)$ и $f''(x_1)$ исчерпывается класс возможных одноместных предикатов, заданных на множестве A_{t+1}. Таким образом, при $l=n=1$ теорема о разложении справедлива для любого k.

3. Пусть $l=1$ и k произвольным образом зафиксировано. Предположим, что при $n=t$ равенство (1) является тождеством, и выведем отсюда, что и при $n=t+1$ равенство (1) будет тождеством. По предположению, согласно (1), имеет место тождество

$$f(x_1, x_2,..., x_t)\equiv \bigvee_{(\sigma_1)} x_1^{\sigma_1} f(\sigma_1, x_2,..., x_t).$$

Поэтому при любом a_i $(1\le i\le k)$

$$f(x_1, x_2,..., x_t, a_i)\equiv \bigvee_{(\sigma_1)} x_1^{\sigma_1} f(\sigma_1, x_2,..., x_t, a_i).$$

Следовательно,

$$f(x_1, x_2,..., x_t, x_{t+1})\equiv \bigvee_{(\sigma_1)} x_1^{\sigma_1} f(\sigma_1, x_2,..., x_t, x_{t+1}).$$

Заметим, что предикатами, для которых справедливо записанное выше тождество, исчерпывается весь класс всевозможных $t+1$-местных предикатов. Таким образом, при $l=1$ теорема о разложении справедлива для любых k и n.

4. Пусть k и n произвольно фиксированы. Предположим, что при $l=t$ равенство (1) является тождеством, и выведем отсюда, что и при $l=t+1$, если $l\le n$, равенство (1) будет тождеством. По предположению, согласно (1), имеем тождество:

$$f(x_1, x_2,..., x_t, x_{t+1},..., x_n)\equiv$$

$$\equiv \bigvee_{(\sigma_1,\sigma_2,..,\sigma_t)} x_1^{\sigma_1} x_2^{\sigma_2}...x_t^{\sigma_t} f(\sigma_1, \sigma_2,..., \sigma_t, x_{t+1},..., x_n).$$

Пользуясь им, а также результатом, полученным в пункте 3 настоящего доказательства, имеем:

$$f(x_1, x_2,..., x_t, x_{t+1}, x_{t+2},..., x_n)\equiv$$

$$\equiv \bigvee_{(\sigma_1,\sigma_2,..,\sigma_t)} x_1^{\sigma_1} x_2^{\sigma_2}...x_t^{\sigma_t} f(\sigma_1, \sigma_2,..., \sigma_t, x_{t+1}, x_{t+2},..., x_n)\equiv$$

$$\equiv \bigvee_{(\sigma_1,\sigma_2,..,\sigma_t)} x_1^{\sigma_1} x_2^{\sigma_2}...x_t^{\sigma_t} \Big(\bigvee_{(\sigma_{t+1})} x_{t+1}^{\sigma_{t+1}} f(\sigma_1, \sigma_2,..., \sigma_t, \sigma_{t+1},$$

$$x_{t+2},..., x_n)\Big)\equiv \bigvee_{(\sigma_1,\sigma_2,...,\sigma_{t+1})} x_1^{\sigma_1} x_2^{\sigma_2}...x_{t+1}^{\sigma_{t+1}} f(\sigma_1, \sigma_2,...,$$
$$\sigma_{t+1}, x_{t+2},..., x_n).$$

Таким образом, теорема доказана для любых k, l, n.

Из теоремы о дизъюнктивном разложении вытекают следующие два *следствия*.

Следствие 1. *Любой конечный предикат* $f(x_1, x_2,..., x_n)$ *может быть представлен в виде*:

$$f(x_1, x_2,..., x_n)\equiv x_1^{a_1} f(a_1, x_2,..., x_n)\vee x_1^{a_2} f(a_2,$$

$$x_2,...,x_n)\vee...\vee x_1^{a_k} f(a_k, x_2,..., x_n). \qquad (2)$$

Следствие 2. *Любой конечный предикат* $f(x_1, x_2,..., x_n)$ *может быть представлен в виде*:

$$f(x_1, x_2,..., x_n)\equiv \bigvee_{f(\sigma_1,\sigma_2,...,\sigma_n)=1} x_1^{\sigma_1} x_2^{\sigma_2}...x_n^{\sigma_n}. \qquad (3)$$

Тождества (2) и (3) получаем, полагая в (1) соответственно $l=1$ и $l=n$. Запись $f(\sigma_1, \sigma_2,..., \sigma_n)=1$ в (3) под знаком дизъюнкции означает, что логическое суммирование ведется только по тем наборам индексов σ_1, σ_2,..., σ_n, которые обращают предикат f в 1. Формула, стоящая в правой части тождества (3), представляет собой СДНФ предиката f. Если предикат f тождественно ложный, то, согласно теореме о дизъюнктивном разложении, в правой части тождества (3) нужно ставить 0.

Тождество (2) может быть использовано в качестве эффективного средства упрощения формул алгебры конечных предикатов. Рассмотрим пример такого упрощения. Пусть $A=\{a, b\}$, $B=\{x_1, x_2, x_3,\}$. Требуется упростить формулу:

$$f(x_1, x_2, x_3)\equiv$$
$$\equiv(x_1^a\vee x_2^b)(x_1^a\vee x_1^b x_3^a)\vee(x_1^a\vee x_1^a)(x_2^b x_3^b\vee x_2^a x_3^a).$$

Производим разложение функции f по переменной x_1:

$$f(x_1, x_2, x_3)\equiv x_1^a f(a, x_2, x_3)\vee x_1^b f(b, x_2, x_3),$$
$$f(a, x_2, x_3)\equiv(a^a\vee x_2^b)(a^a\vee a^b x_3^a)\vee(a^a\vee x_2^a)(x_2^b x_3^b\vee$$
$$\vee x_2^a x_3^a)\equiv(1\vee x_2^b)(1\vee 0\cdot x_3^a)\vee(1\vee x_2^a)(x_2^b x_3^b\vee x_2^a x_3^a)\equiv 1,$$
$$f(b, x_2, x_3)\equiv(b^a\vee x_2^b)(b^a\vee b^b x_3^a)\vee(b^a\vee x_2^a)(x_2^b x_3^b\vee x_2^a x_3^a)\equiv$$
$$\equiv x_2^b x_3^a\vee x_2^a(x_2^b x_3^b\vee x_2^a x_3^a)\equiv x_2^b x_3^a\vee x_2^a x_3^a\equiv$$
$$\equiv(x_2^a\vee x_2^b)x_3^a\equiv x_3^a.$$

Окончательно:

$$f\equiv x_1^a\vee x_1^b x_3^a\equiv(x_1^a\vee x_1^b)(x_1^a\vee x_3^a)\equiv x_1^a\vee x_3^a.$$

Из второго следствия теоремы о разложении вытекает полнота алгебры конечных предикатов: любой конечный предикат можно записать в виде формулы алгебры конечных предикатов. Тождество (3) можно использовать для получения СДНФ предиката, заданного какой-нибудь произвольно выбранной формулой алгебры конечных предикатов. Пример: $A=\{a, b\}$, $B=\{x_1, x_2, x_3,\}$, дан предикат:

$$f(x_1, x_2, x_3)\equiv$$
$$\equiv(x_1^a\vee x_1^b)(x_1^a\vee x_1^b x_3^a)\vee(x_1^a\vee x_2^a)(x_2^b x_3^b\vee x_2^a x_3^a).$$

Требуется отыскать СДНФ указанного предиката. Имеем:

$$f(a, a, a)\equiv(a^a\vee a^b)(a^a\vee a^b a^a)\vee(a^a\vee a^a)(a^b a^b\vee a^a a^a)\equiv 1.$$

Аналогично находим: $f(a, a, b)\equiv 1$, $f(a, b, a)\equiv 1$, $f(a, b, b)\equiv 1$, $f(b, a, a)\equiv 1$, $f(b, a, b)\equiv 1$, $f(b, b, a)\equiv 1$, $f(b, b, b)\equiv 0$. СДНФ предиката f имеет вид:

$$f(x_1, x_2, x_3)\equiv x_1^a x_2^a x_3^a\vee x_1^a x_2^a x_3^b\vee$$
$$\vee x_1^a x_2^b x_3^a\vee x_1^a x_2^b x_3^b\vee x_1^b x_2^a x_3^a\vee x_1^b x_2^b x_3^a.$$

2. Совершенная конъюнктивная нормальная форма

Введем в алгебре конечных предикатов операцию отрицания. Введение отрицания позволит показать, что алгебра конечных предикатов есть

булева алгебра. Кроме того, наличие операции отрицания даст возможность в ряде случаев записывать конечные предикаты в форме более компактных выражений, а также выполнять тождественные преобразования формул более коротким путем. Наконец, операция отрицания позволит естественным образом ввести понятие совершенной конъюнктивной нормальной формы и сформулировать теорему о конъюнктивном разложении предиката.

Операция отрицания предиката – это одноместная функция, заданная на множестве всех k-ичных n-местных предикатов со значениями в том же множестве. Если A – аргумент и B – значение операции отрицания, то условимся писать $(\neg A)=B$ или, в сокращенной форме $\overline{A}=B$. *Отрицанием предиката* $f(x_1, x_2,..., x_n)$ назовем предикат

$$g(x_1, x_2,..., x_n)\equiv \overline{f}\ (x_1, x_2,..., x_n),$$

принимающий значение 0 для всех тех наборов значений аргументов $(x_1, x_2,..., x_n)$, при которых $f(x_1, x_2,..., x_n)=1$, и принимающий значение 1 для всех тех наборов значений аргументов, при которых $f(x_1, x_2,..., x_n)=0$.

При $k\geq 2$ можно определить операцию отрицания аксиоматически, задав ее следующими тождествами:

$$\overline{A\vee B}\equiv \overline{A}\,\overline{B}\,, \qquad (4)$$

$$\overline{A}\,\overline{B}\equiv \overline{A}\vee \overline{B}\,, \qquad (5)$$

$$\overline{x^{a_i}}\equiv x^{a_1}\vee x^{a_2}\vee ...\vee x^{a_{i-1}}\vee x^{a_{i+1}}\vee ...\vee x^{a_k}\,. \qquad (6)$$

Здесь A и B – произвольные формулы, x – произвольная буквенная переменная, a_i – произвольная буква алгебры конечных предикатов, индекс i принимает значения в пределах от 1 до k. Тождества (4) и (5) совпадают по форме с известными в алгебре логики *законами де Моргана*, которые, однако, применяются теперь не к булевым переменным, а к переменным предикатам. Тождество (6) назовем *законом отрицания*. Заметим, что при $k=1$ закон отрицания теряет смысл.

Справедливость законов де Моргана доказывается проверкой тождеств для всех возможных вариантов значений предикатов A и B. Например, убеждаемся при $A=0$ и $B=1$, что $\overline{0\vee 1}\equiv \bar{0}\cdot\bar{1}$ и $\overline{0\cdot 1}\equiv \bar{0}\vee\bar{1}$. Легко устанавливается также справедливость закона отрицания. Действительно, если $x=a_i$, то предикаты, стоящие слева и справа от знака тождества в (6), обращаются в 0, если же $x\neq a_i$, то оба предиката обращаются в 1.

Пользуясь приведенными ранее тождествами алгебры конечных предикатов, а также тождествами (4) – (6), можно доказать, что:

$$\bar{0}\equiv 1\,, \qquad (7)$$

$$\bar{1}\equiv 0. \qquad (8)$$

Действительно,

$$\bar{0}\equiv \overline{x^{a_1}x^{a_2}}\equiv \overline{x^{a_1}}\vee\overline{x^{a_2}}\equiv x^{a_2}\vee x^{a_3}\vee ...\vee x^{a_k}\vee x^{a_1}\vee x^{a_3}\vee$$
$$\vee ...\vee x^{a_k}\equiv 1,$$

$$\bar{1}\equiv -\overline{x^{a_1}\vee x^{a_2}\vee ...\vee x^{a_k}}\equiv \overline{x^{a_1}}\,\overline{x^{a_2}}...\overline{x^{a_k}}\equiv$$

$$\equiv (x^{a_2}\vee x^{a_3}\vee ...\vee x^{a_k})(x^{a_1}\vee x^{a_3}\vee ...\vee x^{a_k})...$$

$$...(x^{a_1}\vee x^{a_2}\vee\vee ...\vee x^{a_{k-1}})\equiv 0.$$

Любая формула с отрицаниями может быть преобразована с помощью зависимостей (4) – (8) в тождественную ей формулу без отрицаний. Действительно, применяя многократно законы де Моргана, мы всегда сможем все знаки отрицания, стоящие над формулой или ее частями, опустить непосредственно на узнавания букв или на 0 и 1. Затем, пользуясь тождествами (6) – (8), исключаем отрицания из формулы. Рассмотрим пример исключения знаков отрицания из формулы. Пусть $A=\{a, b, c\}$. Тогда:

$$\overline{x_1^a\vee x_1^b x_2^c}\equiv \overline{x_1^a}\,\overline{x_1^b x_2^c}\equiv \overline{x_1^a}(\overline{x_1^b}\vee\overline{x_2^c})\equiv$$
$$\equiv (x_1{}^b\vee x_1{}^c)(x_1{}^a\vee x_1{}^c\vee x_2{}^a\vee x_2{}^b).$$

С помощью тождеств (4) – (6) совместно с перечисленными выше основными тождествами алгебры конечных предикатов можно решить вопрос о тождественности двух любых формул со знаками отрицания. Сначала исключаем из формул знаки отрицания, а затем приводим формулы к СДНФ. Отсюда следует, что система тождеств (4) – (6) полна, она совместно с тождествами (4) – (19) [1] аксиоматически определяет все свойства операции отрицания.

В ряде случаев процесс исключения знаков отрицания из формул существенно упрощается, если использовать следующие тождества:

$$x^{a_i}\overline{x^{a_j}}\equiv x^{a_i}\,, \qquad (9)$$

$$x^{a_i}(\overline{x^{a_j}}\vee A)\equiv x^{a_i}\,. \qquad (10)$$

Тождества справедливы при условии, что $i\neq j$, индексы i и j принимают значения в пределах от 1 до k. Буквой A обозначена произвольная формула алгебры конечных предикатов. Тождество (9) назовем *законом поглощения отрицания*, тождество (10) – *обобщенным законом поглощения отрицания*. Докажем справедливость тождества (10):

$$x^{a_i}(\overline{x^{a_j}}\vee A)\equiv x^{a_i}(x^{a_1}\vee x^{a_2}\vee ...\vee x^{a_i}\vee ...$$
$$\vee x^{a_{j-1}}\vee x^{a_{j+1}}\vee ...\vee x^{a_k}\vee A)\equiv x^{a_i}\vee x^{a_i}A\equiv x^{a_i}\,.$$

Рассмотрим пример, иллюстрирующий применение только что приведенных тождеств. Пусть $A=\{a, b, c\}$, требуется упростить формулу

$$f\equiv(\overline{x_1^a}\,\overline{x_1^b}\vee\overline{x_1^c}\vee\overline{x_2^a})\,x_1{}^b x_2{}^c,$$

предварительно исключив из нее отрицания. Применяя тождество (10), непосредственно получаем искомый результат $f \equiv x_1^{\,b} x_2^{\,c}$. Без использования законов поглощения отрицания тот же результат достигается более длинным путем:

$$f \equiv ((x_1^{\,b} \vee x_1^{\,c})(x_1^{\,a} \vee x_1^{\,c}) \vee x_1^{\,a} \vee x_1^{\,b} \vee x_2^{\,b} \vee x_2^{\,c}) x_1^{\,b} x_2^{\,c} \equiv$$

$$\equiv (x_1^{\,c} \vee x_1^{\,a} \vee x_1^{\,b} \vee x_2^{\,b} \vee x_2^{\,c}) x_1^{\,b} x_2^{\,c} \equiv x_1^{\,b} x_2^{\,c} \vee x_1^{\,b} x_2^{\,c} \equiv x_1^{\,b} x_2^{\,c}.$$

Для операции отрицания также справедливы следующие тождества: *закон двойного отрицания*

$$\overline{\overline{A}} \equiv A, \qquad (11)$$

закон исключенного третьего

$$A \vee \overline{A} \equiv 1, \qquad (12)$$

закон противоречия

$$A\overline{A} \equiv 0. \qquad (13)$$

Эти тождества по форме совпадают с одноименными законами алгебры логики. Тождества (11) – (13) можно получить из ранее выведенных тождеств.

Докажем справедливость закона двойного отрицания индукцией по длине формулы. Он выполняется для формул 0 и 1, а также для всевозможных узнаваний букв: $\overline{\overline{0}} \equiv \overline{1} \equiv 0$, $\overline{\overline{1}} \equiv \overline{0} \equiv 1$,

$$\overline{\overline{x^{a_i}}} \equiv \overline{x^{a_1} \vee x^{a_2} \vee \ldots \vee x^{a_{i-1}} \vee x^{a_{i+1}} \vee \ldots \vee x^{a_k}} \equiv$$

$$\equiv \overline{x^{a_1}}\,\overline{x^{a_2}} \ldots \overline{x^{a_{i-1}}}\,\overline{x^{a_{i+1}}} \ldots \overline{x^{a_k}} \equiv$$

$$(x^{a_2} \vee \ldots \vee x^{a_i} \vee \ldots \vee x^{a_k})(x^{a_1} \vee x^{a_3} \vee \ldots \vee$$

$$\vee x^{a_i} \vee \ldots \vee x^{a_k}) \ldots (x^{a_1} \vee \ldots \vee x^{a_i} \vee x^{a_{i+2}} \vee \ldots \vee$$

$$\vee x^{a_k}) \ldots (x^{a_1} \vee \ldots \vee x^{a_i} \vee \ldots \vee x^{a_{k-1}}) \equiv x^{a_i}.$$

Предположим, что закон двойного отрицания справедлив для формул A и B, и установим, что он выполняется также для формул $A \vee B$ и AB:

$$\overline{\overline{A \vee B}} \equiv \overline{\overline{A}\,\overline{B}} \equiv \overline{\overline{A}} \vee \overline{\overline{B}} \equiv A \vee B,$$

$$\overline{\overline{AB}} \equiv \overline{\overline{A} \vee \overline{B}} \equiv \overline{\overline{A}}\,\overline{\overline{B}} \equiv AB.$$

Таким образом, закон двойного отрицания справедлив для всех формул.

Докажем тождество (12). Оно выполняется для формул 0 и 1, а также для всех узнаваний букв: $0 \vee \overline{0} \equiv 0 \vee 1 \equiv 1$, $1 \vee \overline{1} \equiv 1 \vee 0 \equiv 1$,

$$x^{a_i} \vee \overline{x^{a_i}} \equiv$$

$$\equiv x^{a_i} \vee x^{a_1} \vee x^{a_2} \vee \ldots \vee x^{a_{i-1}} \vee x^{a_{i+1}} \vee \ldots \vee x^{a_k} \equiv 1.$$

Если тождество (40) справедливо для формул A и B, то оно справедливо и для формул $A \vee B$ и AB:

$$A \vee B \vee \overline{A \vee B} \equiv A \vee B \vee \overline{A}\;\overline{B} \equiv$$

$$\equiv (A \vee A)(A \vee \overline{B}) \vee (B \vee \overline{A})(B \vee \overline{B}) \equiv A \vee \overline{B} \vee B \vee \overline{A} \equiv 1,$$

$$AB \vee \overline{AB} \equiv AB \vee \overline{A} \vee \overline{B} \equiv (A \vee \overline{A})(B \vee \overline{A})(A \vee \overline{B})(B \vee \overline{B}) \equiv$$

$$\equiv B \vee \overline{A} \vee A \vee \overline{B} \equiv 1.$$

Аналогично доказывается закон противоречия.

Итак, мы видим, что в алгебре конечных предикатов наряду с операциями дизъюнкции и конъюнкции существует операция отрицания со всеми свойствами, которыми наделяет ее булева алгебра (тождества (4), (5), (7), (8), (11) – (13)). Поэтому алгебру конечных предикатов можно рассматривать как разновидность булевой алгебры. Основным множеством в ней служит система всех k-ичных n-местных предикатов с алфавитом букв $\{a_1, a_2, \ldots, a_k\}$ и алфавитом переменных $\{x_1, x_2, \ldots, x_n\}$, роль нуля выполняет тождественно ложный предикат, роль единицы – тождественно истинный предикат, в роли базисных операций выступают дизъюнкция, конъюнкция и отрицание.

В качестве системы тождеств алгебры конечных предикатов, задающей в ней структуру булевой алгебры (и только эту структуру), можно использовать, к примеру, следующий набор аксиом [1]: (10), (4), (6), (8), (39), (32) и тождество:

$$(A \vee \overline{A})B \equiv B. \qquad (14)$$

Подчеркнем еще раз, что алгебра конечных предикатов не есть только булева алгебра, она имеет и свои специфические тождества: закон истинности (18) [1], закон ложности (19) [1] и закон отрицания (6), не вытекающие из аксиом булевой алгебры.

Законы истинности и ложности можно записать в модифицированном виде без использования символов 0 и 1:

$$x^{a_1} \vee x^{a_2} \vee \ldots \vee x^{a_k} \equiv A \vee \overline{A}, \qquad (15)$$

$$x^{a_i}\, x^{a_i} \equiv A\overline{A}, \ (i \neq j) \qquad (16)$$

Тождество (16) может быть выведено из тождеств булевой алгебры и тождеств (6), (15). Действительно, пусть $i \neq j$. Тогда:

$$x^{a_i}\, x^{a_j} \equiv \overline{\overline{x^{a_i} x^{a_j}}} \equiv \overline{\overline{x^{a_i}} \vee \overline{x^{a_j}}} \equiv$$

$$\equiv (x^{a_1} \vee x^{a_2} \vee \ldots \vee x^{a_{i-1}} \vee x^{a_{i+1}} \vee \ldots \vee x^{a_k} \vee x^{a_1} \vee$$

$$\vee x^{a_2} \vee \ldots \vee x^{a_{j-1}} \vee x^{a_{j+1}} \vee \ldots \vee x^{a_k}) \equiv (x^{a_1} \vee x^{a_2} \vee \ldots \vee x^{a_k}) \equiv$$

$$\equiv \overline{A \vee \overline{A}} \equiv A\overline{A}.$$

Таким образом, при наличии операции отрицания, заданной аксиоматически тождествами (4)-(6), законы ложности можно исключить из числа аксиом алгебры конечных предикатов.

Заметим, что введение операции отрицания в алгебре конечных предикатов не расширяет ее выразительных возможностей. Ведь мы знаем, что алгебра конечных предикатов полна и без операции отрицания. Таким образом, введением операции отрицания достигается лишь консервативное расширение [3] алгебры конечных предикатов. В дальнейшем мы не раз будем вводить в алгебре конечных предикатов новые операции. Хотя они и не расширяют выразительных возможностей алгебры

конечных предикатов, однако делают более гибким ее язык, повышают уровень его абстрактности.

Рассмотрим понятие *инверсного предиката*, которое нам понадобится при введении совершенной конъюнктивной нормальной формы. Предикат $g(x_1, x_2,..., x_n)$ назовем инверсным по отношению к предикату $f(x_1, x_2,..., x_n)$, если

$$g(x_1, x_2,..., x_n) \equiv \overline{f}\,(x_1, x_2,..., x_n).$$

Предикат, инверсный предикату f, будем записывать в виде f^*, так что:

$$f^*(x_1, x_2,..., x_n) \equiv \overline{f}\,(x_1, x_2,..., x_n). \qquad (17)$$

Таблица инверсного предиката может быть получена из таблицы исходного предиката заменой в ее последней строке значений предиката на обратные: 0 на 1 и 1 на 0. Для каждого предиката существует инверсный ему предикат. Предикат, инверсный инверсному предикату, совпадает с исходным предикатом:

$$f^{**}(x_1, x_2,..., x_n) \equiv f(x_1, x_2,..., x_n). \qquad (18)$$

Система предикатов, инверсных всевозможным предикатам, совпадает с системой всех предикатов (имеются в виду предикаты, соответствующие вполне определенным наборам множествам букв и переменных). По формуле, представляющей некоторый конечный предикат, нетрудно построить формулу для инверсного предиката. Для этого нужно в исходной формуле заменить все знаки конъюнкции знаками дизъюнкции, знаки дизъюнкции – знаками конъюнкции, знаки 0 – знаками 1 и знаки 1 – знаками 0. Следует также ввести знаки отрицания над теми узнаваниями букв, где они до этого отсутствовали, и исключить знаки отрицания у тех узнаваний букв, над которыми они прежде присутствовали. Например, пусть

$$f \equiv x_1^a x_2^b \vee \overline{x_3^c}\,,$$

тогда

$$f^* \equiv (\overline{x_1^a} \vee \overline{x_2^b}) x_3^c\,.$$

Рассмотрим теперь конъюнктивное разложение предиката. Для этого применим тождество (1) к предикату $f^*(x_1, x_2,..., x_n)$:

$$f^*(x_1, x_2,..., x_l, x_{l+1},..., x_n) \equiv$$

$$\equiv \bigvee_{(\sigma_1,\sigma_2,...,\sigma_l)} x_1^{\sigma_1} x_2^{\sigma_2} ... x_l^{\sigma_l} f^*(\sigma_1, \sigma_2,..., \sigma_l, x_{l+1},..., x_n).$$

Действуем отрицанием на левую и правую части полученного тождества, используя законы де Моргана и учитывая, что $\overline{f^*} \equiv f$. В результате приходим к *теореме о конъюнктивном разложении*, формулируемой следующим образом.

Любой конечный предикат $f(x_1, x_2,..., x_n)$ может быть представлен в виде:

$$f(x_1, x_2,..., x_l, x_{l+1},..., x_n) \equiv$$
$$\bigvee_{(\sigma_1,\sigma_2,...,\sigma_l)} \overline{x_1^{\sigma_1}} \vee \overline{x_2^{\sigma_2}} \vee ... \vee \overline{x_l^{\sigma_l}} \qquad (19)$$
$$\vee f(\sigma_1, \sigma_2,..., \sigma_l, x_{l+1},..., x_n)).$$

Представление предиката $f(x_1, x_2,..., x_n)$ в виде правой части тождества (19) назовем его *конъюнктивным разложением* по переменным $x_1, x_2,..., x_n$. Буквенные переменные $\sigma_1, \sigma_2,..., \sigma_l$ играют роль индексов при образовании многократной конъюнкции. Запись $(\sigma_1, \sigma_2,..., \sigma_l)$ под знаком конъюнкции означает, что логическое перемножение ведется по всевозможным наборам индексов $\sigma_1, \sigma_2,..., \sigma_l$. Таким образом, в правой части тождества (19) присутствуют k^l конъюнктивных членов.

Из теоремы о конъюнктивном разложении вытекают следующие два *следствия*.

Следствие 1. *Любой конечный предикат $f(x_1, x_2, ..., x_n)$ может быть представлен в виде*

$$f(x_1, x_2,..., x_n) \equiv (\overline{x_1^{a_1}} \vee f(a_1, x_2,..., x_n)) \wedge$$
$$\wedge(\overline{x_1^{a_2}} \vee f(a_2, x_2,..., x_n))...(\overline{x_1^{a_k}} \vee \qquad (20)$$
$$\vee f(a_k, x_2,..., x_n)).$$

Следствие 2. *Любой конечный предикат $f(x_1, x_2,..., x_n)$ может быть представлен в виде*

$$f(x_1, x_2,..., x_n) \equiv \bigvee_{f(\sigma_1,\sigma_2,...,\sigma_n)=0} (\overline{x_1^{\sigma_1}} \vee \overline{x_2^{\sigma_2}} \vee ... \vee \overline{x_n^{\sigma_n}}). \qquad (21)$$

Тождества (20) и (21) находим, полагая в виде (19) соответственно $l=1$ и $l=n$. Запись $f(\sigma_1, \sigma_2,..., \sigma_n)=0$ в (21) под знаком конъюнкции означает, что логическое перемножение ведется только по тем наборам индексов $(\sigma_1, \sigma_2,..., \sigma_l)$, которые обращают предикат f в 0. Если предикат f тождественно истинный, то, согласно теореме о конъюнктивном разложении, в правой части тождества (21) нужно ставить 1.

Представление предиката f в виде формулы, получаемой в результате исключения из первой части тождества (21) знаков отрицания с помощью законов отрицания, назовем *совершенной конъюнктивной нормальной формой* предиката f (*СКНФ*). Рассмотрим пример получения СКНФ предиката. Пусть $A=\{a, b, c\}$, $B=\{x_1, x_2\}$. Найти СКНФ предиката:

$$f(x_1, x_2) = x_1^a x_2^b \vee x_1^a x_2^c \vee x_1^b x_2^a\,.$$

Значения $f(\sigma_1, \sigma_2)$ равны 0 для всех наборов (σ_1, σ_2), кроме (a, b), (a, c) и (b, a). По формуле (49) находим:

$$f(x_1, x_2) \equiv (\overline{x_1^a} \vee \overline{x_2^a})(\overline{x_1^b} \vee \overline{x_2^b})(\overline{x_1^b} \vee \overline{x_2^c}) \wedge$$
$$\wedge(\overline{x_1^c} \vee \overline{x_2^a})(\overline{x_1^c} \vee \overline{x_2^b})(\overline{x_1^c} \vee \overline{x_2^c}).$$

Избавляясь от отрицаний с помощью тождеств (34), выводим СКНФ предиката f:

$$f(x_1, x_2) = (x_1^b \vee x_1^c \vee x_2^b \vee x_2^c)(x_1^a \vee x_1^c \vee x_2^a \vee x_2^c) \wedge$$
$$\wedge(x_1^a \vee x_1^c \vee x_2^a \vee x_2^b)(x_1^a \vee x_1^b \vee x_2^b \vee x_2^c) \wedge$$
$$\wedge(x_1^a \vee x_1^b \vee x_2^a \vee x_2^c)(x_1^a \vee x_1^b \vee x_2^a \vee x_2^b).$$

Как известно, в алгебре логики понятия СДНФ и СКНФ двойственны друг другу. Это значит, что

определение понятия СКНФ в алгебре логики может быть получено простой заменой в определении СДНФ слов “конъюнкция”, “дизъюнкция”, “истина”, “ложь” соответственно словами “дизъюнкция”, “конъюнкция”, “ложь”, “истина”. В алгебре конечных предикатов было бы неправильно определять СКНФ по аналогии с СДНФ как конъюнкцию таких дизъюнкций узнаваний букв, в которые каждая переменная входит по одному разу. На самом деле в конъюнктивные члены СКНФ каждая переменная входит k-1 раз. В алгебре конечных предикатов только при k=2 имеет место двойственность понятий СДНФ и СКНФ такого же типа, как и в алгебре логики.

Введем понятия элементарной дизъюнкции, конституэнты нуля и конъюнктивной нормальной формы. Эти понятия не являются двойственными понятиями элементарной конъюнкции, конституэнты единицы и ДНФ, как это имеет место в алгебре логики. *Элементарной дизъюнкцией* назовем такую дизъюнкцию узнаваний букв, в которую каждое узнавание буквы может входить не более одного раза и ни одна из переменных не может входить не более k-1 раз. Элементарной дизъюнкцией, не содержащей ни одного дизъюнктивного члена, будем считать формулу 0. Иными словами, никакая переменная не может быть представлена в элементарной дизъюнкции всеми своими узнаваниями, хотя бы одно из узнаваний должно отсутствовать. Примерами элементарных дизъюнкций при $A=\{a, b, c\}$, $B=\{x_1, x_2, x_3\}$ могут служить формулы:

$$x_1^a \vee x_2^b;\ \ x_1^a \vee x_1^c \vee x_2^b;\ \ x_2^b \vee x_2^c \vee x_3^a \vee x_3^c .$$

Формула

$$x_1^a \vee x_1^b \vee x_1^c \vee x_2^a$$

не является элементарной дизъюнкцией, так как в нее входят все узнавания буквы для переменной x_1. Такая формула выражает тождественно истинный предикат.

Любую конъюнкцию произвольного числа различных элементарных дизъюнкций назовем *конъюнктивной нормальной формой (КНФ)* предиката. В качестве КНФ не содержащей ни одного конъюнктивного члена, примем формулу 1. Пример КНФ:

$$(x_1^a \vee x_2^b)(x_1^a \vee x_1^c \vee x_2^b)(x_2^b \vee x_2^c \vee x_3^a \vee x_3^c) .$$

Любую элементарную дизъюнкцию, в которой встречаются все переменные алгебры конечных предикатов, причем каждая из них входит k-1 раз в составе узнаваний букв с различным показателями, назовем *конституэнтой нуля*. Условимся в конституэнте нуля все узнавания букв располагать в порядке роста номеров переменных, а для одинаковых переменных – в порядке роста номеров букв, играющих роль показателей узнаваний буквы. Пример конституэнты нуля (в той же алгебре):

$$x_1^a \vee x_1^b \vee x_2^b \vee x_2^c \vee x_3^a \vee x_3^c .$$

Используя знак отрицания, можно записывать конституэнты нуля в более компактной форме, близкой к форме записи конституэнты единицы. Конституэнту нуля, приведенную в предыдущем примере, запишем так:

$$\overline{x_1^c} \vee \overline{x_2^a} \vee \overline{x_3^b} .$$

Последняя форма записи конституэнты нуля инверсна форме записи конституэнты единицы: в ней вместо знаков конъюнкции стоят знаки дизъюнкции, а над узнаваниями букв появились знаки отрицания.

Теперь мы можем дать еще одно, равносильное первому, определение *совершенной конъюнктивной нормальной формы* как конъюнкции произвольного числа различных конституэнт нуля. Осуществляя нумерацию конституэнт нуля и СКНФ, можно показать, что каждому k-ичному n-местному предикату соответствует одна СКНФ. В качестве СКНФ тождественно истинного предиката принимаем формулу 1.

Кроме конъюнкции и дизъюнкции предикатов, рассмотренных ранее, введем в алгебре конечных предикатов еще одну двухместную операцию – *импликацию предикатов* $\supset$, определив ее следующим равенством:

$$A \supset B \underset{def}{\equiv} \overline{A} \vee B . \qquad (22)$$

Буквы A и B обозначают произвольные формулы алгебры конечных предикатов. Знак $\underset{def}{\equiv}$ обозначает равенство предикатов по определению. Операцию $\supset$ будем считать младшей по отношению к операциям $\wedge$ и $\vee$. Это значит, что при отсутствии в формуле скобок, регулирующих порядок выполнения операций, операция импликации должна выполнятся после выполнения операций конъюнкции и дизъюнкции.

Свойства импликации предикатов совпадают со свойствами импликации высказываний [3]. Выпишем наиболее важные из них – *рефлексивность импликации*:

$$A \supset A \equiv 1, \qquad (23)$$

транзитивность импликации:

$$(A \supset B)(B \supset C) \supset (A \supset C) \equiv 1, \qquad (24)$$

свойства логических констант:

$$0 \supset A \equiv 1, \qquad (25)$$

$$1 \supset A \equiv A, \qquad (26)$$

$$A \supset 0 \equiv \overline{A}, \qquad (27)$$

$$A \supset 1 \equiv 1, \qquad (28)$$

закон дедукции:

$$A(A \supset B) \supset B \equiv 1, \qquad (29)$$

закон контрапозиции:

$$A \supset B \equiv \overline{B} \supset \overline{A}, \qquad (30)$$

закон импортации:

$$(A \supset (B \supset C)) \supset (AB \supset C) \equiv 1, \qquad (31)$$

закон экспортации:

$$(AB \supset C) \supset (A \supset (B \supset C)) \equiv 1, \qquad (32)$$

закон приведения к абсурду:

$$A\overline{A} \supset B \equiv 1. \qquad (33)$$

Используя операцию импликации, тождество (19) можно записать без знаков отрицания:

$$\begin{gathered} f(x_1, x_2,\ldots, x_l, x_{l+1},\ldots, x_n) \equiv \\ \equiv \bigwedge_{(\sigma_1,\sigma_2,\ldots,\sigma_l)} x_1^{\sigma_1} x_2^{\sigma_2} \ldots x_l^{\sigma_l} \supset \\ \supset f(\sigma_1, \sigma_2,\ldots, \sigma_l, x_{l+1},\ldots,x_n)). \end{gathered} \qquad (34)$$

В соответствии с этим первое и второе следствия теоремы о конъюнктивном разложении запишутся в виде:

$$\begin{gathered} f(x_1, x_2,\ldots, x_n) \equiv \\ \equiv (x_1^{a_1} \supset f(a_1, x_2,\ldots, x_n)) \wedge \\ \wedge (x_1^{a_2} \supset f(a_2, x_2,\ldots, x_n)) \wedge \ldots \\ \wedge (x_1^{a_k} \supset f(a_k, x_2,\ldots, x_n)). \end{gathered} \qquad (35)$$

$$f(x_1, x_2,\ldots, x_n) \equiv (\bigwedge_{f(\sigma_1,\sigma_2,\ldots,\sigma_n)=0} (x_1^{\sigma_1} x_2^{\sigma_2} \ldots x_n^{\sigma_n} \supset 0). \qquad (36)$$

Последнее тождество называется *импликативным разложением предиката*. Оно показывает, что любой конечный предикат может быть представлен в виде суперпозиции операций конъюнкции и импликации, действующих на всевозможные узнавания букв, и на тождественно ложный предикат 0. Вместе с тем, как указывалось выше, при $k \geq 2$ предикат 0 можно выразить в виде конъюнкции некоторых узнаваний букв, например, $x_1^{a_1} x_1^{a_2} = 0$. Таким образом, мы приходим к еще одной полной алгебре конечных предикатов. Ее базис составляют операции конъюнкции и импликации и всевозможные узнавания букв. Чтобы иметь возможность различать обе введенные алгебры, первую назовем *дизъюнктивной алгеброй конечных предикатов*, а вторую – *импликативной*. Импликативная алгебра может быть получена из дизъюнктивной заменой в ее базисе операции дизъюнкции операцией импликации, и наоборот. Кроме двух найденных, существует множество других алгебр конечных предикатов. В частности, любой набор операций, удовлетворяющий условиям теоремы Поста [4], вместе со всевозможными узнаваниями букв можно принять в качестве базиса полной алгебры конечных предикатов. В роли полной системы элементарных операций также подходит любая система операций, через которые выражаются операции конъюнкции и дизъюнкции или операции конъюнкции и импликации. По-видимому, существуют и другие базисы, задающие полные алгебры конечных предикатов. Еще предстоит сформулировать критерий полноты для алгебр конечных предикатов.

В данной статье в качестве языка для записи конечных предикатов принята дизъюнктивная алгебра и различные ее консервативные расширения. Удобство дизъюнктивной алгебры конечных предикатов состоит в том, что на ее языке кратко и изящно записываются законы истинности и ложности. Эти законы в любой алгебре конечных предикатов будут играть особую роль. По существу, они представляют собой требования, выполнение которых необходимо и достаточно для корректного введения переменных на конечных множествах. Закон истинности задает область изменения переменной, а закон ложности обеспечивает попарное различие всех элементов множества, на котором заданна переменная. В любой другой алгебре законы истинности и ложности будут записываться в виде гораздо более громоздких выражений. В дальнейшем дизъюнктивную алгебру и ее консервативные расширения будем называть просто *алгеброй конечных предикатов*.

3. Дизъюнктивная минимизация формул алгебры предикатов

Существует целое множество дизъюнктивных и конъюнктивных нормальных форм, соответствующих одному и тому же конечному предикату. Представляет интерес следующая задача: выбрать из этого множества такую форму, в которую входит наименьшее число узнаваний букв. Назовем эту задачу *канонической задачей минимизации* формул алгебры конечных предикатов. Умение минимизировать формулы, как будет показано ниже, полезно при решении уравнений теории интеллекта, а также при построении схем, реализующих функции интеллекта. Дизъюнктивные и конъюнктивные нормальные формы, получаемые в результате канонической минимизации, назовем *минимальными ДНФ и КНФ*. Проблема канонической минимизации в алгебре конечных предикатов имеет много общего с одноименной проблемой в алгебре логики. Все излагаемые в этой статье методы можно рассматривать как обобщение известных в алгебре логики методов канонической минимизации [5]:

а) ***Отыскание минимальной дизъюнктивной нормальной формы.*** Здесь мы наметим последовательность действий, которые должны быть выполнены при нахождении минимальной ДНФ. Предикат g назовем *импликантой предиката f*, если на любом наборе значений аргументов, для которого $g=1$, имеем также $f=1$. Будем говорить, что импликанта g накрывает своими единицами единицы предиката f. Элементарную конъюнкцию g назовем *собственной частью* элементарной конъюнкции f, если g может быть получена из f выбрасыванием

некоторых узнаваний букв. Например, элементарная конъюнкция $x_1^a x_3^b$ есть собственная часть элементарной конъюнкции $x_1^a x_2^c x_3^b$. Назовем *простой импликантой* предиката f любую элементарную конъюнкцию, обладающую следующими свойствами: она есть импликанта предиката f и никакая ее собственная часть не является импликантой предиката f.

Дизъюнкция любого числа импликант конечного предиката есть импликанта этого предиката. Систему S импликант конечного предиката f назовем *полной*, если любая единица из таблицы значений предиката f накрывается единицами хотя бы одной импликанты системы S. Тому или иному конечному предикату может соответствовать, вообще говоря, несколько различных полных систем импликант. Дизъюнкция всех импликант полной системы импликант каждого конечного предиката выражает собой этот предикат. Система всех простых импликант любого конечного предиката является его полной системой. Дизъюнкция всех простых импликант конечного предиката выражает собой этот предикат; назовем ее *сокращенной дизъюнктивной нормальной формой* данного конечного предиката.

Во многих случаях (однако далеко не всегда) сокращенная дизъюнктивная нормальная форма представляет собой более экономное средство записи конечных предикатов, чем СДНФ. Однако сокращенная форма, как правило, не совпадает с минимальной дизъюнктивной нормальной формой, сложнее ее. *Дизъюнктивным ядром* конечного предиката назовем множество всех таких его простых импликант, исключение каждой из которых из системы всех простых импликант делает эту систему неполной. Элементы дизъюнктивного ядра входят в состав любой полной системы простых импликант. Систему простых импликант конечного предиката назовем *приведенной*, если она полна и никакое ее собственное подмножество не является полной системой. Дизъюнкцию всех простых импликант приведенной системы назовем *тупиковой дизъюнктивной нормальной формой* предиката. Отметим, что предикат может иметь много тупиковых дизъюнктивных нормальных форм.

Любая минимальная дизъюнктивная нормальная форма является, вместе с тем, и тупиковой ДНФ. Многие конечные предикаты имеют несколько различных минимальных ДНФ, содержащих одинаковое число узнаваний букв. Выбирая из числа тупиковых ДНФ одну с наименьшим числом узнаваний букв, получаем минимальную ДНФ предиката. Сформулированные понятия и положения позволяют наметить ход действий, лежащих в основе всех описываемых ниже методов минимизации формул алгебры конечных предикатов. Вначале нужно отыскать все простые импликанты заданного конечного предиката и составить из них сокращенную ДНФ. Затем следует найти все тупиковые ДНФ и из их числа выбрать минимальную дизъюнктивную нормальную форму конечного предиката.

Ниже рассматриваются три алгоритма минимизации формул алгебры конечных предикатов. Алгоритмы сопровождаются примерами. В частном случае при k=2 они превращаются в известные алгоритмы канонической минимизации формул алгебры логики Квайна-Мак-Класки, Порецкого-Блейка и Нельсона [6].

б) ***Обобщение метода Квайна-Мак-Класки.*** Рассмотрим метод дизъюнктивной минимизации формул алгебры конечных предикатов, обобщающий известный метод Квайна-Мак-Класки минимизации формул алгебры логики. Исходной информацией при минимизации служит совершенная дизъюнктивная нормальная форма предиката, для которого отыскивается минимальная ДНФ. Рассмотрим пример. Пусть $A=\{a, b, c\}$, $B=\{x_1, x_2, x_3\}$. Предикат задан следующей СДНФ:

$$f\equiv x_1^a x_2^a x_3^a \vee x_1^a x_2^a x_3^b \vee x_1^a x_2^a x_3^c \vee x_1^a x_2^b x_3^b \vee$$
$$\vee x_1^a x_2^c x_3^a \vee x_1^a x_2^c x_3^b \vee x_1^a x_2^c x_3^c \vee x_1^b x_2^b x_3^a \vee$$
$$x_1^b x_2^b x_3^b \vee x_1^b x_2^b x_3^c \vee x_1^c x_2^a x_3^b \vee x_1^c x_2^b x_3^b \vee$$
$$\vee x_1^c x_2^c x_3^b \vee x_1^c x_2^c x_3^c.$$

Ниже описывается алгоритм минимизации, включающий в себя 9 шагов.

1) Всюду, где это возможно, применяем *операцию неполного дизъюнктивного склеивания*, основанную на тождестве:

$$Ax^{a_1} \vee Ax^{a_2} \vee \ldots \vee Ax^{a_k} \equiv$$
$$\equiv Ax^{a_1} \vee Ax^{a_2} \vee \ldots \vee Ax^{a_k} \vee A. \qquad (37)$$

Здесь A – некоторая элементарная конъюнкция, x – одна из буквенных переменных. Операция неполного дизъюнктивного склеивания состоит в дописывании к исходной ДНФ в виде дополнительных дизъюнктивных членов всевозможных элементарных конъюнкций, которые можно получить переходом от левой части тождества (37) к правой. Операцию неполного дизъюнктивного склеивания сперва применяем к конституэнтам единицы исходной СДНФ, затем – к элементарным конъюнкциям, состоящим из n-1 узнаваний букв, n-2 узнаваний букв и т.д. и, наконец, к элементарным конъюнкциям, состоящим из одного узнавания буквы. В нашем примере имеем:

$$f\equiv x_1^a x_2^a x_3^a \vee x_1^a x_2^a x_3^b \vee x_1^a x_2^a x_3^c \vee x_1^a x_2^b x_3^b \vee x_1^a x_2^c x_3^a \vee$$
$$\vee x_1^a x_2^c x_3^b \vee x_1^a x_2^c x_3^c \vee x_1^b x_2^b x_3^a \vee x_1^b x_2^b x_3^b \vee x_1^b x_2^b x_3^c \vee$$
$$\vee x_1^c x_2^a x_3^b \vee x_1^c x_2^b x_3^b \vee x_1^c x_2^c x_3^b \vee x_1^c x_2^c x_3^c \vee x_1^a x_2^a \vee$$
$$\vee x_1^a x_2^c \vee x_1^b x_2^b \vee x_1^a x_3^b \vee x_1^c x_3^b \vee x_2^b x_3^b.$$

2) Применяем к дизъюнктивным членам полученной формулы всюду, где возможно, операцию элементарного дизъюнктивного поглощения

$$Ax^{\sigma} \vee A \equiv A, \qquad (38)$$

где x^{σ} – некоторый предикат узнавания буквы. В результате получаем сокращенную ДНФ. В примере имеем:

$$f \equiv x_1{}^a x_2{}^a \vee x_1{}^a x_3{}^b \vee x_1{}^a x_2{}^c \vee x_2{}^b x_3{}^b \vee x_1{}^b x_2{}^b \vee x_1{}^c x_3{}^b \vee x_1{}^c x_2{}^c x_3{}^c.$$

3) Составляем импликантную таблицу. *Импликантная таблица* конечного предиката f представляет собой матрицу, строки которой обозначены всевозможными простыми импликантами предиката f, а столбцы – конституэнтами единицы того же предиката. Каждая ячейка таблицы соответствует паре предикатов, один из них представлен конституэнтой единицы, другой – некоторой элементарной конъюнкцией. Если элементарная конъюнкция, соответствующая некоторой строке таблицы, является импликантой для конституэнты единицы, соответствующей некоторому столбцу, то в ячейку, образуемую пересечением строки и столбца, заносим звездочку, в противном случае ячейку оставляем незаполненной. Импликантная таблица, получаемая в примере, представлена в табл. 7.

Таблица 7

Простая импликанта	$x_1{}^a x_2{}^a x_3{}^a$	$x_1{}^a x_2{}^a x_3{}^c$	$x_1{}^a x_2{}^a x_3{}^c$	$x_1{}^a x_2{}^b x_3{}^b$	$x_1{}^a x_2{}^c x_3{}^a$	$x_1{}^a x_2{}^b x_3{}^b$	$x_1{}^a x_2{}^c x_3{}^c$	$x_1{}^b x_2{}^b x_3{}^a$	$x_1{}^b x_2{}^b x_3{}^b$	$x_1{}^b x_2{}^b x_3{}^c$	$x_1{}^c x_2{}^b x_3{}^b$	$x_1{}^c x_2{}^b x_3{}^b$	$x_1{}^c x_2{}^c x_3{}^b$	$x_1{}^c x_2{}^c x_3{}^c$
$x_1{}^a x_2{}^a$	*	*	*											
$x_1{}^a x_3{}^b$		*		*		*								
$x_1{}^a x_2{}^c$					*	*	*							
$x_2{}^b x_3{}^b$				*					*			*		
$x_1{}^b x_2{}^b$								*	*	*				
$x_1{}^c x_3{}^b$											*	*	*	
$x_1{}^c x_2{}^c x_3{}^c$														*

4) По импликантной таблице находим дизъюнктивное ядро предиката, для чего находим те столбцы таблицы, в которых содержится по одной звездочке. Соответствующие звездочкам простые импликанты образуют дизъюнктивное ядро предиката. В примере дизъюнктивное ядро предиката представляет собой следующее множество:

$$\{x_1{}^a x_2{}^a,\ x_1{}^a x_2{}^c,\ x_1{}^b x_2{}^b,\ x_1{}^c x_3{}^b,\ x_1{}^c x_2{}^c x_3{}^c\}.$$

5) По импликантной таблице отыскиваем систему всех конституэнт единицы предиката, которые не накрываются единицами его дизъюнктивного ядра. С этой целью отмечаем все столбцы, в которых содержатся звездочки, соответствующие простым импликантам, вошедшим в состав дизъюнктивного ядра. Столбцы, оставшиеся неотмеченными, соответствуют искомым конституэнтам единицы. В нашем примере разыскиваемая система имеет в своем составе единственную конституэнту единицу $\{x_1{}^a x_2{}^b x_3{}^b\}$.

6) По импликантной таблице методом полного перебора отыскиваем все приведенные системы простых импликант, накрывающих своими единицами все конституэнты единицы системы, найденной на пятом шаге алгоритма. В примере имеем две такие системы: $\{x_1{}^a x_3{}^b\}$, $\{x_2{}^b x_3{}^b\}$, каждая из которых состоит из одной конституэнты единицы.

7) Объединяя дизъюнктивное ядро предиката с каждой из систем, полученных на шестом шаге алгоритма, формируем все приведенные системы простых импликант предиката. В нашем примере имеем две системы:

$$\{x_1{}^a x_2{}^a,\ x_1{}^a x_2{}^c,\ x_1{}^b x_2{}^b,\ x_1{}^c x_3{}^b,\ x_1{}^c x_2{}^c x_3{}^c,\ x_1{}^a x_3{}^b\},$$

$$\{x_1{}^a x_2{}^a,\ x_1{}^a x_2{}^c,\ x_1{}^b x_2{}^b,\ x_1{}^c x_3{}^b,\ x_1{}^c x_2{}^c x_3{}^c,\ x_2{}^b x_3{}^b\}.$$

8) Строим все тупиковые ДНФ предиката. В примере имеем две такие формы:

$$x_1{}^a x_2{}^a \vee x_1{}^a x_2{}^c \vee x_1{}^b x_2{}^b \vee x_1{}^c x_3{}^b \vee x_1{}^c x_2{}^c x_3{}^c \vee x_1{}^a x_3{}^b,$$

$$x_1{}^a x_2{}^a \vee x_1{}^a x_2{}^c \vee x_1{}^b x_2{}^b \vee x_1{}^c x_3{}^b \vee x_1{}^c x_2{}^c x_3{}^c \vee x_1{}^b x_3{}^b.$$

9) Из числа тупиковых ДНФ выбираем минимальную ДНФ, имеющую наименьшее число узнаваний букв. В нашем примере обе тупиковые ДНФ имеют одинаковое число узнаваний букв — 13, любая из них может быть принята в качестве минимальной ДНФ. Исходная СДНФ предиката f имеет 42 узнавания буквы. В результате минимизации число узнаваний букв в формуле сократилось более чем в три раза.

в) ***Обобщение методов Порецкого-Блейка и Нельсона.*** Ниже приводится описание алгоритма минимизации формул алгебры конечных предикатов, обобщающего алгоритм Порецкого-Блейка дизъюнктивной минимизации формул алгебры логики. Исходной информацией при минимизации для этого алгоритма служит произвольно выбранная ДНФ предиката, для которого отыскивается минимальная ДНФ. Рассмотрим пример. Пусть $A=\{a, b, c\}$. Предикат f задан следующей ДНФ:

$$f \equiv x_1^a x_2^a \vee x_1^a x_2^b \vee x_1^a x_3^a \vee x_1^b x_2^c x_3^a \vee x_1^c x_2^c x_3^a.$$

1) Применяем к дизъюнктивным членам исходной ДНФ всюду, где возможно, *операцию группировки узнаваний букв*:

$$Ax^{\sigma_1} \vee Ax^{\sigma_2} \vee \ldots \vee Ax^{\sigma_r} \equiv A(x^{\sigma_1} \vee x^{\sigma_2} \vee \ldots \vee x^{\sigma_r}). \qquad (39)$$

Указанная операция не используется в методе Порецкого-Блейка при минимизации формул алгебры логики, она становится необходимой только в алгебре конечных предикатов при дизъюнктивной минимизации в случае, когда $k>2$. В результате выполнения операции группировки узнавания букв получаем дизъюнкцию некоторых скобочных форм. В нашем примере:

$$f \equiv (x_1^a) x_2^a \vee (x_1^a) x_2^b \vee (x_1^a) x_3^a \vee (x_1^b \vee x_1^c) x_2^c x_3^a \vee$$
$$\vee x_1^a (x_2^a \vee x_2^b) \vee x_1^b (x_2^c) x_3^a \vee x_1^c (x_2^c) x_3^a \vee x_1^a (x_3^a) \vee$$
$$\vee x_1^b x_2^c (x_3^a) \vee x_1^c x_2^c (x_3^a).$$

2) К скобочным формам всюду, где возможно, применяем *операцию обобщенного дизъюнктивного склеивания*, основанную на использовании тождества:

$$\begin{gathered} A(x^{\sigma_1} \vee x^{\sigma_2} \vee \ldots \vee x^{\sigma_r}) \vee \\ \vee B(x^{\sigma_{r+1}} \vee x^{\sigma_{r+2}} \vee \ldots \vee x^{\sigma_s}) \equiv \\ \equiv A(x^{\sigma_1} \vee x^{\sigma_2} \vee \ldots \vee x^{\sigma_r}) \vee \\ \vee B(x^{\sigma_{r+1}} \vee x^{\sigma_{r+2}} \vee \ldots \vee x^{\sigma_s}) \vee AB. \end{gathered} \qquad (40)$$

Это тождество справедливо лишь в том случае, когда множество $\{\sigma_1, \sigma_2, \ldots, \sigma_s\}$совпадает с алфавитом A, причем в записи множества допускаются повторения букв. Результатом операции обобщенного дизъюнктивного склеивания является множество всех ненулевых конъюнкций вида AB. В нашем примере получаем множество, состоящее из единственного произведения $\{x_2^c x_3^a\}$. Заметим, что тождество (40), лежащее в основе операции обобщенного дизъюнктивного склеивания в алгебре конечных предикатов, выглядит сложнее, чем соответствующее тождество в алгебре логики. В частном случае при k=2 тождество (40) переходит в известное тождество алгебры логики: $Ax \vee B\bar{x} \equiv Ax \vee B\bar{x} \vee AB$, используемое в алгоритме Порецкого-Блейка. Докажем справедливость тождества (40):

$$A(x^{\sigma_1} \vee x^{\sigma_2} \vee \ldots \vee x^{\sigma_r}) \vee B(x^{\sigma_{r+1}} \vee x^{\sigma_{r+2}} \vee \ldots \vee x^{\sigma s}) \equiv$$

$$\equiv A(x^{\sigma_1} \vee x^{\sigma_2} \vee \ldots \vee x^{\sigma_r}) \vee AB(x^{\sigma_1} \vee x^{\sigma_2} \vee \ldots \vee x^{\sigma_r}) \vee$$

$$\vee B(x^{\sigma_{r+1}} \vee x^{\sigma_{r+2}} \vee \ldots \vee x^{\sigma_s}) \vee AB(x^{\sigma_{r+1}} \vee x^{\sigma_{r+2}} \vee \ldots \vee$$

$$\vee x^{\sigma_s}) \equiv AB(x^{\sigma_1} \vee x^{\sigma_2} \vee \ldots \vee x^{\sigma_s}) \vee$$

$$\vee A(x^{\sigma_1} \vee x^{\sigma_2} \vee \ldots \vee$$

$$. \vee x^{\sigma_r}) \vee B(x^{\sigma_{r+1}} \vee x^{\sigma_{r+2}} \vee \ldots \vee x^{\sigma_s}) \equiv AB(x^{a_1} \vee$$

$$\vee x^{a_2} \vee \ldots \vee x^{a_k}) \vee A(x^{\sigma_1} \vee x^{\sigma_2} \vee \ldots \vee x^{\sigma_r}) \vee B(x^{\sigma_{r+1}} \vee$$

$$\vee x^{\sigma_{r+2}} \vee \ldots \vee x^{\sigma s}) \equiv AB \vee A(x^{\sigma_1} \vee x^{\sigma_2} \vee \ldots \vee x^{\sigma_r}) \vee$$

$$\vee B(x^{\sigma_{r+1}} \vee x^{\sigma_{r+2}} \vee \ldots \vee x^{\sigma s}).$$

3) Исходную ДНФ пополняем дизъюнктивными членами из системы, полученной на втором шаге алгоритма. В нашем примере имеем:

$$f \equiv x_1^a x_2^a \vee x_1^a x_2^b \vee x_1^a x_3^a \vee x_1^b x_2^c x_3^a \vee x_1^c x_2^c x_3^a \vee x_2^c x_3^a.$$

4) Применяем всюду, где только возможно, *операцию дизъюнктивного поглощения* $A \vee AB \equiv A$. В результате выполнения этого шага алгоритма получаем сокращенную ДНФ конечного предиката. В примере сокращенная ДНФ имеет вид:

$$f \equiv x_1^a x_2^a \vee x_1^a x_2^b \vee x_1^a x_3^a \vee x_2^c x_3^a.$$

5) От сокращенной ДНФ переходим к минимальной ДНФ по методу, описанному в пункте б). В примере получаем следующую минимальную ДНФ:

$$f \equiv x_1^a x_2^a \vee x_1^a x_2^b \vee x_2^c x_3^a.$$

Переходим к описанию алгоритма минимизации формул алгебры конечных предикатов, обобщающего метод Нельсона минимизации формул алгебры логики. Исходной информацией при минимизации для этого алгоритма служит произвольно выбранная КНФ предиката, для которой отыскивается минимальная ДНФ. Рассмотрим пример. Пусть A=$\{a, b, c\}$, B=$\{x_1, x_2, x_3\}$. Предикат f задан следующей КНФ:

$$f \equiv (x_1^a \vee x_2^c)(x_2^a \vee x_2^b \vee x_3^a).$$

1) Раскрываем скобки и уничтожаем все нулевые дизъюнктивные члены. В примере получаем:

$$f \equiv x_1^a x_2^a \vee x_1^a x_2^b \vee x_1^a x_3^a \vee x_2^c x_3^a.$$

2) Применяем операцию дизъюнктивного поглощения. В результате получаем сокращенную ДНФ. В нашем примере применение операции дизъюнктивного поглощения не приводит к упрощениям, поэтому формула, выведенная на первом шаге алгоритма, есть сокращенная ДНФ.

3) По методу, описанному в пункте б), переходим от сокращенной ДНФ к минимальной ДНФ, которая в примере равна:

$$f \equiv x_1^a x_2^a \vee x_1^a x_2^b \vee x_2^c x_3^a.$$

Отметим, что в рассмотренном примере исходная КНФ предиката f имеет пять узнаваний букв, это меньше, чем в минимальной ДНФ, где их шесть. Этим примером демонстрируется важность задачи отыскания минимальной КНФ в алгебре конечных предикатов. Решение указанной задачи излагается в следующем пункте.

4. Конъюнктивная минимизация формул алгебры предикатов

г) ***Отыскание минимальной конъюнктивной нормальной формы.*** В алгебре логики имеет место принцип двойственности, в силу которого методы отыскания минимальной КНФ оказываются совершенно аналогичными методам отыскания минимальной ДНФ. В алгебре конечных предикатов при k>2 положение иное. Здесь мы лишены принципа двойственности, он теперь утрачивает силу, операции конъюнкции и дизъюнкции теперь уже не равноправны. Поэтому методы конъюнктивной минимизации должны рассматривается специально, здесь нельзя обойтись ссылкой на аналогию с методами дизъюнктивной минимизации.

Предикат g назовем *имплицентой* предиката f, если на любом наборе значений аргументов, для которого g=0, имеем также f=0. Будем говорить, что имплицента g накрывает своими нулями нули предиката f. Элементарную дизъюнкцию g назовем *собственной частью* элементарной дизъюнкции f, если g может быть получена из f выбрасыванием некоторых узнаваний букв. Например, элементарная дизъюнкция $x_1^a \vee x_1^b \vee x_3^c$ есть собственная часть элементарной дизъюнкции $x_1^a \vee x_1^b \vee x_2^c \vee x_3^c$.

Назовем *простой имплицентой* предиката f любую элементарную дизъюнкцию, обладающую следующими свойствами: она является имплицентой предиката f и никакая ее собственнная часть не является имплицентой предиката f.

Конъюнкция любого числа имплицент конечного предиката является имплицентой этого предиката. Систему S имплицент конечного предиката f назовем *полной*, если любой нуль из таблицы значений предиката f накрывается нулями хотя бы одной имплиценты системы S. Тому или иному конечному предикату может соответствовать, вообще говоря, несколько различных полных систем имплицент. Конъюнкция всех имплицент полной системы имплицент некоторого конечного предиката выражает собой этот предикат. Система всех простых имплицент любого конечного предиката есть его полная система. Конъюнкция всех простых имплицент конечного предиката выражает собой этот предикат, назовем ее *сокращенной* КНФ данного конечного предиката.

Конъюнктивным ядром конечного предиката назовем множество всех таких его простых имплицент, исключение каждой из которых из системы всех простых имплицент делает систему неполной. Элементы конъюнктивного ядра входят в состав любой полной системы простых имплицент. Систему простых имплицент конечного предиката назовем *приведенной*, если она полна и никакое ее собственное подмножество не является полной системой. Конъюнкцию всех простых имплицент приведенной системы назовем *тупиковой конъюнктивной нормальной формой* предиката. Выбирая из числа тупиковых КНФ одну с наименьшим числом узнаваний, получаем минимальную КНФ.

Рассмотрим метод конъюнктивной минимизации формул алгебры конечных предикатов, *обобщающий* метод Квайна-Мак-Класки. Исходной информацией при минимизации служит СКНФ предиката, для которого отыскивается минимальная КНФ. Рассмотрим пример. Пусть $A=\{a, b, c\}$, $B=\{x_1, x_2, x_3\}$. Предикат задан следующей СКНФ:

$$f \equiv (\overline{x_1^a} \vee \overline{x_2^a} \vee \overline{x_3^a})(\overline{x_1^a} \vee \overline{x_2^b} \vee \overline{x_3^a}) \wedge$$
$$\wedge(\overline{x_1^b} \vee \overline{x_2^a} \vee \overline{x_3^a})(\overline{x_1^b} \vee \overline{x_2^b} \vee \overline{x_3^a})(\overline{x_1^c} \vee \overline{x_2^c} \vee \overline{x_3^a}) \wedge$$
$$\wedge(\overline{x_1^c} \vee \overline{x_2^c} \vee \overline{x_3^b})(\overline{x_1^c} \vee \overline{x_2^c} \vee \overline{x_3^c}).$$

СКНФ представлена в сокращенной записи с использованием знаков отрицания. Последующей обработке подвергается сама СКНФ, а не ее сокращенная запись.

Ниже описывается алгоритм минимизации:

1) Всюду, где это возможно, применяем *операцию неполного конъюнктивного склеивания*, основанного на тождестве

$$(A \vee x^{a_i})(A \vee x^{a_j}) \equiv (A \vee x^{a_i})(A \vee x^{a_j})A\,, \qquad (41)$$

которое справедливо при условии $i \neq j$. Операция неполного конъюнктивного склеивания состоит в дописывании в исходной КНФ в виде дополнительных конъюнктивных членов всевозможных элементарных дизъюнкций A. Последние можно получить переходом от левой части тождества (40) к правой. Операцию неполного конъюнктивного склеивания сначала применяем к конституэнтам нуля исходной СКНФ, затем — к элементарным дизъюнкциям, состоящим из $n(k\text{-}1)\text{-}1$ узнаваний букв, $n(k\text{-}1)\text{-}2$ узнаваний букв и т.д., наконец, к элементарным дизъюнкциям, состоящим из одного узнавания буквы. В нашем примере имеем

$$f \equiv (x_1^b \vee x_1^c \vee x_2^b \vee x_2^c \vee x_3^b \vee x_3^c)(x_1^b \vee x_1^c \vee x_2^a \vee x_2^c \vee x_3^b \vee$$
$$\vee x_3^c)(x_1^a \vee x_1^c \vee x_2^b \vee x_2^c \vee x_3^b \vee x_3^c)(x_1^a \vee x_1^c \vee x_2^a \vee x_2^c \vee x_3^b \vee$$
$$\vee x_3^c)(x_1^a \vee x_1^b \vee x_2^a \vee x_2^b \vee x_3^b \vee x_3^c)(x_1^a \vee x_1^b \vee x_2^a \vee x_2^b \vee x_3^a \vee$$
$$\vee x_3^c)(x_1^a \vee x_1^b \vee x_2^a \vee x_2^b \vee x_3^a \vee x_3^b)(x_1^b \vee x_1^c \vee x_2^c \vee x_3^b \vee$$
$$\vee x_3^c)(x_1^c \vee x_2^a \vee x_2^c \vee x_3^b \vee x_3^c)(x_1^a \vee x_1^c \vee x_2^c \vee x_3^b \vee x_3^c) \wedge$$
$$\wedge(x_1^a \vee x_1^b \vee x_2^a \vee x_2^b \vee x_3^c)(x_1^a \vee x_1^b \vee x_2^a \vee x_2^b \vee x_3^a) \wedge$$
$$\wedge(x_1^c \vee x_2^c \vee x_3^b \vee x_3^c)(x_1^a \vee x_1^b \vee x_2^a \vee x_2^b)\,.$$

2) Применяем к конъюнктивным членам полученной формулы *операцию элементарного конъюнктивного поглощения*:

$$(A \vee x^{\sigma})A \equiv A\,. \qquad (42)$$

В результате получаем сокращенную КНФ. В нашем примере имеем:

$$f \equiv (x_1^a \vee x_1^b \vee x_2^a \vee x_2^b)(x_1^c \vee x_2^c \vee x_3^b \vee x_3^c).$$

В примере сокращенная КНФ совпадает с минимальной КНФ. В более сложных случаях процесс минимизации продолжается.

3) Составляем *имплицентную таблицу*. Имплицентная таблица конечного предиката f представляет собой матрицу, строки которой обозначены всевозможными простыми имплицентами предиката f, а столбцы — конституэнтами нуля того же предиката. Каждая ячейка таблицы соответствует паре предикатов, один из которых представлен конституэнтой нуля, другой — некоторой элементарной дизъюнкцией. Если элементарная дизъюнкция, соответствующая некоторой строке таблицы, является имплицентой для конституэнты нуля, соответствующей некоторому столбцу, то в ячейку, образуемую пересечением строки и столбца, заносим звездочку, в противном случае ячейку оставляем незаполненной.

4) По имплицентной таблице находим конъюктивное ядро предиката. Для этого находим те столбцы таблицы, в которых содержится только по одной звездочке. Соответствующие звездочкам простые имплиценты образуют конъюнктивное ядро предиката.

5) По имплицентной таблице отыскиваем систему всех простых конституэнт нуля предиката, которые не накрываются нулями его конъюнктивного ядра. С этой целью отмечаем все столбцы, в которых содержатся звездочки, соответствующие простым имплицентам, вошедшие в состав конъюнктивного ядра. Столбцы, оставшиеся неотмеченными, соответствуют искомым конституэнтам единицы.

6) По имплицентной таблице методом полного перебора отыскиваем все приведенные системы простых имплицент, накрывающих своими нулями все конституэнты нуля системы, найденной на пятом шаге алгоритма.

7) Объединяя конъюнктивное ядро предиката с каждой из систем, полученных на шестом шаге алгоритма, формируем все приведенные системы простых имплицент предиката.

8) Строим все тупиковые КНФ предиката.

Приведем описание алгоритма минимизации формул алгебры конечных предикатов, *обобщающего* алгоритм Порецкого-Блейка конъюнктивной минимизации формул алгебры логики. Исходной информацией при минимизации для этого алгоритма служит произвольно выбранная КНФ предиката, для которой отыскивается минимальная КНФ. Рассмотрим пример. Пусть $A=\{a, b, c\}$, $B=\{x_1, x_2\}$. Предикат f задан следующей КНФ:

$$f \equiv (x_1^b \vee x_1^c \vee x_2^a \vee x_a^b)(x_1^a \vee x_1^b \vee x_2^a \vee x_2^b)$$
$$(x_1^a \vee x_1^c \vee x_2^c)(x_1^a \vee x_1^b \vee x_2^b \vee x_2^c).$$

1) К конъюнктивным членам исходной КНФ всюду, где возможно, применяем *операцию обобщенного конъюнктивного склеивания*, основанную на использовании тождества:

$$(A \vee x^{a_i})(B \vee x^{a_j}) \equiv \equiv (A \vee x^{a_i})(B \vee x^{a_j})(A \vee B), \quad (43)$$

справедливого в случае, когда $i \neq j$. Результатом операции обобщенного конъюнктивного склеивания является множество всех не равных единице элементарных дизъюнкций вида $A \vee B$. В нашем примере получаем следующее множество элементарных дизъюнкций:

$$\{x_1^b \vee x_1^c \vee x_2^a \vee x_2^b,\ x_1^a \vee x_1^b \vee x_2^a \vee x_2^b,\ x_1^b \vee x_2^a \vee x_2^b,$$
$$x_1^a \vee x_1^b \vee x_2^b \vee x_2^c,\ x_1^a \vee x_1^b \vee x_2^b,\ x_1^a \vee x_1^c \vee x_2^b \vee x_2^c,$$
$$x_1^a \vee x_2^b \vee x_2^c\}.$$

Заметим, что тождество (43), лежащее в основе операции обобщенного конъюнктивного склеивания, в алгебре конечных предикатов выглядит несколько иначе, чем соответствующее тождество в алгебре логики. В частном случае при $k=2$ тождество (41) переходит в известное тождество алгебры логики

$$(A \vee x)(B \vee \bar{x}) \equiv (A \vee x)(B \vee \bar{x})(A \vee B),$$

используемое в алгоритме Порецкого-Блейка. Докажем справедливость тождества (41):

$$(A \vee x^{a_i})(B \vee x^{a_j}) \equiv (A \vee x^{a_i})(A \vee x^{a_i} \vee B)(B \vee x^{a_j}) \wedge$$
$$\wedge (B \vee x^{a_j} \vee A) \equiv (A \vee x^{a_i})(B \vee x^{a_j})(A \vee B \vee x^{a_i} x^{a_j}) \equiv$$
$$\equiv (A \vee x^{a_i})(B \vee x^{a_j})(A \vee B).$$

2) Исходную КНФ пополняем конъюнктивными членами из системы, полученной на первом шаге алгоритма. В нашем примере имеем

$$f \equiv (x_1^b \vee x_1^c \vee x_2^a \vee x_2^b)(x_1^a \vee x_1^b \vee x_2^a \vee x_2^b) \wedge$$
$$\wedge (x_1^a \vee x_1^c \vee x_2^c)(x_1^a \vee x_1^b \vee x_2^b \vee x_2^c)(x_1^b \vee x_2^a \vee x_2^b) \wedge$$
$$\wedge (x_1^a \vee x_2^b \vee x_2^b)(x_1^a \vee x_1^c \vee x_2^b \vee x_2^c)(x_1^a \vee x_2^b \vee x_2^c).$$

Таблица 8

Простая имплицента	Конституэнта нуля				
	$x_1^b \vee x_1^c \vee x_2^a \vee x_2^b$	$x_1^a \vee x_1^c \vee x_2^b \vee x_2^c$	$x_1^a \vee x_1^c \vee x_2^a \vee x_2^c$	$x_1^a \vee x_1^b \vee x_2^b \vee x_2^c$	$x_1^a \vee x_1^b \vee x_2^a \vee x_2^b$
$x_1^a \vee x_1^c \vee x_2^c$		*	*		
$x_1^b \vee x_2^a \vee x_2^b$	*				*
$x_1^a \vee x_1^b \vee x_2^b$				*	*
$x_1^a \vee x_2^b \vee x_2^c$		*		*	

3) Применяем всюду, где возможно, *операцию конъюнктивного поглощения* $A(A \vee B) \equiv A$. В результате выполнения этого шага алгоритма получаем сокращенную КНФ конечного предиката. В нашем примере имеем следующую сокращенную КНФ:

$$f \equiv (x_1^a \vee x_1^c \vee x_2^c)(x_1^b \vee x_2^a \vee x_2^b) \wedge$$
$$\wedge (x_1^a \vee x_1^b \vee x_2^b)(x_1^a \vee x_2^b \vee x_2^c).$$

4) Выполняем шаги 3-9 предыдущего алгоритма. В нашем примере имплицентная таблица имеет вид, представленный табл. 8, по которой находим конъюнктивное ядро предиката:

$$\{x_1^a \vee x_1^c \vee x_2^c, x_1^b \vee x_2^a \vee x_2^b\}.$$

Имеются две тупиковые КНФ:

$$(x_1^0 \vee x_1^A \vee x_2^A)(x_1^b \vee x_2^a \vee x_2^b)(x_1^a \vee x_1^b \vee x_2^b),$$
$$(x_1^a \vee x_1^c \vee x_2^c)\ (x_1^b \vee x_2^a \vee x_2^b)(x_1^a \vee x_2^b \vee x_2^c),$$

каждая из которых может быть принята в качестве минимальной КНФ предиката f.

Переходим к описанию алгоритма минимизации формул алгебры конечных предикатов, *обобщающего* алгоритм Нельсона конъюнктивной минимизации формул алгебры логики. Исходной информацией при минимизации служит произвольно выбранная ДНФ предиката, для которого отыскивается минимальная КНФ. Рассмотрим пример. Пусть $A=\{a, b, c\}$, $B=\{x_1, x_2, x_3\}$. Предикат задан следующей ДНФ:

$$f \equiv x_1^a x_2^a \vee x_1^a x_2^b \vee x_2^c x_3^a .$$

1) Пользуясь вторым законом дистрибутивности, переходим к некоторой КНФ предиката *f*. В примере имеем:

$$f \equiv (x_1^a \vee x_2^c)(x_1^a \vee x_3^a)(x_1^a \vee x_2^b \vee x_2^c) \wedge$$
$$\wedge (x_1^a \vee x_2^b \vee x_3^a) \wedge (x_1^a \vee x_2^a \vee x_2^c) \wedge$$
$$\wedge (x_1^a \vee x_2^a \vee x_3^a)(x_2^a \vee x_2^b \vee x_3^a).$$

2) Применяем операцию конъюнктивного поглощения. В результате получаем сокращенную КНФ. В нашем примере сокращенная КНФ имеет вид:

$$f \equiv (x_1^a \vee x_2^c)(x_1^a \vee x_3^a)(x_2^a \vee x_2^b \vee x_3^c) .$$

3) Выполняем шаги 3-9 алгоритма, описанного первым в пункте г). В примере получаем следующую минимальную КНФ:

$$f \equiv (x_1^a \vee x_2^c)(x_2^a \vee x_2^b \vee x_3^a) .$$

Заметим, что в области минимизации формул алгебры конечных предикатов много важных задач еще ждет своего решения. К их числу относятся: дальнейшая разработка методов канонической минимизации; разработка оценок сложности минимальных форм; разработка методов минимизации скобочных форм; разработка методов факторизации (разыскания форм с наименьшим числом вхождения знаков конъюнкции и дизъюнкции).

Выводы

У читателя, ознакомившегося с [1] и настоящей статьей, может возникнуть вопрос: почему в работе, посвященной теории интеллекта, так много внимания уделяется разработке формального языка в виде алгебры конечных предикатов? Не лучше ли было бы, довольствуясь существующим математическим аппаратом, сразу же взяться за моделирование какой-нибудь сложной интеллектуальной деятельности, например, игры в шахматы или доказательства теорем. К своему стыду, авторы когда-то именно так и поступили. Объектом математического описания был выбран русский язык. В качестве формальных языков, на которых велось описание этого объекта, были испробованы языки программирования [7], аппарат теории графов [8], язык теории алгоритмов [9], логические исчисления [10]. Много лет ушло в малоэффективных попытках, пока наконец стало ясно, что все эти средства настолько плохо приспособлены для целей формального описания человеческого языка, что создают труднопреодолимые препятствия при его моделировании. Поскольку естественный язык лежит в основе интеллектуальной деятельности человека, то можно не сомневаться, что и моделирование вышележащих слоев интеллекта будет столь же неэффективным, если ограничится существующими математическими средствами.

Трудности, с которыми пришлось столкнуться при моделировании естественного языка, в конце концов были проанализированы. Из них родились требования, предъявляемые к формальному языку, способному эффективно описывать человеческий язык. В свою очередь, сформулированные требования, можно сказать, принудительно привели к алгебре конечных предикатов. Проводившееся затем математическое описание явлений русского языка средствами алгебры конечных предикатов продемонстрировало преимущества этой алгебры, причем настолько большие, что в дальнейшем ни разу не возникала необходимость обратиться к помощи какого-нибудь другого формального языка. Ниже излагаются доводы, которые убедили авторов в том, что алгебра конечных предикатов как раз и есть тот формальный язык, на котором должно вестись описание проявлений человеческого языка (а может быть, и многих других функций интеллекта), что никакие другие формальные средства не могут составить ему конкуренцию и что именно этот язык призван стать фундаментом, на котором должно строиться здание теории интеллекта.

Человеческий язык как явление дискретное, естественно, должен описываться средствами дискретной математики. Между тем выбор средств указанного типа весьма ограничен. Это — языки программирования для ЭВМ, логические исчисления, языки теории алгоритмов, аппарат теории графов. При попытке использования языков программирования или языков теории алгоритмов приходится столкнуться со следующими труднопреодолимыми препятствиями. Эти языки, как известно, предназначены для описания алгоритмов [11], то есть процедур с однозначным исходом. Между тем естественный язык многозначен, что проявляется, например, в виде омографичности слов, то есть неоднозначности их смысла. Языки программирования и теории алгоритмов – это такие языки, которые могут описывать только однозначные функции. Естественный же язык требует формальных средств для описания многозначных функций, то есть соответствий произвольного вида, иначе говоря, – отношений.

Для описания человеческого языка лучше всего подошел бы аппарат уравнений, подобный аппарату, используемому в математическом анализе, отличающийся от последнего тем, что он предназначен для формализации не непрерывных, а дискретных процессов. Такой язык дают логические исчисления, а именно: исчисления высказываний и исчисления предикатов. Однако, чтобы иметь возможность эффективно решать указанные уравнения, необходимо довести интересующие нас исчисления до уровня алгебраической системы. Сделано же это только в исчислении высказываний. В результате мы имеем алгебру логики и аппарат

булевых уравнений [12]. Однако аппарат булевых уравнений для формального описания естественного языка наталкивается на серьезное неудобство, заключающееся в том, что в алгебре логики используются лишь двоичные знаки, в то время как в естественном языке фигурируют буквенные, то есть многозначные символы.

Этого недостатка, казалось бы, можно избежать, обратившись к аппарату многозначной логики [13]. Однако при ближайшем рассмотрении обнаруживается, что многозначная логика развита только в сторону описания однозначных функций, а не отношений. Учение об уравнениях в многозначной логике, насколько нам известно, совершенно не развито. Развитие же в этом направлении многозначной логики принудительно приводит к алгебре конечных предикатов. Действительно, чтобы иметь возможность записывать самые общие уравнения многозначной логики, в правой их части нет необходимости ставить произвольные формулы, достаточно писать константы. Необязательно использовать все константы, достаточно взять всего две знака: 0 и 1. Но как только мы так поступим, немедленно приходим к понятию конечного предиката, а следовательно, и к алгебре конечных предикатов.

Использование исчисления предикатов [3] для целей математического описания человеческого языка также наталкивается на определенную трудность: исчисление очень слабо развито применительно к нуждам описания конечных объектов. Исчисление предикатов не располагает даже средствами для формульной записи любых индивидуальных конечных отношений. Вместе с тем, человеческий язык – явление сугубо конечное и он требует для своей формализации аппарата конечной математики. Пытаясь алгебраизировать конечный фрагмент исчисления предикатов, мы не сможем прийти ни к чему иному, как только к алгебре конечных предикатов. Наконец, обратившись к аппарату теории графов [14], мы обнаружим, что, хотя он и используется для описания конечных отношений, однако совершенно не содержит в себе выразительных средств для записи этих отношений в виде уравнений некоторой алгебры. Если же мы захотим перевести информацию, содержащуюся в графах, на язык таблиц, то увидим, что с помощью графов выражаются именно конечные предикаты.

Таким образом, какой бы путь мы ни избрали при разработке приемлемых формальных средств для математического описания человеческого языка, мы неизбежно приходим к алгебре конечных предикатов. Вместе с тем в данной статье установлено, что алгебра конечных предикатов полна, то есть на ее языке могут быть описаны любые конечные отношения. Поэтому любой другой математический аппарат, предназначенный для описания произвольных конечных отношений, в логическом смысле обязательно будет равносилен алгебре конечных предикатов.

Список литературы: 1. *Бондаренко, М.Ф.* Об алгебре конечных предикатов [Текст] / *М.Ф. Бондаренко, Ю.П. Шабанов-Кушнаренко, С.Ю. Шабанов-Кушнаренко* // Бионика интеллекта: науч.-техн. журнал. – 2011. – № 3. – С. 3-13. **2.** *Мальцев, А.И.* Алгебраические системы [Текст] / А.И. Мальцев. – М.: Наука, 1970. – 310 с. **3.** *Ершов, Ю.Л.* Математическая логика / Ю.Л. Ершов, Е.А. Палютин. – М.: Наука, 1979. – 372 с. **4.** *Глушков, В.М.* Введение в кибернетику [Текст] / В.М. Глушков. – К.: Изд. АН УССР, 1964. – С. 67-75. **5.** *Глушков, В.М.* Синтез цифровых автоматов [Текст] / В.М. Глушков. – М.: Физматгиз, 1962. – 316 с. **6.** *Шабанов-Кушнаренко, Ю.П.* Об одной математической модели морфологической классификации множества имен существительных русского языка [Текст] / Ю.П. Шабанов-Кушнаренко, Л.И. Якименко. – Проблемы бионики. 1971. – Вып. 6. – С. 104-107. **7.** *Шабанов-Кушнаренко, Ю.П.* Математическая модель определения классов одинаковых слов [Текст] / Ю.П. Шабанов-Кушнаренко, Л.И. Якименко // Проблемы бионики. – 1971. – Вып. 7. – С. 103-105. **8.** *Шабанов-Кушнаренко, Ю.П.* К вопросу об автоматическом морфологическом анализе причастий русского языка [Текст] / Ю.П. Шабанов-Кушнаренко, Е.А. Соловьева // Проблемы бионики. – 1974. – Вып. 12. – С. 46-50. **9.** *Осыка, А.Ф.* Модель определения исходной формы числительных русского языка [Текст] / А.Ф. Осыка, Ю.П. Шабанов-Кушнаренко // Проблемы бионики. – 1976. – Вып. 16. – С. 78-84. **10.** *Осыка, А.Ф.* К вопросу о моделировании грамматической обработки имен числительных [Текст] / А.Ф. Осыка. – Пробл. бионики, 1979, Вып. 22, С. 129-137. **11.** *Мальцев, А.И.* Алгоритмы и рекурсивные функции [Текст] / А.И. Мальцев. – М.: Наука, 1965. – С. 9-11. **12.** *Закревский, А.Д.* Логические уравнения [Текст] / А.Д. Закревский. – Минск: Наука и техника, 1975. – С. 92. **13.** Дискретная математика и математические вопросы кибернетики [Текст] / Под ред. С.В. Яблонского и О.Б. Лупанова. – М.: Наука, 1974. – Т. 1. – С. 35-66. **14.** *Белов, В.В.* Теория графов [Текст] / В.В. Белов, Е.Н. Воробьев, В.Е. Шаталов. – М.: Высш. школа, 1976. – 389 с.

Поступила в редколлегию 15.06.2011

УДК 519.7

Нормальні форми формул алгебри скінченних предикатів / М.Ф. Бондаренко, Ю.П. Шабанов-Кушнаренко, С.Ю. Шабанов-Кушнаренко // Біоніка інтелекту: наук.-техн. журнал. – 2011. – № 3 (77). – С. 14-29.

Розглянуто побудову нормальних форм формул алгебри скінченних предикатів, методи кон'юнктивної мінімізації. Сформульовано теорему про конюнктивний розклад предикатів.

Табл. 8. Бібліогр.: 14 найм.

UDC 519.7

PDNF in final predicates algebra / M.F. Bondarenko, S.Yu. Shabanov-Kushnarenko, Yu.P. Shabanov-Kushnarenko // Bionics of Intelligense: Sci. Mag. – 2011. – № 3 (77). –P. 14-29.

The construction of the final predicates algebra formulas normalized forms and the of conjunctive minimization methods is considered. The theorem about the predicates conjunctive decomposition is formulated.

Ref.: 14 items.

УДК 519.7

М.Ф. Бондаренко, Ю.П. Шабанов-Кушнаренко
ХНУРЭ, г. Харьков, Украина

УРАВНЕНИЯ ТЕОРИИ ИНТЕЛЛЕКТА

Разрабатывается математический аппарат для формализации и моделирования систем искусственного интеллекта. Предложены способы записи уравнений теории интеллекта, а также методы их аналитического решения. Предложено несколько различных методов аналитического представления конечных алфавитных операторов с помощью формул универсальной алгебры.

АЛГЕБРА КОНЕЧНЫХ ПРЕДИКАТОВ, ПРЕДИКАТ, ИНТЕЛЛЕКТ, УРАВНЕНИЕ, УНИВЕРСАЛЬНАЯ АЛГЕБРА, АЛФАВИТНЫЙ ОПЕРАТОР

Введение

Предикаты – это основной математический инструмент теории интеллекта, они используются для формального описания объектов бионики интеллекта. Алгебра предикатов – это логический аппарат первой ступени, ее можно рассматривать как аналог школьной алгебры, на языке которой записываются числовые функции. Аппаратом второй ступени в логике является *алгебра предикатных операций*, которую можно рассматривать как логический аналог изучаемого в вузах математического анализа. Алгебры предикатов и предикатных операций рекомендуются в качестве языка формального описания для разработчиков, создающих средства искусственного интеллекта. Язык алгебры предикатов представляет собой универсальное средство формального описания любых объектов, наблюдаемых в мире, на нем можно выразить любое отношение. Язык алгебры предикатных операций тоже универсален, но он используется уже в ином качестве: на нем можно выразить любые *действия над отношениями*. Когда возникает необходимость формально описать какой-нибудь процесс или объект, то прибегают к услугам алгебры предикатов. Когда же требуется реально воспроизвести уже описанный ранее процесс или создать в натуре спроектированный объект (то есть нужно перейти от *слов* к *делу*), то используется аппарат алгебры предикатных операций.

Отношения выражают внутреннее строение объектов, свойства объектов и связи между ними. Они представляют собой *универсальное* средство формального представления любых объектов. За две с половиной тысячи лет существования науки ей не удалось обнаружить в мире ни одного объекта, о котором можно было бы с уверенностью сказать, что он, в принципе, не поддается формальному представлению с помощью отношений. Никакие другие известные средства формального представления объектов (например, школьная алгебра и математический анализ) свойством универсальности не обладают. *Естественный язык*, представляющий собой *универсальное* средство духовного общения людей, можно рассматривать как инструмент для выражения отношений. Обращаясь с *речью* к другим людям, мы передаем им вполне определенный *смысл* произносимого *предложения*, который есть не что иное, как некоторое отношение. *Обмен мыслями* между людьми осуществляется за счет передачи и приема отношений. Каждая *мысль* представляет собой какое-то отношение. *Мышление* же есть процесс преобразования отношений, получения новых отношений из уже имеющихся. *Информация*, поступающая к нам из внешнего мира, имеет вид отношений, характеризующих структуру, свойства и связи окружающих нас предметов и процессов. *Результатом действий человека* во внешнем мире является приведение структуры предметов и процессов в соответствие тем отношениям, которые были сформированы в уме человека в результате его мыслительной деятельности.

1. Канонические уравнения

Любая интересующая нас информации о деятельности и суждениях интеллекта, в принципе, может быть записана в виде уравнений алгебры конечных предикатов. В данной статье будут рассмотрены способы записи таких уравнений, а также методы их аналитического решения. Для большего удобства записи уравнений алгебры конечных предикатов нам понадобятся, кроме уже введенных четырех операций – дизъюнкции, конъюнкции, отрицания и импликации, еще две двуместные операции – *равнозначность* ~ и *неравнозначность* ⊕ *предикатов*. Эти операции формально определим следующими равенствами:

$$A \sim B \underset{def}{\equiv} AB \vee \overline{A}\,\overline{B} \qquad (1)$$

$$A \oplus B \underset{def}{\equiv} A\overline{B} \vee \overline{A}B \qquad (2)$$

Здесь буквами A и B обозначены произвольно выбранные формулы алгебры конечных предикатов. Условимся в случае отсутствия в формуле скобок, регулирующих порядок выполнения операций, непосредственно после выполнения операций отрицания, конъюнкции, дизъюнкции импликации выполнять операции равнозначности и лишь затем – операции неравнозначности.

Свойства введенных операций аналогичны свойствам одноименных операций, рассматриваемых в алгебре логики. Все они могут быть выведены из только что принятых определений. Перечислим наиболее важные тождества алгебры конечных предикатов, в которых фигурируют знаки ~, ⊕, а именно – *рефлексивность равнозначности*:

$$A\sim A\equiv 1; \tag{3}$$

антирефлексивность неравнозначности:

$$A\oplus A\equiv 0; \tag{4}$$

коммутативность равнозначности и неравнозначности:

$$A\sim B\equiv B\sim A; \tag{5}$$

$$A\oplus B\equiv B\oplus A; \tag{6}$$

ассоциативность равнозначности и неравнозначности:

$$(A\sim B)\sim C\equiv A\sim(B\sim C); \tag{7}$$

$$(A\oplus B)\oplus C\equiv A\oplus(B\oplus C); \tag{8}$$

дистрибутивность неравнозначности:

$$(A\oplus B)\,C\equiv AC\oplus BC; \tag{9}$$

транзитивность равнозначности:

$$(A\sim B)(B\sim C)\supset(A\sim C)\equiv 1; \tag{10}$$

свойства логических констант:

$$A\sim 0\equiv \overline{A}; \tag{11}$$

$$A\sim 1\equiv A; \tag{12}$$

$$A\oplus 0\equiv A; \tag{13}$$

$$A\oplus 1\equiv \overline{A}. \tag{14}$$

Уравнение вида

$$f_1(x_1, x_2,..., x_n)=1, \tag{15}$$

где f – произвольно выбранный фиксированный конечный предикат, назовем *каноническим уравнением* алгебры конечных предикатов для предиката f. К каноническому виду может быть приведено любое уравнение алгебры конечных предикатов. Действительно, уравнение

$$f_1(x_1, x_2,..., x_n)=f_2(x_1, x_2,..., x_n), \tag{16}$$

где f_1, f_2 – произвольно выбранные фиксированные конечные предикаты, является наиболее общим видом уравнения алгебры конечных предикатов, однако оно может быть записано в виде равносильного ему канонического уравнения:

$$f_1(x_1, x_2,..., x_n)\sim f_2(x_1, x_2,..., x_n)=1. \tag{17}$$

Система канонических уравнений

$$\begin{cases} f_1(x_1, x_2,..., x_n)=1, \\ f_2(x_1, x_2,..., x_n)=1, \\ \ldots\ldots\ldots\ldots\ldots \\ f_m(x_1, x_2,..., x_n)=1. \end{cases} \tag{18}$$

может быть записана в виде одного равносильного ей канонического уравнения:

$$\begin{gathered} f_1(x_1, x_2,..., x_n)\wedge f_2(x_1, x_2,..., x_n)\wedge\ldots \\ \wedge f_m(x_1, x_2,..., x_n)=1. \end{gathered} \tag{19}$$

Равносильными назовем такие уравнения или системы уравнений, множества всех решений которых совпадают между собой. В дальнейшем мы ограничимся рассмотрением только канонических уравнений, имея в виду, что любое уравнение или система уравнений алгебры конечных предикатов могут быть заменены равносильными им каноническими уравнениями.

Любое каноническое уравнение может быть интерпретировано как некоторое высказывание. В самом деле, уравнение $x^\sigma=1$ равносильно высказыванию «$x=\sigma$»: если $x^\sigma=1$, то $x=\sigma$; если $x=\sigma$, то $x^\sigma=1$. Точно так же, уравнения $A\vee B=1$, $AB=1$, $\overline{A}=1$, $A\supset B=1$, $A\sim B=1$, $A\oplus B=1$ соответственно равносильны высказываниям «A или также B», «A и B», «ложно, что A», «если A, то B», «A равносильно B», «или A, или B». На том же основании можно утверждать, что любое сложное высказывание, у которого в качестве простых высказываний фигурируют канонические уравнения, может быть представлено в виде некоторого канонического уравнения. Для перехода от такого высказывания к каноническому уравнению нужно в высказывании логические связки «или также», «и», «ложно, что», «если – то», «равносильно», «или – или» заменить соответственно символами $\vee$, $\wedge$, , $\supset$, ~, $\oplus$ и приравнять полученную формулу к единице.

Как решить каноническое уравнение $f(x_1, x_2,..., x_n)=1$ алгебры конечных предикатов? Решить уравнение – это значит отыскать множество всех наборов значений переменных $(x_1, x_2,..., x_n)$, удовлетворяющих этому уравнению, то есть обращающих его левую часть в 1. Мы уже располагаем универсальным, хотя, быть может, громоздким и не всегда удобным, методом решения канонического уравнения. В качестве такого метода может быть использован алгоритм приведения любой формулы алгебры конечных предикатов к СДНФ, описанный в п. 3 [1]. Пусть задано уравнение (15) и требуется его решить. Представляя предикат f в виде СДНФ, имеем:

$$f(x_1, x_2,..., x_n)=\bigvee_{f(\sigma_1,\sigma_2,...,\sigma_n)=1} x_1^{\sigma_1}x_2^{\sigma_2}...x_n^{\sigma_n}.$$

Каждой конституэнте единицы $x_1^{\sigma_1}x_2^{\sigma_2}...x_n^{\sigma_n}$ СДНФ предиката f соответствует решение $(\sigma_1, \sigma_2,..., \sigma_n)$ уравнения (15). Вместе взятые, эти решения образуют систему всех решений уравнения (15).

Рассмотрим пример решения уравнения вида (15) при $A=\{a, b\}$, $B=\{x_1, x_2, x_3\}$. Предикат f задан формулой:

$$f(x_1, x_2, x_3)\equiv$$
$$(x_1{}^a\vee x_2{}^b)(x_1{}^a\vee x_1{}^b x_3{}^a)\vee(x_1{}^a\vee x_2{}^a)(x_2{}^b x_3{}^b\vee x_2{}^a x_3{}^a).$$

СДНФ этого предиката имеет вид:

$$f\equiv x_1^{\,a}x_2^{\,a}x_3^{\,a}\vee x_1^{\,a}x_2^{\,a}x_3^{\,b}\vee x_1^{\,a}x_2^{\,b}x_3^{\,a}\vee \vee x_1^{\,a}x_2^{\,b}x_3^{\,b}\vee x_1^{\,b}x_2^{\,a}x_3^{\,a}\vee x_1^{\,b}x_2^{\,b}x_3^{\,b}.$$

По СДНФ предиката f находим систему T всех решений уравнения $f(x_1, x_2, x_3)=1$:

$$T=\{(a, a, a), (a, a, b), (a, b, a), (a, b, b), (b, a, a), (b, b, b)\}.$$

Система всех решений уравнения (15) может быть представлена, как правило, в значительно более компактном виде, чем в форме СДНФ, если воспользоваться алгоритмом получения некоторой ДНФ предиката, описанным в [1]. Более компактное представление системы всех решений предиката достигается использованием минимальной ДНФ. Каждой элементарной конъюнкции

$$x_{i_1}^{\sigma i_1}x_{i_2}^{\sigma i_2}\ldots x_{i_s}^{\sigma i_s}$$

($s\le n$) ДНФ предиката соответствует некоторое подмножество системы всех решений уравнения (15), составленное из всевозможных наборов значений переменных $(x_1, x_2,..., x_n)$, у которых на $i_1, i_2,..., i_s$ местах стоят соответственно буквы $\sigma_{i_1},\sigma_{i_2},...,\sigma_{i_s}$. Объединение всех таких подмножеств дает систему всех решений уравнения (15).

Рассмотрим пример представления системы всех решений уравнения алгебры конечных предикатов при $A=\{a, b, c\}$ и $B=\{x_1, x_2, x_3\}$с помощью минимальной ДНФ. Предикат f задан формулой

$$f(x_1, x_2, x_3)\equiv(x_1^{\,a}\vee x_2^{\,c})(x_2^{\,a}\vee x_2^{\,b}\vee x_3^{\,a}).$$

Минимальная ДНФ этого предиката имеет вид:

$$f=x_1^{\,a}x_2^{\,a}\vee x_1^{\,a}x_2^{\,b}\vee x_2^{\,c}x_3^{\,a}.$$

Система T всех решений уравнения $f(x_1, x_2, x_3)=1$ может быть представлена в следующей сокращенной форме $T=\{(a, a, x_3), (a, b, x_3), (x_1, c, a)\}$. В данной записи под x_1 и x_3 понимаются произвольные буквы алфавита A. Полная же запись системы всех решений имеет вид: $T=\{(a, a, a), (a, a, b), (a, a, c), (a, b, a), (a, b, b), (a, b, c), (a, c, a), (b, c, a), (c, c, a)\}$.

Введенные ранее понятия конечного предиката, формулы алгебры конечных предикатов, канонического уравнения и множества всех его решений допускают содержательную логическую интерпретацию. Формулы алгебры конечных предикатов можно интерпретировать как *высказывания* или *суждения интеллекта*. Каноническое уравнение $A=1$, где A – произвольно выбранная формула алгебры конечных предикатов, можно интерпретировать как утверждение об *истинности высказывания A*, уравнение $A=0$ можно интерпретировать как утверждение о *ложности высказывания A*. Конечный предикат $f(x_1, x_2,..., x_n)$, обозначаемый формулой A, а также эквивалентное ему множество T всех решений $(x_1, x_2,..., x_n)$ уравнения $A=1$ можно интерпретировать как *содержание высказывания A*.

Формулы, обозначающие тождественно истинный предикат, назовем *тождественно истинными*, их можно интерпретировать как *бессодержательные* или *тавтологические высказывания*. Формулы, обозначающие тождественно ложный предикат, назовем *тождественно ложными*. Их можно интерпретировать как *противоречивые высказывания*. Формулу, не являющуюся тождественно ложной, можно рассматривать как *выполнимое высказывание*. В случае, когда формула $A\supset B$ тождественно истинна, высказывание B можно интерпретировать как *логическое следствие* высказывания A. Если тождественно истинна формула $A\sim B$, то высказывания A и B можно интерпретировать как *логически равносильные*. Если формула $A\sim B$ – тождественно ложна, то высказывания A и B можно интерпретировать как *логически несовместимые*, в противном случае – как *логически совместимые*, иначе говоря, как высказывания, имеющие общее содержание.

Рассмотрим пример логического использования понятия уравнения алгебры конечных предикатов. Решим средствами алгебры конечных предикатов одну простую задачу. Пусть задано, что $x\in\{a, b, c\}$ $x\ne a$, $x\ne b$. Требуется определить значение буквенной переменной x. Человек, лишь взглянув на условие задачи, мгновенно находит: $x=c$. Алгебраическое же решение данной задачи требует определенной затраты усилий. Условие $x\in\{a, b, c\}$ означает, что $x=a$ или $x=b$, или $x=c$. Оно формально запишется в виде канонического уравнения $x^a\vee x^b\vee x^c=1$. Условия $x\ne a$ и $x\ne b$ на языке алгебры конечных предикатов запишутся в виде уравнений $\overline{x^a}=1$, $\overline{x^b}=1$. Все вместе условия задачи запишутся в виде уравнения:

$$(x^a\vee x^b\vee x^c)\,\overline{x^a}\;\overline{x^b}=1.$$

Упрощаем левую часть этого уравнения:

$$(x^a\vee x^b\vee x^c)\,\overline{x^a}\;\overline{x^b}\,x^b\equiv x^a\,\overline{x^a}\;\overline{x^b}\vee x^b\,\overline{x^a}\;\overline{x^b}\vee x^c\,\overline{x^a}\;\overline{x^b}\equiv \equiv 0\vee 0\vee x^c\equiv x^c.$$

Таким образом, $x^c=1$, откуда находим $x=c$. Рассмотрим другой пример на ту же тему: доказать, что из $x\in\{a, b, c\}$, $x\ne a$, $x\ne b$ следует $x=c$. Чтобы решить эту задачу, записываем формулу, соответствующую утверждению, которое требуется доказать:

$$(x^a\vee x^b\vee x^c)\,\overline{x^a}\;\overline{x^b}\supset x^c.$$

Производим упрощение этой формулы

$$(x^a\vee x^b\vee x^c)\;\overline{x^a}\;\overline{x^b}\supset x^c\equiv x^c\supset x^c\equiv 1.$$

Таким образом, формула тождественно истинна, а значит, утверждение $x=c$ следует из условия $x\in\{a, b, c\}$, $x\ne a$, $x\ne b$.

Приведенные примеры могут служить иллюстрацией к следующему общему утверждению: процесс решения уравнений и определения значения формул алгебры конечных предикатов мож-

но интерпретировать как *мышление интеллекта*, являющееся основой интеллектуальной деятельности человека. Машинное мышление возможно в той же мере, в какой возможно машинное решение уравнений и машинное определение значений формул алгебры конечных предикатов. Одной из важнейших задач теории интеллекта является разработка методов оперирования с формулами и уравнениями алгебры конечных предикатов, предназначенных для их машинной реализации.

2. Универсальная алгебра

До сих пор говорилось об алгебре конечных предикатов в единственном числе, на самом же деле была введена не одна, а целое семейство таких алгебр. Данное семейство включает в себя бесконечное число различных алгебр. Каждой паре – алфавиту букв $A=\{a_1, a_2,..., a_k\}$ и алфавиту переменных $B=\{x_1, x_2,..., x_n\}$ – соответствует отличная от других алгебра конечных предикатов. Всякий раз, прежде чем приступить к рассмотрению той или иной задачи, мы указывали алфавиты A и B, производя тем самым выбор конкретной алгебры конечных предикатов, на языке которой затем велось решение задачи. Указанный образ действий, однако, не всегда удобен, в частности при рассмотрении проблем, включающих в себя ряд взаимосвязанных задач. Если для каждой из таких задач выбрать наиболее экономную при заданных условиях алгебру конечных предикатов и на ее языке описывать решение задачи, то впоследствии могут возникнуть трудности, обусловленные «языковым барьером», при соединении в единое целое результатов, полученных при решении различных задач. Так, например, если формула A записана на языке одной алгебры конечных предикатов, а формула B – на языке другой, то система уравнений $\{A=1, B=1\}$, вообще говоря, теперь уже не равносильна уравнению $A \vee B=1$.

К такого рода проблемам относится проблема математического описания функций человеческого интеллекта. Интеллект человека – это очень сложная система, его функционирование не может быть описано «одним ударом». Теория интеллекта, без сомнения, будет развиваться постепенно, накапливая отдельные результаты. Чтобы в процессе развития не столкнуться с языковым барьером при объединении отдельных результатов в единую систему, очевидно, нужно с самого начала выбрать в качестве формального языка весьма мощную и единую для всех задач теории интеллекта алгебру, оперирующую достаточно большим числом букв и переменных. Тогда все объекты теории интеллекта можно будет описывать на одном и том же языке.

Однако возникает серьезное неудобство: всякий раз, описывая те или иные функции интеллекта, в том числе самые простые (а именно с простейшими функциями теория интеллекта будет иметь дело в первую очередь), придется использовать в качестве формального языка громоздкий математический аппарат, рассчитанный на применение к сложным объектам. Так, например, для записи СДНФ простейшего предиката придется ввести в действие весь арсенал букв и переменных. Такая СДНФ должна иметь невообразимо большую длину, оперировать ею будет практически невозможно. Вместе с тем, резервирование с самого начала огромного числа букв и переменных не спасает положения: сколь бы ни была обширна выбранная алгебра конечных предикатов, рано или поздно она станет недостаточной для удовлетворения постоянно растущих нужд развивающейся теории интеллекта. Таким образом, оба подхода – использование для каждой конкретной задачи собственной экономной специализированной алгебры и использование алгебры, общей для всех возможных в теории интеллекта задач, – обладают достоинствами и недостатками. Хотелось бы иметь такой третий подход, который, соединяя достоинства первых двух подходов, был бы свободен от присущих им недостатков. Такой подход существует. Он основан на использовании универсальной алгебры конечных предикатов, к рассмотрению которой мы и переходим

Теперь мы будем во всех случаях пользоваться единой алгеброй с алфавитом букв $A=\{a_1, a_2,..., a_\chi\}$ и алфавитом переменных $B=\{x_1, x_2,..., x_\nu\}$, называемой *универсальной алгеброй конечных предикатов*, особенность которой заключается в том ,что числа χ и ν букв и переменных в ее алфавитах всегда будут оставаться неопределенными. Вместе с тем, при практическом применении универсальной алгебры не должна возникать необходимость узнать точные значения этих чисел. О числах χ и ν будем считать известным лишь то, что они конечны и настолько велики, что удовлетворяют любым нуждам теории интеллекта, которые могут возникнуть даже в самом отдаленном будущем при ее развитии. При указанном подходе невозможно перечислить все буквы и переменные универсальной алгебры, однако такое перечисление нам никогда и не понадобится. Просто мы будем считать, что в алфавитах A и B содержатся все нужные знаки и в универсальной алгебре можно будет пользоваться любыми знаками без каких бы то ни было ограничений. Поскольку у отдельного человека (и даже у всего человечества в целом за всю историю его существования) может находиться в обращении лишь конечное число знаков, мы не вступаем в противоречие с требованием конечности алфавитов A и B универсальной алгебры.

Приступая к рассмотрению той или иной конкретной задачи теории интеллекта, будем каждый раз перечислять все переменные, которые мы собираемся в этой задаче использовать. Остальные

переменные универсальной алгебры, не указанные в этом перечне, будем молчаливо считать фиктивными. Обозначим переменные, введенные в данной конкретной задаче, через $x_1, x_2,..., x_n$. Для каждой такой переменной укажем область ее задания, то есть множество всех тех букв, которые эта переменная может принимать в данной задаче. С этой целью вводим следующую систему уравнений:

$$\begin{cases} x_1^{a_{11}} \vee x_1^{a_{21}} \vee ... \vee x_1^{a_{k_1 1}} = 1, \\ x_2^{a_{12}} \vee x_2^{a_{22}} \vee ... \vee x_2^{a_{k_2 2}} = 1, \\ ..... \\ x_n^{a_{1n}} \vee x_n^{a_{2n}} \vee ... \vee x_n^{a_{k_n n}} = 1. \end{cases} \quad (20)$$

Эти уравнения означают, что $x_1 \in \{a_{11}, a_{21},..., a_{k_1 1}\}$, $x_2 \in \{a_{12}, a_{12},..., a_{k_2 2}\},..., x_n \in \{a_{1n}, a_{2n},..., a_{k_n n}\}$; они задают для каждой переменной, участвующей в задаче, свою область изменения. Здесь $a_{11}, a_{21},..., a_{k_1 1}, a_{12}, a_{22},..., a_{k_2 2},..., a_{1n}, a_{2n},..., a_{k_n n}$ – буквы алфавита A, предназначенные для использования в задаче; $k_1, k_2,..., k_n$ – числа, не превышающие число χ. На систему уравнений (20) целесообразно смотреть, как на часть математического описания объекта, рассматриваемого в задаче. При решении задачи уравнения (20) должны учитываться наравне с другими фигурирующими в ней уравнениями.

Требование предварительного задания области изменения для каждой переменной, встречающейся в задаче, не более обременительно, чем необходимость указания алфавита букв при выборе специализированной алгебры конечных предикатов для той же задачи. Если область изменения некоторой переменной точно неизвестна, то всегда допустимо взять для этой переменной область задания «с запасом», включив в нее и такие значения, которые переменная, быть может, никогда не будет принимать. В частности, при желании можно для всех переменных, участвующих в задаче, принять одну и ту же область задания. В универсальной алгебре, ввиду невозможности перечисления всех букв алфавита A, практически не удается воспользоваться законом истинности:

$$x_j^{a_1} \vee x_j^{a_2} \vee ... \vee x_j^{a_\chi} = 1, \ (1 \le j \le n). \quad (21)$$

Указанное обстоятельство, однако, не приводит к затруднениям, так как в универсальной алгебре на смену системе тождеств (21) в каждой конкретной задаче приходит система (20) уравнений того же типа. Уравнения (20) назовем *усеченными законами истинности*.

В универсальной алгебре по той же причине нельзя практически применить и закон отрицания

$$\overline{x_j^{a_i}} = x_j^{a_1} \vee x_j^{a_2} \vee ... \vee x_j^{a_{i-1}} \vee x_j^{a_{i+1}} \vee ... \vee x_j^{a_\chi}, \quad (1 \le j \le n), \quad (22)$$

лежащий в основе определения операции отрицания. Однако это не ведет к трудностям, поскольку в универсальной алгебре на смену законам отрицания приходит система равенств

$$\overline{x_j^{\sigma_{ij}}} = x_j^{\sigma_{1j}} \vee x_j^{\sigma_{2j}} \vee ... \vee x_j^{\sigma_{i-1,j}} \vee \vee x_j^{\sigma_{i+1,j}} \vee ... \vee x_j^{\sigma_{k_j,j}}, \quad (23)$$

легко выводимых из усеченных законов истинности. Здесь i пробегает значения в пределах от 1 до k_j, j – в пределах от 1 до n. Уравнения (23) назовем *усеченными законами отрицания*. Они дают возможность ввести операцию отрицания также и в универсальной алгебре, в которой можно без каких бы то ни было ограничений пользоваться всеми тождествами алгебры конечных предикатов, кроме законов истинности и отрицания. Все законы ложности сохраняют силу, однако в конкретной задаче фактически используются только те, в которых фигурируют введенные в задаче переменные и показатели узнавания при них из областей изменения для данных переменных.

Как было уже отмечено выше, в универсальной алгебре невозможно практически воспользоваться совершенными дизъюнктивными и конъюнктивными нормальными формами – важным инструментом алгебры конечных предикатов. К счастью, им на смену приходят усеченные формы, к рассмотрению которых мы и приступаем. Изучение вопроса начнем с введения *смешанных* n-разрядных *числовых кодов* или кодов типа $(k_1, k_2,..., k_n)$. *Типом кода* назовем набор чисел $(k_1, k_2,..., k_n)$. Введем n множеств символов $R_j = \{0, 1,..., k_{j-1}\}$, $(1 \le j \le n)$. Символы множества R_j назовем k_j-ичными цифрами j-го разряда, под этими символами понимаем первые k_j чисел натурального ряда. Число k_j назовем основанием j-го разряда. При построении смешанных кодов в их j-м разряде разрешается использовать лишь цифры из множества R_j. Каждый код: $a_1 a_2 ... a_{n-1} a_n$ обозначает некоторое число, называемое значением этого кода и вычисляемое по формуле:

$$a_1 a_2 ... a_{n-1} a_n = a_1 k_2 ... k_n + a_2 k_3 ... k_n + ... + a_{n-1} k_n + a_n. \quad (24)$$

В качестве примера найдем с помощью формулы (24) числовое значение смешанного кода 122 типа (2, 4, 5). Десятичная запись значения этого кода равна $122_{(2,\,4,\,5)} = 1 \cdot 4 \cdot 5 + 2 \cdot 5 + 2 = 32_{10}$. Справа внизу у смешанного кода 122 указан его тип (2, 4, 5).

Для нахождения смешанного кода типа $(k_1, k_2,..., k_n)$ заданного числа нужно разделить это число на k_n, полученную целую часть частного разделить на k_{n-1} и т. д., наконец, целую часть частного n-1-го деления разделить на k_1. Считывая остатки этих делений в обратном порядке, получаем смешанный код заданного числа. В целой части частного после n-го деления должен получиться 0, в противном случае для заданного числа код типа $(k_1, k_2,..., k_n)$ не существует. В качестве примера найдем код

типа (4, 3, 5) числа, заданного десятичным кодом 38. Выполняем последовательные деления, остатки делений указываем в скобках: 38:5=7(3), 7:3=2(1), 2:4=0(2). Имеем $38_{10}=213_{(4,3,5)}$. Заметим, что число $L(k_1, k_2,..., k_n)$ всех различных n-разрядных кодов типа $(k_1, k_2,..., k_n)$ определяется формулой:

$$L(k_1, k_2,..., k_n)=k_1k_2...k_n. \qquad (25)$$

При использовании универсальной алгебры описание объектов той или иной конкретной задачи осуществляется с помощью *усеченных форм*, то есть таких формул алгебры конечных предикатов, в которых встречаются не все буквы и переменные универсальной алгебры, а только их часть, указанная в условии задачи. Предикат, заданный усеченной формой, можно представить в виде *усеченной таблицы* его значений, которая составляется только для тех переменных и их значений, которые указаны в условии рассматриваемой задачи. Каждая переменная x_j при этом пробегает все значения из заданной области $T_j=\{a_{1j}, a_{2j},..., a_{k_j j}\}$. Наборы значений аргументов, указанные в задаче (назовем их *усеченными наборами*), $(x_1, x_2,..., x_n)$ удобно интерпретировать как n-разрядные смешанные числовые коды типа $(k_1, k_2,..., k_n)$. Буквы $a_{1j}, a_{2j},..., a_{k_j j}$, расположенные на j-ом месте усеченного набора, интерпретируем как числа 0, 1,..., k_j-1. Заметим, что одни и те же буквы, стоящие на разных местах в наборе, могут интерпретироваться как различные числа.

В усеченной таблице наборы $x_1, x_2,..., x_n$ будем располагать в порядке возрастания числовых значений соответствующих этим наборам смешанных кодов. Порядок расположения кодов назовем *лексикографическим*. Усеченную таблицу значений иногда удобно рассматривать как полную таблицу некоторого конечного предиката, заданного на декартовом произведении $T_1\times T_2\times...\times T_n$. Указанный предикат будем называть *усеченным конечным предикатом* или усечением исходного предиката универсальной алгебры. Усеченные предикаты могут быть все пронумерованы, для чего воспользуемся способом, описанным в п. 1.1. Всего существует $2^{k_1k_2\cdots k_n}$ различных предикатов, заданных на декартовом произведении $T_1\times T_2\times...\times T_n$.

Универсальную алгебру, вместе с введенными в ней усеченными законами истинности и отрицания, можно рассматривать как некую разновидность алгебры конечных предикатов. Назовем такую алгебру *усеченной*. Уравнения (20) играют в ней роль тождеств этой алгебры. В усеченной алгебре сохраняют силу все понятия и результаты, рассмотренные в [1], в том числе понятия СДНФ и СКНФ. Отличие усеченной алгебры от ранее рассмотренной алгебры конечных предикатов состоит в том, что в ней каждая буквенная переменная определена на своей собственной области. В неусеченной же алгебре все переменные заданы на единой области. СДНФ и СКНФ усеченной алгебры можно использовать в качестве заменителей недосягаемых СДНФ и СКНФ универсальной алгебры при рассмотрении той или иной конкретной задачи, приняв их в качестве *усеченных СДНФ* и *СКНФ* универсальной алгебры.

Аналогично все другие понятия и результаты усеченной алгебры конечных предикатов могут быть приняты в качестве «усеченных» понятий и результатов универсальной алгебры. Понятия и результаты усеченной алгебры могут быть получены тем же путем, что и в неусеченной алгебре с той разницей, что вместо законов истинности и отрицания нужно везде применять усеченные законы истинности и отрицания. Универсальной алгеброй можно пользоваться при решении многих произвольных взаимосвязанных друг с другом задач столь же свободно, как и конкретной алгеброй конечных предикатов, используемой для решения какой-нибудь одной задачи. Единственной платой за полученную свободу служит необходимость введения в каждой конкретной задаче дополнительных уравнений – усеченных законов истинности, ограничивающих области изменения всех переменных, фигурирующих в данной задаче.

Рассмотрим пример пользования усеченными формами. Заданы следующие усеченные законы истинности:

$$x_1{}^a\vee x_1{}^b=1,\ x_2{}^c\vee x_2{}^d=1,\ x_3{}^a\vee x_3{}^d=1. \qquad (а)$$

Требуется преобразовать к усеченной СДНФ формулу:

$$f=(x_1{}^a\vee x_2{}^d)(x_1{}^a\vee x_1{}^b x_3{}^a)\vee(x_1{}^a\vee x_2{}^c)(x_2{}^d x_3{}^d\vee x_2{}^c x_3{}^a). \qquad (б)$$

Преобразование производим, руководствуясь алгоритмом преобразования к СДНФ произвольной формулы алгебры конечных предикатов, описанным в п. 3 [1]. Раскрывая скобки и производя упрощения, имеем:

$$f=x_1{}^a\vee x_1{}^a x_2{}^d\vee x_1{}^b x_2{}^d x_3{}^a\vee x_1{}^a x_2{}^d x_3{}^d\vee x_1{}^a x_2{}^c x_3{}^a\vee x_2{}^c x_3{}^a.$$

Во все конъюнкции вводим недостающие переменные, при этом, вопреки указаниям, содержащимся в описании алгоритма, пользуемся не законами истинности, а равенствами (а):

$$f=x_1{}^a(x_2{}^c\vee x_2{}^d)(x_3{}^a\vee x_3{}^d)\vee x_1{}^a x_2{}^d(x_3{}^a\vee x_3{}^d)\vee x_1{}^b x_2{}^d x_3{}^a\vee$$
$$\vee x_1{}^a x_2{}^d x_3{}^d\vee x_1{}^a x_2{}^c x_3{}^a\vee(x_1{}^a\vee x_1{}^b)x_2{}^c x_3{}^a.$$

Раскрывая скобки и производя упрощения, получаем усеченную СДНФ:

$$f=x_1{}^a x_2{}^c x_3{}^a\vee x_1{}^a x_2{}^c x_3{}^d\vee x_1{}^a x_2{}^d x_3{}^a\vee x_1{}^a x_2{}^d x_3{}^d\vee$$
$$\vee x_1{}^b x_2{}^c x_3{}^a\vee x_1{}^b x_2{}^d x_3{}^a.$$

Важным свойством усеченной СДНФ является то, что она указывает все решения системы уравнений, математически описывающей объект в кон-

кретной рассматриваемой задаче. В рассмотренном примере СДНФ указывает множество {(*a*, *c*, *a*), (*a*, *c*, *d*), (*a*, *d*, *a*), (*a*, *d*, *d*), (*d*, *c*, *a*), (*b*, *d*, *a*)} всех решений системы, состоящей из уравнения (б) и уравнений (а).

3. Аналитическое представление алфавитных операторов в виде уравнений

Формулы универсальной алгебры – удобное средство для аналитического представления конечных алфавитных операторов, нашего главного орудия математического описания деятельности интеллекта. Можно предложить несколько различных методов аналитического представления конечных алфавитных операторов. Их целесообразно разделить на две группы: методы неявного и явного задания операторов. Рассмотрим два метода неявного задания алфавитных операторов и один метод явного задания. Метод, который мы рассмотрим первым, позволяет осуществить *неявное задание алфавитного оператора одним уравнением.* Пусть $y_1y_2...y_q=F(x_1x_2...x_p)$ – произвольно выбранный конечный алфавитный оператор, преобразующий входные слова $x_1x_2...x_p$ длины p в выходные слова $y_1y_2...y_q$ длины q. Чтобы получить в универсальной алгебре корректное математическое описание объекта, нужно ограничить области изменения всех буквенных переменных, относящихся к этому объекту. В данном случае объектом математического описания служит алфавитный оператор F, в роли буквенных переменных выступают символы $x_1, x_2,..., x_p, y_1, y_2,..., y_q$.

Требуемые ограничения вводим с помощью усеченных законов истинности для буквенных переменных входного слова:

$$\begin{cases} x_1^{a_{11}} \vee x_1^{a_{21}} \vee ... \vee x_1^{a_{k_1 1}} = 1, \\ x_2^{a_{12}} \vee x_2^{a_{22}} \vee ... \vee x_2^{a_{k_2 2}} = 1, \\ ..... \\ x_p^{a_{1p}} \vee x_p^{a_{2p}} \vee ... \vee x_p^{a_{k_p p}} = 1; \end{cases} \quad (26)$$

для буквенных переменных выходного слова:

$$\begin{cases} y_1^{b_{11}} \vee y_1^{b_{21}} \vee ... \vee y_1^{b_{l_1 1}} = 1, \\ y_2^{b_{12}} \vee y_2^{b_{22}} \vee ... \vee y_2^{b_{l_2 2}} = 1, \\ ..... \\ y_q^{b_{1q}} \vee y_q^{b_{2q}} \vee ... \vee y_q^{b_{l_q q}} = 1. \end{cases} \quad (27)$$

Смысл указанных уравнений состоит в том, что $x_1 \in \{a_{11}, a_{21},..., a_{k_1 1}\}$, $x_2 \in \{a_{12}, a_{22},..., a_{k_2 2}\}$,..., $x_p \in \{a_{1p}, a_{2p},..., a_{k_p p}\}$, $y_1 \in \{b_{11}, b_{21},..., b_{l_1 1}\}$, $y_2 \in \{b_{12}, b_{22},..., b_{l_2 2}\}$,..., $y_q \in \{b_{1q}, b_{2q},..., b_{l_q q}\}$.

Построим усеченный конечный предикат $f(x_1, x_2,..., x_p, y_1, y_2,..., y_q)$ с областью определения $S_1 \times S_2 \times ... \times S_p \times T_1 \times T_2 \times ... \times T_q$, где $S_i=\{a_{1i}, a_{2i},..., a_{k_i i}\}$, $T_j=\{b_{1j}, b_{2j},..., b_{l_j j}\}$, $(1 \le i \le p, 1 \le j \le q)$, задавая его значения следующим правилом:

$$f(x_1, x_2,..., x_p, y_1, y_2,..., y_q) = \begin{cases} 1, \text{если } y_1y_2...y_q = F(x_1x_2...x_p), \\ 0, \text{если } y_1y_2...y_q \ne F(x_1x_2...x_p). \end{cases} \quad (28)$$

Запишем уравнение

$$\bigvee_{F(\sigma_1\sigma_2...\sigma_p)=\varepsilon_1\varepsilon_2...\varepsilon_q} x_1^{\sigma_1} x_2^{\sigma_2} ... x_p^{\sigma_p} y_1^{\varepsilon_1} x_2^{\varepsilon_2} ... x_q^{\varepsilon_q} = 1, \quad (29)$$

левая часть которого совпадает с усеченной СДНФ предиката f. Запись $F(\sigma_1,\sigma_2,...,\sigma_p)=\varepsilon_1\varepsilon_2...\varepsilon_q$ под знаком $\vee\vee$ означает, что логическое суммирование ведется по тем буквенным наборам $(\sigma_1,\sigma_2,...,\sigma_p,\varepsilon_1\varepsilon_2...\varepsilon_q)$, для которых имеет место равенство $F(\sigma_1,\sigma_2,...,\sigma_p)=\varepsilon_1\varepsilon_2...\varepsilon_q$. Уравнение (29) можно принять в качестве математической записи конечного алфавитного оператора F. Действительно, подставляя в него вместо $x_1, x_2,..., x_p$ соответственно буквы $\sigma_1, \sigma_2,..., \sigma_p$ из областей $S_1, S_2,..., S_p$ и решая уравнение (29) относительно набора переменных $(y_1, y_2,..., y_q)$ получаем единственное решение $(\varepsilon_1, \varepsilon_2,..., \varepsilon_q)$. Составленное из букв этого решения слово $\varepsilon_1\varepsilon_2...\varepsilon_q$ совпадает с выходным словом оператора F для входного слова $\sigma_1\sigma_2...\sigma_p$.

Отметим, что уравнение (29) не только задает реакции оператора F в области его определения $S_1 \times S_2 \times ... \times S_p$, но, кроме того, формирует саму область определения. Если мы захотим найти реакцию оператора F на входное слово, находящееся за пределами области определения оператора, то, подставив буквы этого слова в (29), получим противоречие: 0=1. Следовательно, в данном случае указанное уравнение не имеет решений. Таким образом, характеризуемый им оператор не ставит в соответствие заданному входному слову никакого выходного слова, иначе говоря, алфавитный оператор в этом случае не определен. Если представление конечного алфавитного оператора F в виде усеченной СДНФ нас чем-то не устраивает, то можно путем тождественных преобразований левой части уравнения (29) перейти к любому другому формальному представлению этого уравнения. В процессе тождественных преобразований, поскольку здесь мы имеем дело с усеченными формулами универсальной алгебры, следует вместо закона истинности пользоваться равенствами (26) и (27). Если в процессе тождественных преобразований левой части уравнения (29) приходится использовать какие-либо из равенств (26) и (27), то преобразованное уравнение уже нельзя считать полной записью оператора F. Для полноты представления оператора F следует к преобразованному уравнению присовокупить те из уравнений (26), (27), которые были использованы при тождественных преобразованиях.

Рассмотрим пример аналитического представления конечного алфавитного оператора по первому методу. Пусть $y_1y_2y_3=F(x_1x_2)$ – алфавитный

оператор, который ставит в соответствие троичным цифрам x_1, x_2 их сумму в виде трехразрядного двоичного кода $y_1y_2y_3$. Требуется записать в виде уравнения универсальной алгебры алфавитный оператор F. Записываем уравнения, ограничивающие значения введенных переменных:

$$x_1^{\,0} \vee x_1^{\,2} \vee x_1^{\,2} = 1,\ x_2^{\,0} \vee x_2^{\,1} \vee x_2^{\,2} = 1,$$
$$x_1^{\,0} \vee x_1^{\,1} = 1,\ x_2^{\,0} \vee x_2^{\,1} = 1,\ x_3^{\,0} \vee x_3^{\,1}. \qquad \text{(а)}$$

Составляем таблицу соответствия между входными словами x_1x_2 и выходными словами $y_1y_2y_3$, задаваемого оператором F (табл. 1).

Таблица 1

x_1	0	0	0	1	1	1	2	2	2
x_2	0	1	2	0	1	2	0	1	2
y_1	0	0	0	0	0	0	0	0	1
y_2	0	0	1	0	1	1	1	1	0
y_3	0	1	0	1	0	1	0	1	0

По таблице записываем уравнение (29), связывающее переменные x_1, x_2, y_1, y_2, y_3 точно так, как их связывает алфавитный оператор F:

$$x_1^{\,0}x_2^{\,0}y_1^{\,0}y_2^{\,0}y_3^{\,0} \vee x_1^{\,0}x_2^{\,1}y_1^{\,0}y_2^{\,0}y_3^{\,1} \vee x_1^{\,0}x_2^{\,2}y_1^{\,0}y_2^{\,1}y_3^{\,0} \vee$$
$$\vee x_1^{\,1}x_2^{\,0}y_1^{\,0}y_2^{\,0}y_3^{\,1} \vee x_1^{\,1}x_2^{\,1}y_1^{\,0}y_2^{\,1}y_3^{\,0} \vee x_1^{\,1}x_2^{\,2}y_1^{\,0}y_2^{\,1}y_3^{\,1} \vee \text{(б)}$$
$$\vee x_1^{\,2}x_2^{\,0}y_1^{\,0}y_2^{\,1}y_3^{\,0} \vee x_1^{\,2}x_2^{\,1}y_1^{\,0}y_2^{\,1}y_3^{\,1} \vee x_1^{\,2}x_2^{\,2}y_1^{\,1}y_2^{\,0}y_3^{\,0} = 1.$$

В левой части уравнения (б) записана усеченная СДНФ предиката f, определяемая по оператору F соотношениями (28). Пользуясь первым законом дистрибутивности (8), уравнение (б) можно представить в более компактном виде, переходя к скобочной форме:

$$((x_1^{\,0}y_2^{\,0} \vee x_1^{\,2}y_2^{\,1})(x_2^{\,0}y_3^{\,0} \vee x_2^{\,1}y_3^{\,1}) \vee$$
$$\vee (x_1^{\,0}x_2^{\,2} \vee x_1^{\,1}x_2^{\,1})y_2^{\,1}y_3^{\,0} \vee \qquad \text{(в)}$$
$$(x_2^{\,0}y_2^{\,0} \vee x_2^{\,2}y_2^{\,1})x_1^{\,1}y_3^{\,1})y_1^{\,0} \vee x_1^{\,2}x_2^{\,2}y_1^{\,1}y_2^{\,0}y_3^{\,0})) = 1.$$

Дополнять уравнение (в) уравнениями (а) в данном случае нет необходимости, поскольку они не приняли участия при преобразовании уравнения (б) в уравнение (в). Обратим внимание на то, что формульное представление предиката f, полученное в примере, неизмеримо компактнее табличного. Усеченная таблица значений предиката f должна была бы содержать $3{\cdot}3{\cdot}2{\cdot}2{\cdot}2{=}72$ столбца. Уравнение же, выражающее предикат f, содержит всего 26 узнаваний букв. Определим, например, с помощью уравнения (в) выходное слово алфавитного оператора F для входного слова 12. Имеем $x_1{=}1$, $x_2{=}2$. Подставляем значения x_1.и x_2 в уравнение (в):

$$((1^0y_2^{\,0} \vee 1^2y_2^{\,1})(2^0y_3^{\,0} \vee 2^1y_3^{\,1}) \vee (1^02^2 \vee 1^12^1)y_2^{\,1}y_3^{\,0} \vee (2^0y_2^{\,0} \vee$$
$$\vee 2^2y_2^{\,1})1^1y_3^{\,1})y_1^{\,0} \vee 1^22^2y_1^{\,1}y_2^{\,0}y_3^{\,0}))=1.$$

Из полученного уравнения исключаем константы. В результате получаем уравнение $y_1^{\,0}y_2^{\,1}y_3^{\,1}{=}1$, откуда находим: $y_1^{\,0}{=}1$, $y_2^{\,1}{=}1$, $y_3^{\,1}{=}1$. Таким образом, $y_1{=}0$, $y_2{=}1$, $y_3{=}1$. Выходное слово равно 011.

Располагая уравнением, описывающим алфавитный оператор F, можно решать обратную задачу: по заданному выходному слову отыскать соответствующее ему для оператора F входное слово. Рассмотрим пример. Для алфавитного оператора, введенного в предыдущем примере, по выходному слову 100 требуется найти соответствующее ему входное слово. Имеем $y_1{=}1$, $y_2{=}0$, $y_3{=}0$. Подставляем значения y_1, y_2, y_3 в уравнение (в) и производим упрощения. В результате получаем: $x_1^{\,2}x_2^{\,2}{=}1$, входное слово равно 22. Если в качестве выходного слова взять слово 010, то для него уравнение (в) принимает вид:

$$x_1^{\,2}x_3^{\,0} \vee x_1^{\,0}x_2^{\,2} \vee x_1^{\,1}x_2^{\,1} = 1.$$

Согласно последнему уравнению, выходному слову 010 соответствует три входных слова: 20, 02 и 11. Беря же в качестве выходного слова 101 и подставляя $y_1{=}1$, $y_2{=}0$, $y_3{=}1$ в уравнение (в), получаем в его левой части 0; мы приходим к равенству $0{=}1$, то есть к противоречию. Следовательно, не существует ни одного входного слова, которому бы алфавитный оператор F ставил в соответствие выходное слово 101.

Уравнение универсальной алгебры, описывающее алфавитный оператор F, можно использовать при решении различного рода логических задач, в которых фигурирует оператор F. Рассмотрим пример одной из указанных задач. В ней для того же алфавитного оператора, который рассматривался в предыдущих примерах параграфа, требуется найти входное и выходное слова, если известно, что обе буквы входного слова одинаковы, а первые и последние буквы выходного слова различны. По условию имеем $x_1{=}x_2$ и $y_1{\neq}y_3$. Условие $x_1{=}x_2$ запишем в виде высказывания: «$x_1{=}0$ и $x_2{=}0$, или $x_1{=}1$ и $x_2{=}1$, или $x_1{=}2$ и $x_2{=}2$», которое, в свою очередь, математически описывается следующим уравнением универсальной алгебры:

$$x_1^{\,0}x_2^{\,0} \vee x_1^{\,1}x_2^{\,1} \vee x_1^{\,2}x_2^{\,2} = 1. \qquad \text{(г)}$$

Условие $y_1{\neq}y_3$ представим в виде равносильного ему высказывания «$y_1{=}1$ и $y_3{=}0$, или $y_1{=}0$, $y_3{=}1$» а затем – в виде уравнения

$$y_1^{\,1}y_3^{\,0} \vee y_1^{\,0}y_3^{\,1} = 1. \qquad \text{(д)}$$

Решим совместно уравнения (б), (г) и (д). Для этого запишем все три уравнения в виде единого канонического уравнения:

$$(x_1^{\,0}x_2^{\,0}y_1^{\,0}y_2^{\,0}y_3^{\,0} \vee x_1^{\,0}x_2^{\,1}y_1^{\,0}y_2^{\,0}y_3^{\,1} \vee x_1^{\,0}x_2^{\,2}y_1^{\,0}y_2^{\,1}y_3^{\,0} \vee$$
$$\vee x_1^{\,1}x_2^{\,0}y_1^{\,0}y_2^{\,0}y_3^{\,1} \vee$$
$$\vee x_1^{\,1}x_2^{\,1}y_1^{\,0}y_2^{\,1}y_3^{\,0} \vee x_1^{\,1}x_2^{\,2}y_1^{\,0}y_2^{\,1}y_3^{\,1} \vee x_1^{\,2}x_2^{\,0}y_1^{\,0}y_2^{\,1}y_3^{\,0} \vee$$
$$\vee x_1^{\,2}x_2^{\,1}y_1^{\,0}y_2^{\,1}y_3^{\,1} \vee x_1^{\,2}x_2^{\,2}y_1^{\,1}y_2^{\,0}y_3^{\,0})(x_1^{\,0}x_2^{\,0} \vee x_1^{\,1}x_2^{\,1} \vee$$
$$\vee x_1^{\,2}x_2^{\,2})(y_1^{\,1}y_3^{\,0} \vee y_1^{\,0}y_3^{\,1}) = 1,$$

в левой части которого раскрываем скобки и производим упрощения, пользуясь законами ложности [1, (19)]. В результате получаем $x_1^2x_2^2y_1^1y_2^0y_3^0=1$, откуда находим $x_1=x_2=2$, $y_1=1$, $y_2=y_3=0$. Таким образом, условию задачи удовлетворяет единственная пара слов: входное слово 22 и соответствующее ему выходное слово 100 алфавитного оператора F.

Перейдем к рассмотрению второго метода представления алфавитных операторов. Метод позволяет осуществить *неявное задание алфавитного оператора системой уравнений*, число которых совпадает с числом букв выходного слова этого оператора. Когда указывают некоторый конечный алфавитный оператор $y_1y_2...y_q=F(x_1, x_2,..., x_p)$, то тем самым задают систему

$$\begin{cases} y_1 = F_1(x_1x_2...x_p), \\ y_2 = F_q(x_1x_2...x_p), \\ ..... \\ y_q = F_q(x_1x_2...x_p) \end{cases} \qquad (30)$$

алфавитных операторов F_1, F_2, ..., F_q, каждый из которых формирует в зависимости от входного слова $x_1x_2...x_p$ соответственно одну букву y_1, y_2, ..., y_q выходного слова $y_1y_2...y_q$ алфавитного оператора F. Записывая для каждого из алфавитных операторов (30) соответствующее ему уравнение (29) универсальной алгебры, получаем аналитическое представление алфавитного оператора F в виде следующей системы, состоящей из q уравнений:

$$\begin{cases} \bigvee\limits_{F(\sigma_1\sigma_2...\sigma_p)=\varepsilon_1} x_1^{\sigma_1}x_2^{\sigma_2}...x_p^{\sigma_p}y_1^{\varepsilon_1}=1, \\ \bigvee\limits_{F(\sigma_1\sigma_2...\sigma_p)=\varepsilon_2} x_1^{\sigma_1}x_2^{\sigma_2}...x_p^{\sigma_p}y_2^{\varepsilon_2}=1, \\ ..... \\ \bigvee\limits_{F(\sigma_1\sigma_2...\sigma_p)=\varepsilon_q} x_1^{\sigma_1}x_2^{\sigma_2}...x_p^{\sigma_p}y_q^{\varepsilon_q}=1. \end{cases} \qquad (31)$$

Рассмотрим пример аналитического представления алфавитного оператора по второму методу. Описываем тот же оператор, что и в предыдущих примерах. По табл. 1 составляем три уравнения типа (31):

$$\begin{gathered} x_1^0x_2^0y_1^0 \vee x_1^0x_2^1y_1^0 \vee x_1^0x_2^2y_1^0 \vee x_1^1x_2^0y_1^0 \vee \\ \vee x_1^1x_2^1y_1^0 \vee x_1^1x_2^2y_1^0 \vee x_1^2x_2^0y_1^0 \vee \\ x_1^2x_2^1y_1^0 \vee x_1^2x_2^2y_1^1=1, \\ x_1^0x_2^0y_2^0 \vee x_1^0x_2^1y_2^0 \vee x_1^0x_2^2y_2^1 \vee x_1^1x_2^0y_2^0 \vee \\ \vee x_1^1x_2^1y_2^1 \vee x_1^1x_2^2y_2^1 \vee x_1^2x_2^0y_2^1 \vee \\ \vee x_1^2x_2^1y_2^1 \vee x_1^2x_2^2y_2^0=1, \\ x_1^0x_2^0y_3^0 \vee x_1^0x_2^1y_3^1 \vee x_1^0x_2^2y_3^0 \vee x_1^1x_2^0y_3^1 \vee \\ \vee x_1^1x_2^1y_3^0 \vee x_1^1x_2^2y_3^1 \vee \\ \vee x_1^2x_2^0y_3^0 \vee x_1^2x_2^1y_3^1 \vee x_1^2x_2^2y_3^0=1. \end{gathered} \qquad (е)$$

Записанная система уравнений равносильна уравнению (б), полученному для того же алфавитного оператора по первому методу. При сравнении описаний (б) и (е) видим, что в данном конкретном примере первый метод дает более простое представление алфавитного оператора, чем второй (45 узнаваний буквы против 81), однако каждое из уравнений (е) проще, чем уравнение (б) (27 узнаваний буквы против 45).

Заметим, что нетрудно перейти от описания алфавитного оператора одним уравнением к описанию того же оператора системой уравнений. Для получения описания функции $y_i=F(x_1x_2...x_p)$, где $1\le i\le q$, нужно все узнавания букв для переменных $y_1,..., y_{i-1}, y_{i+1},..., y_q$ в исходном уравнении положить равными 1. Например, подставляя 1 в уравнение (б) вместо y_2^0, y_2^1, y_3^0, y_3^1, находим первое из уравнений (е). Для перехода от описания алфавитного оператора системой уравнений к описанию того же оператора одним уравнением надо образовать конъюнкцию всех уравнений системы. Например, образуя конъюнкцию левых частей всех уравнений (е) и приравнивая ее логическому значению 1, выведем уравнение, равносильное уравнению (б). Попытаемся упростить левые части уравнений (е). Осуществляя в них вынос за скобки некоторых из узнаваний и применяя усеченные законы истинности (а) и вытекающие из них усеченные законы отрицания, а также используя сокращения с помощью знаков ~ и $\oplus$, перепишем систему (е) в более компактном виде:

$$\begin{gathered} x_1^2x_2^2 \sim y_1^1=1,\ (x_1^2 \oplus x_2^2) \vee x_1^2x_2^2 \sim y_2^1=1, \\ (x_1^2 \oplus x_2^2) \sim y_3^1=1. \end{gathered} \qquad (ж)$$

Сравнивая систему (ж) с уравнением (в), видим, что в результате преобразования выражений, полученных вторым методом, мы пришли в данном примере к более компактному результату, чем тот, который был достигнут на базе первого метода (11 узнаваний буквы против 26). Правда, систему (ж) можно считать полным описанием заданного алфавитного оператора только при условии присоединения к ней системы уравнений (а), а это добавляет еще 12 узнаваний букв. Заметим, что без предварительного перехода к описанию алфавитного оператора F системой уравнений было бы затруднительно перейти от уравнения (в) к системе (ж). Таким образом, использование второго метода неявного представления алфавитного оператора в данном случае привело, в конечном счете, к отысканию более простой записи этого оператора. Практика решения одних и тех же задач обоими методами показывает, что в зависимости от характера задачи в одних случаях более предпочтительным оказывается первый метод, а в других – второй. Было бы интересно на базе изучения асимптотических оценок сложности формул выяснить вопрос о

рациональных областях применения двух этих методов. Представляет интерес и разработка других методов аналитического представления алфавитных операторов, в чем-то более предпочтительных в сравнении с только что описанными методами.

Рассмотрим метод *явного задания алфавитного оператора*. Выражения, которые получаются при явном задании алфавитных операторов, имеют, как правило, большую суммарную длину, чем выражения неявного задания, однако они незаменимы для построения переключательных цепей, реализующие конечные алфавитные операторы. Явное задание алфавитного оператора можно рассматривать как формальную запись переключательной цепи, воспроизводящей реакции этого алфавитного оператора. Хранить же информацию об алфавитных операторах в памяти более удобно в форме выражений их неявного задания.

Сформулируем и докажем важное утверждение, которое назовем *теоремой о явном задании конечного алфавитного оператора*. Ограничим область изменения буквенной переменной y значениями b_1, b_2,..., b_l, то есть опишем ее уравнением

$$y^{b_1} \vee y^{b_2} \vee \ldots y^{b_l} = 1. \qquad (32)$$

Тогда *любой однозначный, вообще говоря, частичный, алфавитный оператор* $y = G(x_1, x_2, \ldots, x_m, y) = 1$, *заданный в неявной форме уравнением*

$$g(x_1, x_2, \ldots, x_m, y)=1, \qquad (33)$$

может быть представлен в явной форме с помощью следующей системы уравнений:

$$\begin{aligned} &g(x_1, x_2, \ldots, x_m, b_1) = y^{b_1}, \\ &g(x_1, x_2, \ldots, x_m, b_2) = y^{b_2}, \\ &\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\ &g(x_1, x_2, \ldots, x_m, b_l) = y^{b_l}. \end{aligned} \qquad (34)$$

Чтобы доказать только что сформулированное утверждение, установим, что система, состоящая из уравнений (32) и (33), равносильна системе уравнений (32) и (34). Зафиксируем произвольным образом значения переменных x_1, x_2,..., x_m, y. Предположим, что набор (x_1, x_2,..., x_m, y) является решением системы, состоящей из уравнений (32) и (33). Отсюда следует существование такого i($1 \le i \le l$), что $y=b_i$ и $g(x_1, x_2, \ldots, x_m, b_i)=1$. В силу однозначности алфавитного оператора G для любого $j \ne i$ ($1 \le j \le l$) имеем $g(x_1, x_2, \ldots, x_m, b_j)=0$. Поэтому все уравнения системы (34), кроме i-го, обращаются в тождества вида 0=0, i-е же уравнение обращается в тождество вида 1=1. Таким образом, набор (x_1, x_2,..., x_m, y) есть также решение системы, состоящей из уравнений (32) и (34).

Предположим теперь, что набор (x_1, x_2,..., x_m, y) не является решением системы уравнений (32) и (33). Если $y \in \{b_1, b_2, \ldots, b_l\}$, то условие (32) не выполняется, а вместе с ним не выполняется и система условий (32) и (34). Если же $y=b_i$ ($1 \le i \le l$), то не выполняется условие (33), поэтому $g(x_1, x_2, \ldots, x_m, b_i)=0$. В результате i-е уравнение системы (34) обращается в противоречие вида 0=1. Таким образом, набор (x_1, x_2,..., x_m, y) не является также решением системы уравнений (32) и (34). Следовательно, системы уравнений (32), (33) и (32), (34) равносильны друг другу. Теорема доказана. Отметим, что в случае, когда уравнением (33) задан многозначный в области значений (b_1, b_2,..., b_l)=1 алфавитный оператор G, системы уравнений (32), (33) и (32), (34) уже не будут равносильными друг другу. Действительно, предположим, что существуют два набора (x_1, x_2,..., x_m, b_i) и (x_1, x_2,..., x_m, b_j), причем $i \ne j$ и $1 \le i, j \le l$, удовлетворяющие уравнению (33). Однако ни один из этих наборов не удовлетворяет системе уравнений (34). Набор (x_1, x_2,..., x_m, b_i) обращает в противоречие вида 1=0 j-е уравнение системы (34), а набор (x_1, x_2,..., x_m, b_j) – i-е уравнение.

Зададим алфавитный оператор $y_1y_2...y_n=F(x_1x_2...x_m)$ системой уравнений:

$$\begin{aligned} &f_1(x_1, x_2, \ldots, x_m, y_1) = 1, \\ &f_2(x_1, x_2, \ldots, x_m, y_2) = 1, \\ &\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\ &f_n(x_1, x_2, \ldots, x_m, y_n) = 1. \end{aligned} \qquad (35)$$

Каждым из этих уравнений определяется значение соответствующей буквы выходного слова $y_1y_2...y_n$ алфавитного оператора F для входного слова $x_1x_2...x_m$. Тем самым уравнения (35) задают в неявной форме систему алфавитных операторов:

$$\begin{aligned} &y_1 = F_1(x_1x_2...x_m), \\ &y_1 = F_2(x_1x_2...x_m), \\ &\ldots\ldots\ldots\ldots\ldots\ldots \\ &y_n = F_n(x_1x_2...x_m), \end{aligned} \qquad (36)$$

формирующих отдельные буквы выходного слова алфавитного оператора F.

Переходя описанным выше методом к представлению в явной форме алфавитных операторов F_1, F_2, ..., F_n, получаем в явной форме аналитическое задание алфавитного оператора F: Здесь имеется в виду, что области изменения переменных y_i ($1 \le i \le n$) ограничены множествами букв $T_i = \{b_{i1}, b_{i2}, \ldots, b_{il_i}\}$, где l_i – число букв в множестве T_i. Всего в системе (37) содержится $l_1+l_2+...+l_n$ уравнений. Система (37) допускает непосредственное вычисление букв выходного слова $y_1y_2...y_n$ алфавитного оператора F по заданным значениям букв входного слова $x_1x_2...x_m$. Задание же алфавитного оператора F в виде системы (35) не дает такой возможности. В этом случае буквы выходного слова могут быть найдены только путем решения уравнения.

$$\left.\begin{array}{l} f_1(x_1,x_2,...,x_m,b_{11}) = y_1^{b_{11}}, \\ f_1(x_1,x_2,...,x_m,b_{12}) = y_1^{b_{12}}, \\ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\ f_1(x_1,x_2,...,x_m,b_{1l_1}) = y_1^{b_{1l_n}}, \\ f_2(x_1,x_2,...,x_m,b_{21}) = y_2^{b_{21}}, \\ f_2(x_1,x_2,...,x_m,b_{22}) = y_2^{b_{22}}, \\ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\ f_2(x_1,x_2,...,x_m,b_{2l_2}) = y_2^{b_{2l_n}}, \\ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\ f_n(x_1,x_2,...,x_m,b_{n1}) = y_n^{b_{n1}}, \\ f_n(x_1,x_2,...,x_m,b_{n2}) = y_n^{b_{n2}}, \\ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\ f_n(x_1,x_2,...,x_m,b_{nl_n}) = y_n^{b_{nl_n}}. \end{array}\right\} \quad (37)$$

Рассмотрим пример. В роли оператора F возьмем оператор $y_1y_2y_3=F(x_1x_2)$, который ставит в соответствие троичным цифрам x_1, x_2 их сумму в виде трехразрядного двоичного кода $y_1y_2y_3$. В предыдущем разделе было получено неявное описание этого оператора в виде системы (е). Требуется записать заданный алфавитный оператор F в явном виде. Значения переменных $y_1y_2y_3$ – двоичные, поэтому принимаем $T_1=T_2=T_3=\{0, 1\}$. Отправляясь от уравнений (е) и поочередно подставляя в них вместо переменных y_1, y_2, y_3 значения 0 и 1, после упрощения получаем следующую систему, задающую оператор $y_1y_2y_3=F(x_1x_2)$ в явном виде:

$$\begin{gathered} (x_1^0 \vee x_1^1)(x_2^0 \vee x_2^1 \vee x_2^2) \vee x_1^2(x_2^0 \vee x_2^1) = y_1^0; \\ x_1^2x_2^2 = y_1^1;\ x_1^0(x_2^0 \vee x_2^1) \vee x_1^1x_2^2 = y_2^0; \\ x_1^0x_2^2 \vee x_1^1(x_2^1 \vee x_2^2) \vee x_1^2(x_2^0 \vee x_2^1) = y_2^1; \\ (x_1^0 \vee x_1^2)(x_2^0 \vee x_2^2) \vee x_1^1x_2^1 = y_3^0; \\ x_1^1(x_2^0 \vee x_2^2) \vee (x_1^0 \vee x_1^2)x_2^1 = y_3^1. \end{gathered} \quad (з)$$

Уравнения (37) можно вывести непосредственно по таблице алфавитного оператора, минуя его неявную форму задания (35). С этой целью в левой части уравнений записываем усеченные СДНФ, составленные из всех конституэнт единицы, наборы показателей которых совпадают с входными словами, такими, что им соответствует одна и та же буква на заданном месте в выходных словах. В правой части уравнений пишем переменную, соответствующую заданному месту в выходных словах, с показателем, совпадающим с буквой, стоящей на заданном месте. Например, по табл. 1 для алфавитного оператора $y_1y_2y_3=F(x_1x_2)$, рассмотренного выше, записываем следующие уравнения типа (37):

$$\begin{gathered} x_1^0x_2^0 \vee x_1^0x_2^1 \vee x_1^0x_2^2 \vee x_1^1x_2^0 \vee x_1^1x_2^1 \vee x_1^1x_2^2 \vee \\ x_1^2x_2^0 \vee x_1^2x_2^1 = y_1^0;\ x_1^2x_2^2 = y_1^1; \\ x_1^0x_2^0 \vee x_1^0x_2^1 \vee x_1^1x_2^2 = y_2^0; \\ x_1^0x_2^2 \vee x_1^1x_2^1 \vee x_1^1x_2^2 \vee x_1^2x_2^0 \vee x_1^2x_2^1 = y_2^1; \\ x_1^0x_2^0 \vee x_1^0x_2^2 \vee x_1^1x_2^1 \vee x_1^2x_2^0 \vee x_1^1x_2^2 = y_3^0; \\ x_1^0x_2^1 \vee x_1^1x_2^0 \vee x_1^1x_2^2 \vee x_1^2x_2^1 = y_3^1. \end{gathered} \quad (и)$$

Из них путем тождественных преобразований могут быть получены уравнения (з).

Рассмотрим пример вычисления выходного слова $y_1y_2...y_n$ алфавитного оператора F по заданному входному слову $x_1x_2...x_m$. Определим двоичную сумму $y_1y_2y_3$ троичных слагаемых $x_1=1$, $x_2=2$ с помощью явного задания алфавитного оператора $y_1y_2y_3=F(x_1x_2)$. Подставляя в левые части равенств (з) или (и) заданные значения x_1, x_2, x_3, имеем $y_1^0=1$, $y_1^1=0$, $y_2^0=0$, $y_2^1=1$, $y_3^0=0$, $y_3^1=1$. Таким образом, $y_1y_2y_3=011$.

4. Декомпозиция уравнений

Одна из важных задач теории интеллекта состоит в том, чтобы научиться производить *декомпозицию уравнений* алгебры конечных предикатов, то есть замену одного сложного уравнения эквивалентной ему системой более простых уравнений. Такую декомпозицию ежеминутно производит человек, выражая свою мысль (сложное уравнение) в форме последовательности отдельных предложений (системы более простых уравнений). Декомпозиция уравнений важна как один из этапов процесса решения уравнений алгебры конечных предикатов, поскольку в ряде случаев систему простых уравнений решать значительно легче, чем эквивалентное ей одно сложное уравнение. Декомпозиция уравнений также может служить мощным средством упрощения и сокращения записи уравнений алгебры конечных предикатов.

Тот факт, что человек никогда не испытывает затруднений при выражении мыслей в виде последовательности сравнительно коротких высказываний, свидетельствует о том, что мысли обладают одной важной особенностью: вне зависимости от уровня своей сложности они допускают выражение в виде конъюнкции (состоящей, быть может, из очень большого числа конъюнктивных членов) достаточно простых высказываний. Указанное свойство человеческого интеллекта назовем *конъюнктивностью интеллекта*. Очевидно, что свойство конъюнктивности чрезвычайно ограничивает класс уравнений алгебры конечных предикатов, которые могут эффективно решаться человеческим интеллектом. В свете данного вывода выглядит поразительной способность человеческого интеллекта эффективно познавать окружающий мир. Мы полагаем, что эту способность можно удовлетвори-

тельным образом объяснить, лишь признав конъюнктивность самого физического мира, то есть наличие такой его структуры, которая допускает описание в виде конъюнкции достаточно простых высказываний. Конъюнктивность представляется нам одним из фундаментальных качеств человеческого интеллекта. Она указывает на особую важность для теории интеллекта задачи декомпозиции уравнений алгебры конечных предикатов.

Рассмотрим условия, при которых уравнение $A=1$ может быть представлено в виде эквивалентной ему системы уравнений $B=1$, $C=1$. Здесь A, B, C – формулы алгебры конечных предикатов. Очевидно, такое представление возможно в том и только том случае, когда

$$A \equiv B \wedge C. \qquad \text{(а)}$$

Пусть задана некоторая формула A. Какой надо взять формулу B, чтобы для нее нашлась формула C, удовлетворяющая равенству (а)? Необходимое и достаточное условие состоит в следующем: формула B должна быть имплицентой для формулы A, то есть $A \supset B \equiv 1$. Иными словами, предикат B можно выбрать любым, но при условии, что он сохраняет все единичные значения предиката A. Это же самое можно сказать и о выборе предиката C: его можно выбрать любым, но с тем условием, чтобы он был имплицентой предиката A, то есть чтобы выполнялось условие $A \supset C \equiv 1$.

Если же к заданной формуле A подходящая формула B уже подобрана, то класс допустимых предикатов C, удовлетворяющих условию (а), сужается. Теперь, наряду с заранее сформулированными условиями, при выборе предиката C должно соблюдаться дополнительное условие $C \supset A \vee \overline{B} \equiv 1$ Действительно, согласно (а) имеем $A \sim BC \equiv 1$, что после тождественных преобразований дает: $(A \supset B)(A \supset C)(C \supset A \vee \overline{B}) \equiv 1$. Таким образом, предикат C должен выбираться с таким расчетом, чтобы он был импликантой предиката $A \vee \overline{B}$ и имплицентой предиката A. Иными словами, предикат C можно выбрать любым, но при условии, что он сохраняет все единичные значения предиката A и все нулевые значения предиката $A \vee \overline{B}$.

Проиллюстрируем сказанное примером. Требуется представить формулу

$$A \equiv y^1 z^0 t^1 \vee x^0 y^1 t^1 \vee x^0 y^1 z^0 \vee x^0 z^0 t^1$$

в виде конъюнкции возможно более простых формул B и C. Переменные x, y, z, t заданы в области {0, 1}. Значения предиката A указаны в таблице 2, составленной в форме диаграммы Вейча [2]. При формировании предиката B мы должны сохранить все единицы предиката A, часть же его нулей можно заместить единицами с таким расчетом, чтобы достичь определенного упрощения формулы для предиката B.

Таблица 2

xy \ zt	00	01	11	10	
00	0	1	0	0	
01	1	1	1	0	
11	0	1	0	0	A
10	0	0	0	0	

Таблица 3

xy \ zt	00	01	11	10	
00	1	1	0	0	
01	1	1	1	0	
11	0	1	1	0	B
10	0	0	0	0	

Наибольшего упрощения формулы для предиката B можно достичь, замещая все нули единицами, но тогда формула B превратится в истину, а формула C совпадет с формулой A, и в результате эффективная декомпозиция формулы A не будет достигнута. В табл. 3 даны значения предиката B, полученного в результате более умеренного замещения нулей единицами. Очень сильно упрощать формулу для предиката B не следует, так как это может привести к неоправданному усложнению формулы C. По табл. 3 находим минимальную ДНФ: $B \equiv y^1 t^1 \vee x^0 z^0$. В табл. 4 переносим все единицы предиката A (то есть все единицы табл. 2) и все нули предиката $A \vee \overline{B}$.

Таблица 4

xy \ zt	00	01	11	10
00	0	1		
01	1	1	1	
11		1	0	
10				

Нули надо ставить только в тех ячейках, в которых в табл. 3 стоят единицы, «лишние» по сравнению с табл. 2. В ячейках табл. 4, оставшихся незаполненными, можно проставить нули и единицы произвольным образом, руководствуясь соображениями минимизации формулы C. Предикат C

задаем табл. 5, ей соответствует следующая минимальная ДНФ:

$$C \equiv x^0 t^1 \vee y^1 z^0.$$

Таблица 5

xy \ zt	00	01	11	10
00		1	1	0
01	1	1	1	0
11	1	1		0
10	0	0	0	0

C

Таким образом:

$y^1 z^0 t^1 \vee x^0 y^1 t^1 \vee x^0 y^1 z^0 \vee x^0 z^0 t^1 \equiv (y^1 t^1 \vee x^0 z^0) \wedge (x^0 t^1 \vee y^1 z^0)$.

Мы совершили декомпозицию уравнения

$$y^1 z^0 t^1 \vee x^0 y^1 t^1 \vee x^0 y^1 z^0 \vee x^0 z^0 t^1 = 1 \qquad \text{(б)}$$

на систему более простых уравнений:

$$\begin{cases} y^1 t^1 \vee x^0 z^0 = 1, \\ x^0 t^1 \vee y^1 z^0 = 1. \end{cases} \qquad \text{(в)}$$

Заметим, что в данном примере декомпозиция уравнения привела не только к уменьшению длины каждого уравнения, но и к упрощению полной его записи, поскольку общее число узнаваний буквы в системе уравнений (в) меньше, чем в исходном уравнении (8 против 12).

Если бы мы в качестве первого сомножителя выбрали более простую формулу, $B' \equiv y^1 \vee t^1$, что соответствует табл. 6, то в роли второго сомножителя вынуждены были бы взять более сложное выражение (см. табл. 7 и 8):

$$C' \equiv x^0 z^0 \vee y^1 z^0 t^1 \vee x^0 y^1 t^1.$$

Такое построение приводит к системе уравнений

$$\left.\begin{array}{c} y^1 \vee t^1 = 1, \\ x^0 z^0 \vee y^1 z^0 t^1 \vee x^0 y^1 t^1 = 1, \end{array}\right\} \qquad \text{(г)}$$

более сложной, чем система (в) (10 узнаваний буквы против 8).

Таблица 6

xy \ zt	00	01	11	10
00	0	1	1	0
01	1	1	1	1
11	1	1	1	1
10	0	1	1	0

B'

Таблица 7

xy \ zt	00	01	11	10
00		1	0	
01	1	1	1	0
11	0	1	0	0
10		0	0	

Наиболее сложное уравнение в системе (в) имеет 4 узнавания буквы, в системе (г) – 8 узнаваний буквы. Таким образом, вариант декомпозиции (в) по обоим показателям удачнее варианта (г). Результат декомпозиции, вообще говоря, оказывается различным в зависимости от того, будем ли мы стремится к минимизации наиболее сложного уравнения системы или же к минимизации суммарной сложности уравнений системы. Общий метод решения этих задач здесь не приводится.

Таблица 8

xy \ zt	00	01	11	10
00	1	1	0	
01	1	1	1	0
11	0	1	0	0
10	0	0	0	0

C'

Задачу декомпозиции уравнения можно сформулировать и по-другому. Требуется заменить уравнение $A=1$ равносильной ему системой уравнений $\{A_1=1, A_2=1, ..., A_n=1\}$, причем число n уравнений нас не лимитирует и может быть принято достаточно большим. Однако важно, чтобы каждое уравнение $A_i=1$ ($1 \le i \le n$) было как можно более простым. Похожую задачу решает школьный учитель, излагающий первоклассникам свою мысль в виде последовательности достаточно простых высказываний. Метод решения этой задачи продемонстрируем на примере. Задано уравнение (б). Предикату, стоящему в левой части этого уравнения, соответствует табл. 2. Единицы таблицы охватываем какой-нибудь системой прямоугольников максимального размера, соответствующих наиболее коротким простым импликантам (табл. 9).

В результате получаем предикат $A_1 \equiv y^1 \vee t^1$. Крестиками в таблице отмечены «лишние» ячейки, попавшие внутрь прямоугольников. Содержательно эти ячейки будем интерпретировать как признак неполноты выражения мысли A высказыванием

A_1. Чем больше крестиков содержится в таблице, тем менее полно выражена мысль. Далее формируем предикаты $A_2 \equiv y^1 \vee z^0$, $A_3 \equiv x^0 \vee y^1$, $A_4 \equiv z^0 \vee t^1$, $A_5 \equiv x^0 \vee t^1$, $A_6 \equiv x^0 \vee z^0$ с таким расчетом, чтобы, во-первых, каждый из них был возможно более простым и, во-вторых, чтобы с введением очередного предиката сокращалось число крестиков (таблицы 10÷13).

Таблица 9

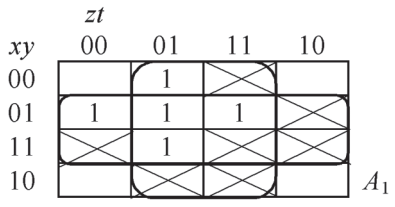


Таблица 10

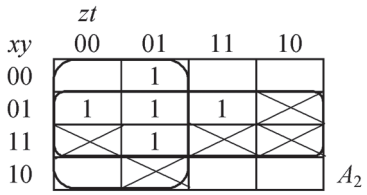


Таблица 11

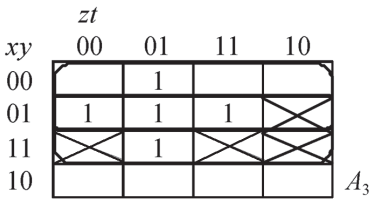
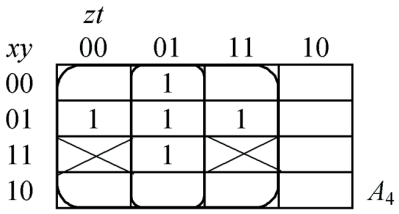


Таблица 12



После введения предиката A_6 все крестики исчезают (табл. 14). Это значит, что мы достигли полного выражения мысли A системой высказываний $A_1 \div A_6$. Таким образом, мы совершили декомпозицию уравнения (б) на систему простых уравнений

$$y^1 \vee t^1 = 1,\ y^1 \vee z^0 = 1,\ x^0 \vee y^1 = 1,\ z^0 \vee t^1 = 1,\ x^0 \vee t^1 = 1,\ x^0 \vee z^0 = 1.$$

Заметим, что в результате декомпозиции суммарное число узнаваний буквы в системе уравнений осталось таким же, как и в исходном уравнении (12 узнаваний буквы).

Таблица 13

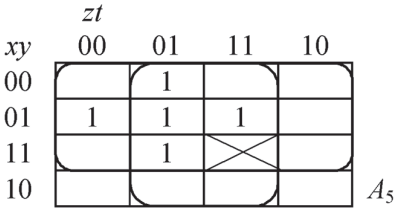
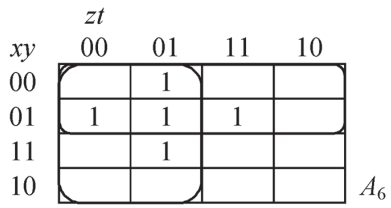


Таблица 14



Задачи только что рассмотренного типа можно решать также и чисто аналитически без привлечения диаграмм Вейча. Рассмотрим пример. Дано уравнение:

$$x_1^c x_2^a x_3^a \vee x_1^c x_2^a x_3^b \vee x_1^b x_3^b \vee x_1^c x_2^c x_3^b \vee x_1^b x_2^b x_3^c = 1.$$

Переменные x_1, x_2, x_3 – троичные со значениями a, b, c. Требуется заменить это уравнение равносильной ему системой возможно более коротких уравнений. Ищем первое уравнение системы, отправляясь от предиката:

$$x_1^c x_2^a x_3^a \vee x_1^c x_2^a x_3^b \vee x_1^b x_2^b \vee x_1^c x_2^c x_3^b \vee x_1^b x_2^b x_3^c \vee$$
$$\vee x_1^a x_2^a x_3^a \vee x_1^a x_2^b x_3^a \vee x_1^a x_2^c x_3^a \vee x_1^a x_2^a x_3^b \vee$$
$$\vee x_1^a x_2^b x_3^b \vee x_1^a x_2^c x_3^b \vee x_1^a x_2^a x_3^c \vee x_1^a x_2^b x_3^c \vee$$
$$\vee x_1^a x_2^c x_3^c \vee x_1^b x_2^a x_3^a \vee x_1^b x_2^b x_3^a \vee x_1^b x_2^c x_3^a \vee$$
$$x_1^b x_2^a x_3^c \vee x_1^b x_2^c x_3^c \vee x_1^c x_2^b x_3^a \vee x_1^c x_2^c x_3^a \vee$$
$$\vee x_1^c x_2^b x_3^b \vee x_1^c x_2^a x_3^c \vee x_1^c x_2^b x_3^c \vee x_1^c x_2^c x_3^c.$$

В указанном предикате сохранена вся левая часть исходного уравнения. Кроме того, в него введены в роли дополнительных членов конституэнты единицы для всех наборов (x_1, x_2, x_3), не являющихся решениями исходного уравнения. Эти конституэнты единицы заключены в квадратные скобки. Производим такую минимизацию полученной формулы, при которой разрешается использовать любые дополнительные члены, но не все одновременно. Все дизъюнктивные члены, не охваченные квадратными скобками, в процессе минимизации должны быть использованы обязательно.

После минимизации получаем левую часть первого уравнения $x_1^b \vee x_1^c$. В процессе минимизации были использованы следующие дополнительные конституэнты единицы:

$$x_1^b x_2^a x_3^a,\ x_1^b x_2^b x_3^a,\ x_1^b x_2^c x_3^a,\ x_1^b x_2^a x_3^c,\ x_1^b x_2^c x_3^c,$$
$$x_1^c x_2^b x_3^a,$$
$$x_1^c x_2^c x_3^a,\ x_1^c x_2^b x_3^b,\ x_1^c x_2^a x_3^c,\ x_1^c x_2^b x_3^c,\ x_1^c x_2^c x_3^c.$$

Далее, отправляясь от того же предиката, ищем второе уравнение системы. Теперь в процессе минимизации не должны использоваться все конституэнты единицы из только что приведенного перечня. Все те конституэнты единицы, которые не приведены в этом перечне и охвачены квадратными скобками, могут использоваться или не использоваться по нашему усмотрению. Получаем левую часть второго уравнения $x_1^b \vee x_2^a \vee x_2^c$. При минимизации не использовались из нашего перечня сле-

дующие конституэнты единицы: $x_1^c x_2^b x_3^a$, $x_1^c x_2^b x_3^b$, $x_1^c x_2^b x_3^c$. Повторяем процесс минимизации для еще более сокращенного перечня конституэнт единицы:

$$x_1^b x_2^a x_3^a,\ x_1^b x_2^b x_3^a,\ x_1^b x_2^c x_3^a,\ x_1^b x_2^a x_3^c,$$
$$x_1^b x_2^c x_3^c,\ x_1^c x_2^c x_3^a,\ x_1^c x_2^a x_3^c,\ x_1^c x_2^c x_3^c.$$

Получаем левую часть третьего уравнения $x_1^c x_2^a \vee x_3^b \vee x_3^c$. Перечень оставшихся конституэнт единицы становится еще меньшим:

$$x_1^b x_2^a x_3^c,\ x_1^b x_2^c x_3^c,\ x_1^c x_2^a x_3^c,\ x_1^c x_2^c x_3^c.$$

Наконец, получаем левую часть четвертого уравнения $x_1^b x_2^b \vee x_3^a \vee x_3^b$. Перечень конституэнт единицы превращается в пустое множество, что служит признаком окончания процесса построения системы уравнений. Итак, мы заменили исходное уравнение

$$x_1^c x_2^a x_3^a \vee x_1^c x_2^a x_3^b \vee x_1^b x_2^b \vee x_1^c x_2^c x_3^b \vee x_1^b x_2^b x_3^c = 1 \quad (д)$$

системой равносильных ему уравнений:

$$x_1^b \vee x_1^c = 1,\ x_1^b \vee x_2^a \vee x_2^c = 1,\ x_1^c x_2^a \vee x_3^b \vee x_3^c = 1,$$
$$x_1^b x_2^b \vee x_3^a \vee x_3^b = 1.$$

Суммарное число букв в системе осталось почти тем же, что и в исходном уравнении (13 узнаваний буквы против 14), зато максимальная сложность уравнений резко уменьшилась (с 14 до 4 узнаваний буквы).

Рассмотрим один специальный, но важный метод декомпозиции уравнений алгебры конечных предикатов, основанный на применении теоремы о конъюнктивном разложении. Пусть требуется произвести декомпозицию произвольно выбранного уравнения $f(x_1, x_2,..., x_n)=1$, где переменная x_1 определена на множестве букв $\{a_1, a_2,..., x_k\}$. Тогда согласно первому следствию теоремы о конъюнктивном разложении имеем:

$$f(x_1, x_2,..., x_n) \equiv (\overline{x_1^{a_1}} \vee f(a_1, x_2,..., x_n))...$$
$$(\overline{x_1^{a_k}} \vee f(a_k, x_2,..., x_n)).$$

Таким образом, исходное уравнение заменяется системой, состоящей из k уравнений

$$\begin{aligned} &\overline{x_1^{a_1}} \vee f(a_1, x_2, \dots, x_n) = 1, \\ &\dots\dots\dots\dots\dots\dots \\ &\overline{x_1^{a_k}} \vee f(a_k, x_2, \dots, x_n) = 1. \end{aligned} \quad (е)$$

При желании процесс декомпозиции можно продолжить, раскладывая левую часть каждого из уравнений (е) по переменной x_2 и т.д.

Для иллюстрации метода приведем декомпозицию уравнения (д), рассмотренного в предыдущем примере. Производим разложение левой части уравнения (д) по переменной x_1. В результате получаем следующую систему уравнений:

$$\overline{x_1^a} = 1,\ \overline{x_1^b} \vee x_3^b \vee x_2^b x_3^c = 1,\ \overline{x_1^c} \vee x_2^a x_3^a \vee x_2^a x_3^b \vee x_2^c x_3^b = 1.$$

Согласно закону отрицания:

$$\overline{x_1^a} \equiv x_1^b \vee x_1^c,\quad \overline{x_1^b} \equiv x_1^a \vee x_1^c,\quad \overline{x_1^c} \equiv x_1^a \vee x_1^b.$$

Окончательно имеем:

$$x_1^b \vee x_1^c = 1,\ x_1^a \vee x_1^c \vee x_3^b \vee x_2^b x_3^c = 1,$$
$$x_1^a \vee x_1^b \vee x_2^a x_3^a \vee x_2^a x_3^b \vee x_2^c x_3^b = 1.$$

Суммарное число предикатов узнавания буквы в полученной системе несколько увеличилось по сравнению с тем, которое было в исходном уравнении (15 узнаваний буквы против 14), зато сложность каждого уравнения уменьшилась (соответственно 2, 5 и 8 узнаваний буквы против 14). Продолжив разложение второго и третьего уравнений по переменным x_2 и x_3, можно добиться еще большего упрощения отдельных уравнений, однако их число при этом увеличится.

Рассмотренный выше процесс замены уравнения равносильной ему системой более простых уравнений назовем *эквивалентной декомпозицией*. Наряду с нею возможна еще и *неэквивалентная декомпозиция уравнений*. Идею неэквивалентной декомпозиции поясним на примере. Дано уравнение:

$$x_1^b x_2^c x_3^b \vee (x_1^b x_2^c x_3^b \vee x_1^c)(x_2^c x_3^b \vee x_2^a) = 1.$$

Вводим обозначения для предикатов $t_1 = x_2^c x_3^b$ и $t_2 = x_1^b t_1$, входящих несколько раз в исходное уравнение. После подстановки имеем $t_2 \vee (t_2 \vee x_1^c)(t_1 \vee x_2^a) = 1$. Таким образом, мы заменили исходное уравнение системой, состоящей из трех более простых уравнений:

$$t_1 = x_2^c x_3^b,\ t_2 = x_1^b t_1,\ t_2 \vee (t_2 \vee x_1^c)(t_1 \vee x_2^a) = 1.$$

Наиболее сложное из этих уравнений имеет 2 узнавания буквы и 3 вхождения логических переменных против 10 узнаваний буквы в исходном уравнении. Полученная система уравнений, строго говоря, неэквивалентна исходному уравнению, так как в ней содержатся дополнительные логические переменные t_1 и t_2. Однако если эти переменные рассматривать исключительно как промежуточные, воздерживаясь от их использования в роли независимых переменных, то мы обнаружим, что система уравнений связывает переменные x_1, x_2, x_3 тем же самым отношением, что и исходное уравнение.

Рассмотрим один специальный, но важный способ введения промежуточных логических переменных в уравнение алгебры конечных предикатов. Пусть задано произвольное уравнение

$$f(x_1, x_2,..., x_n) = 1, \quad (ж)$$

где переменная x_1 определена на множестве букв $\{a_1, a_2,..., x_k\}$. Тогда согласно первому следствию теоремы о дизъюнктивном разложении имеем:

$$f(x_1, x_2, ..., x_n) \equiv \overline{x_1^{a_1}}\, f(a_1, x_2, ..., x_n) \vee ... \vee$$
$$\vee\, \overline{x_1^{a_k}}\, f(a_k, x_2, ..., x_n)).$$

Обозначим

$$\begin{gathered} f(a_1, x_2, ..., x_n) = t_1, \\ ..... \\ f(a_k, x_2, ..., x_n) = t_k. \end{gathered} \qquad (з)$$

После замены уравнение (ж) запишется в виде:

$$t_1 x_1^{a_1} \vee t_2 x_1^{a_2} \vee ... \vee t_k x_1^{a_k} = 1. \qquad (и)$$

Таким образом, мы заменили уравнение (ж) системой уравнений (з) и (и). Каждое из уравнений (з) в случае надобности также можно заменить некоторой системой уравнений, производя разложение по переменной x_2, и т. д.

Рассмотрим пример декомпозиции уравнения только что описанным методом. Дано уравнение:

$$(x_1^a \vee x_1^b) x_2^b x_3^c \vee x_1^b x_2^a x_3^b \vee x_1^c = 1. \qquad (к)$$

Производим разложение по переменной x_1. Записываем уравнение (и):

$$t_1 x_1^a \vee t_2 x_1^b \vee t_3 x_1^c = 1.$$

Согласно (в):

$$t_1 = (a^a \vee a^b) x_2^b x_3^c \vee a^b x_2^a x_3^b \vee a^c \equiv x_2^b x_3^c,$$
$$t_2 = (b^a \vee b^b) x_2^b x_3^c \vee b^b x_2^a x_3^b \vee b^c \equiv x_2^b x_3^c \vee x_2^a x_3^b,$$
$$t_3 = (c^a \vee c^b) x_2^b x_3^c \vee c^b x_2^a x_3^b \vee c^c \equiv 1.$$

Таким образом, уравнение (к) мы заменили системой уравнений:

$$t_1 x_1^a \vee t_2 x_1^b \vee t_3 x_1^c = 1,\ t_1 = x_2^b x_3^c,\ t_2 = t_1 \vee x_2^a x_3^b. \qquad (л)$$

Исходное уравнение (к) содержит 8 узнаваний буквы, наиболее же сложное из уравнений (л) – только 3 узнавания буквы и 2 вхождения логических переменных. Каждое уравнение системы проще, чем исходное уравнение, однако вместе взятые уравнения системы (л) сложнее (7 узнаваний буквы и 5 вхождений логических переменных против 8 узнаваний буквы в исходном уравнении). При определении сложности уравнений мы условно приравняли одно вхождение логической переменной к одному узнаванию буквы.

Выводы

Описанными выше методами широко пользуются люди в своей интеллектуальной деятельности. Когда какое-то понятие (то есть формулу) нам приходится использовать многократно, то мы вводим специальный термин (то есть обозначение формулы), его называющий, после чего для краткости в высказываниях употребляем уже не сами понятия, а только лишь соответствующие им термины. Очевидно, что метод неэквивалентной декомпозиции будет эффективным не для любых уравнений, а лишь для тех из них, у которых имеются черты иерархической структуры. Можно предположить, что уравнения именно такого типа будут наиболее важны для теории интеллекта. Об этом свидетельствует тот факт, что люди в своей интеллектуальной деятельности чрезвычайно широко пользуются терминологией, а это значит, что наши мысли имеют иерархическую структуру, построены по блочному принципу. Это свойство интеллекта человека назовем *иерархичностью интеллекта*. Весьма вероятно, что и окружающий нас физический мир также имеет иерархическую структуру, иначе было бы трудно понять, каким образом нам удается успешно его познавать. Мы ожидаем, что прием неэквивалентной декомпозиции уравнений найдет применение при решении многих задач теории интеллекта.

Список литературы: 1. *Бондаренко, М.Ф.* Об алгебре конечных предикатов [Текст] / М.Ф. Бондаренко, Ю.П. Шабанов-Кушнаренко, С.Ю. Шабанов-Кушнаренко // Бионика интеллекта. – 2011. – № 3. – С. 3-13. **2.** Справочник по цифровой вычислительной технике. – К.: Изд-во «Техніка», 1979. – 511 с.

Поступила в редколлегию 12.09.2011

УДК 519.7

Рівняння теорії інтелекту / Бондаренко М.Ф., Шабанов-Кушнаренко Ю.П. // Біоніка інтелекту: наук.-техн. журнал. – 2011. – № 3 (77). – С. 30-45.

Розвивається спеціалізований математичний апарат для ефективного моделювання роботи механізмів інтелекту людини – алгебри предикатів і предикативних операцій. Запропоновані способи запису предикативних рівнянь, а також методи їх аналітичного розв'язання.

Бібліогр.: 2 найм.

UDC 519.7

The intellect theory equations / M.F. Bondarenko, Yu.P. Shabanov-Kushnarenko // Bionics of Intelligense: Sci. Mag. – 2011. – № 3 (77). –P. 30-45.

The specialized mathematical apparatus develops for the effective design of the human intellect mechanisms work - algebra of predicates and predicate operations. The methods of the predicate equalizations record, and also methods of their analytical decision, are offered.

Ref.: 2 items.

УДК 65.012.23 + 519.766.2

Н.В.Голян[1], Ю.П. Шабанов-Кушнаренко[2]
[1, 2] ХНУРЭ, Харьков, Украина, veragolyan@yandex.ru

ПРЕДИКАТНЫЕ МОДЕЛИ НЕЯВНЫХ СВЯЗЕЙ МЕЖДУ ПРОЦЕДУРАМИ БИЗНЕС-ПРОЦЕССА

Произведен анализ извлеченной информации из журналов регистрации событий бизнес-процесса с тем, чтобы формализовать реальное поведение БП. Такой анализ данных особенно важен в тех случаях, когда регистрируется происходящая последовательность событий, т.е. исполнители имеют возможность принимать решение о порядке дальнейшего прохождения процесса.

БИЗНЕС-ПРОЦЕСС, ПРОЦЕДУРА, ЛОГИЧЕСКАЯ СЕТЬ, ИНТЕЛЛЕКТУАЛЬНЫЙ АНАЛИЗ

Введениие

Широкое внедрение процессно-ориентированных систем на сегодняшний день связано с тем, что они позволяют формализовать знания о протекании бизнес-процессов на предприятиях и в организациях. В то же время, построение формальных моделей бизнес-процессов требует значительных временных и материальных затрат, а также оно подвержено влиянию человеческого фактора, поскольку часто проводится во взаимодействии с экспертами и исполнителями, цели которых могут не совпадать с целями моделируемых бизнес-процессов и общими целями предприятия. Данное противоречие может приводить к расхождению между реальным бизнес-процессом и его разработанной моделью. Это может приводить к построению неадекватных моделей бизнес-процессов.

Один из важных подходов к решению данной проблемы реализуется на основе методологии интеллектуального анализа бизнес-процессов. Интеллектуальный анализ направлен на получение моделей реально выполняющихся бизнес-процессов на основе исследования журналов регистрации событий таких процессов (файлов - логов).

Такой анализ данных особенно важен в тех случаях, когда регистрируется происходящая последовательность событий, однако процесс частично либо полностью не формализован, т.е. исполнители имеют возможность принимать решения о порядке дальнейшего прохождения процесса, исходя из имеющейся у них локальной информации и с учетом своих знаний о правилах работы бизнес-процесса. Фактически, в таких процессах реализуются неявные связи между процедурами БП. Указанные связи основаны на знаниях, не входящих в описание процесса, могут приводить к отклонениям от его документированного поведения и при этом практически не распознаются на основе существующих подходов в области интеллектуального анализа бизнес-процессов.

В настоящее время разработаны алгоритмы построения моделей бизнес-процессов на основе анализа журналов регистрации событий и [1-4]. В то же время, разработанные подходы не позволяют до конца решить проблему выявления неявных связей и, с их помощью, конструкций неявного выбора в структуре бизнес-процессов.

1. Постановка задачи

Неявные зависимости в бизнес-процессах отражают непрямые причинно-следственные связи между процедурами и обладают свойствами связности, достижимости и не обладают свойством последовательности. В случае неявных зависимостей между рассматриваемыми процедурами бизнес-процесса существует цепочка других процедур (непрямая связь), что и затрудняет выявление таких фрагментов.

Все вышеизложенное определяет важность формализации неявных связей между процедурами.

Задача авторов статьи состоит в получении формальных моделей неявных связей между процедурами бизнес-процесса, которые бы обладали следующими особенностями:

– отражение параллельного и последовательного выполнения текущего фрагмента бизнес-процесса и остальных его подпроцессов;

– охват необходимого и достаточного набора свойств неявных связей, позволяющих выделить их на основе анализа последовательности процедур выполнившегося бизнес-процесса.

2. Разработка предикатных моделей неявных связей между процедурами

Разработка метода идентификации ситуаций неявного выбора требует формализации основных признаков таких ситуаций, а именно формализации неявных связей различного вида, что требует разработки моделей представления неявных связей между процедурами.

Данный подраздел посвящен разработке предикатных моделей представления неявных связей между процедурами бизнес-процесса на основе алгебро-логической модели обобщенной конструкции неявного выбора. Данная модель объединяет четыре схемы взаимодействия конструкции неявного выбора с другими фрагментами бизнес-процесса. Поэтому далее будут рассмотрены и формализованы в виде предикатных моделей 4 типа неявных связей между процедурами.

Реализация моделей неявных связей базируется на предикатной модели представления непрямой связи между процедурами в журнале регистрации событий бизнес-процесса.

Рассмотрим и доказательно формализуем неявные связи всех четырех типов.

Неявная связь типа 1: выходы внешнего подпроцесса – входы анализируемого фрагмента.

Данный тип связи основан на взаимодействиях вида: выход внешнего подпроцесса – входы конечной процедуры P_i и промежуточной цепочки процедур P_2, ..., P_n. (рис. 1). В соответствии с данной схемой взаимодействия сформулируем набор условий, определяющих связь данного типа:

1) Между начальной P_1 и конечной P_n процедурами исследуемого фрагмента отсутствует явная связь.

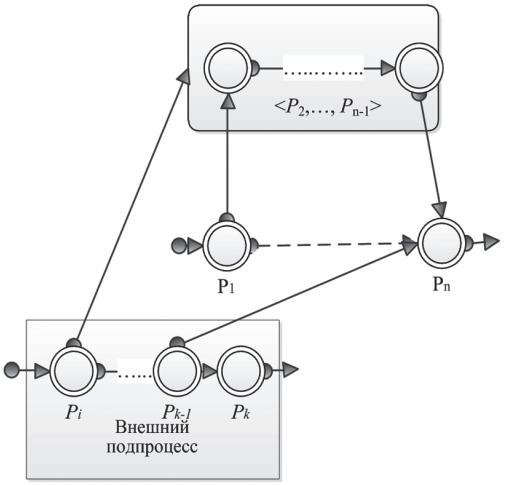


Рис. 1. Неявная связь типа 1: выходы внешнего подпроцесса – входы анализируемого фрагмента

2) У начальной процедуры фрагмента P_1 в полученной на основании анализа журнала операций модели имеется только один выход, который является входом для процедуры $P_j, j = \overline{2, n\text{-}1}$.

3) У процедуры P_j имеется второй вход, который является общим для конечной процедуры P_n.

4) Общий для процедур P_j и P_n вход является результатом работы процедуры P_k, которая относится к иной ситуации данного бизнес-процесса или иному подпроцессу. Отметим, что разделение по ситуациям или подпроцессам заложено в структуре журнала регистрации событий, поскольку в таком журнале обычно существует графа «Код ситуации».

5) Внешняя по отношению к анализируемому фрагменту процедура P_k (обобщая – весь внешний подпроцесс) не связана с начальной процедурой P_1 по входу-выходу.

6) Между процедурами P_1 и P_n имеется непрямая связь.

При выполнении рассмотренных шести условий между процедурами P_1 и P_n имеется неявная связь первого типа.

Подведя итог изложенному, можно сказать, что данная связь позволяет идентифицировать конструкцию неявного выбора в полученной в результате анализа журнала регистрации событий модели бизнес-процесса. Выявление такой конструкции происходит тогда, когда в анализируемом фрагменте модели имеется такая промежуточная последовательность из одной или более процедур, что вход данной последовательности одновременно со входом конечной процедуры фрагмента определяется выходом внешнего подпроцесса. Также конечная процедура фрагмента имеет второй вход, тогда по этому второму входу имеется неявная связь между начальной и конечной процедурами.

Неявная связь типа 2: докажем наличие неявной связи на основе упрощенной схемы взаимодействия конструкции неявного выбора с другими фрагментами бизнес-процесса на базе связи по входу в P_n. Упрощение заключается в том, что последовательность процедур $<P_2 \ldots P_n>$ заменяется одной процедурой P_n, а внешний подпроцесс заменяется отдельной процедурой P_2. Данное упрощение не влияет на суть доказательства, поскольку представление части процесса в виде последовательности процедур либо единой обобщенной процедуры, реализующей весь подпроцесс, зависит от степени детализации модели (рис. 2).

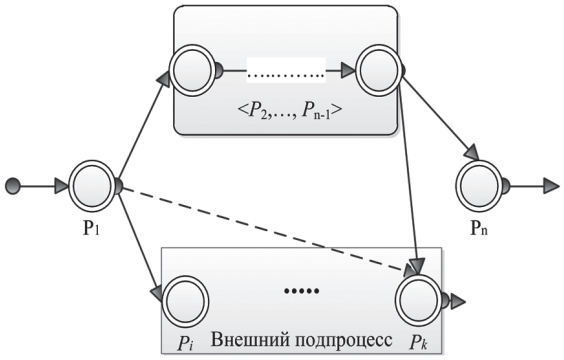


Рис. 2. Неявная связь типа 2: выходы анализируемого фрагмента – входы внешнего подпроцесса

Исходя из представленных на рис. 2 взаимодействий, сформулируем набор условий, определяющих неявную связь данного типа:

1) Между начальной P_1 и конечной P_n процедурами исследуемого фрагмента отсутствует явная связь.

2) У начальной процедуры фрагмента P_1 в полученной на основании анализа журнала операций модели имеется два выхода (или больше – в общем случае), которые являются входами:

– для начальной процедуры $P_j, j = \overline{2, n\text{-}1}$ внешнего подпроцесса;

– для процедуры P_2 текущего фрагмента бизнес-процесса;

3) Между процедурами P_1 и P_n имеется непрямая связь в смысле выражения.

4) Результаты выполнения процедуры $P_{n\text{-}1}$ используются как входные для процедур P_k и P_n.

5) Конечная процедура внешнего подпроцесса P_k не связана непрямой связью с начальной процедурой P_n. Точнее, такая связь в общем случае не гарантируется.

При выполнении рассмотренных пяти условий между процедурами P_1 и P_k имеется неявная связь второго типа.

Набор сформулированных условий позволяет идентифицировать конструкцию неявного выбора в полученной в результате анализа журнала регистрации событий модели бизнес-процесса в том случае, если в текущем фрагменте модели существует два (в общем случае более двух) варианта протекания процесса, которые завершаются процедурами P_k и P_n. Их выполнение зависит от того, по какой цепочке — $<P_2 \dots P_{n-1}>$ или по внешнему подпроцессу $<P_i \dots P_k>$ пойдет реализация бизнес-процесса после выполнения начальной процедуры. Тогда, если при реальном исполнении процесса, отраженном в журнале регистрации событий, выполнена процедура P_n, значит произошла реализация цепочки $<P_2 \dots P_{n-1}>$.

Неявная связь типа 3: внешний подпроцесс выполняется параллельно, влияя на последовательность $<P_2 \dots P_{n-1}>$ (рис. 3).

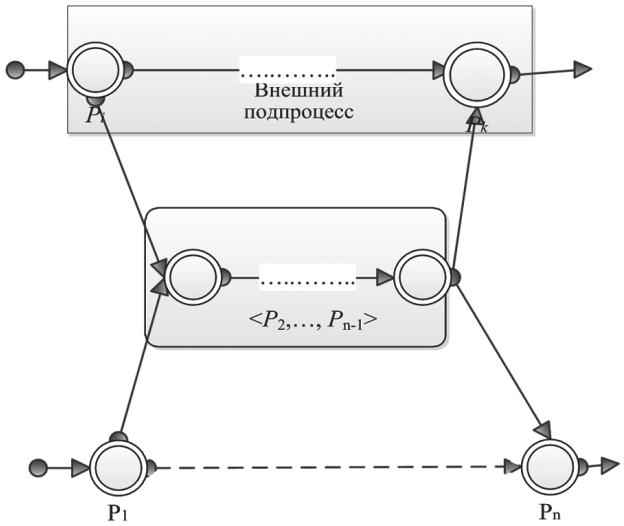


Рис. 3. Неявная связь типа 3: внешний подпроцесс выполняется параллельно

На основании анализа представленной схемы сформулируем набор условий, определяющих связи данного типа:

– выход начальной процедуры P_1 – вход в промежуточный фрагмент $<P_2 \dots P_n>$;

– выход начальной процедуры внешнего подпроцесса P_i – вход в промежуточный фрагмент $<P_2 \dots P_n>$;

– выход процедуры P_{n-1} – вход в конечную процедуру внешнего подпроцесса $P_k\$$

– выход последней процедуры промежуточной последовательности текущего фрагмента P_{n-1} вход в конечную процедуру внешнего подпроцесса P_k.

На основании анализа представленной схемы сформулируем набор условий, определяющих связи данного типа:

1) Между начальной P_1 и конечной P_n процедурами исследуемого фрагмента отсутствует явная связь.

2) Между начальной P_1 и конечной P_n процедурами исследуемого фрагмента существует непрямая связь .

3) У первой процедуры P_2 промежуточной последовательности $<P_2 \dots P_n>$ имеется два входа:

– из начальной процедуры P_1 текущего фрагмента бизнес-процесса внешнего подпроцесса;

– из начальной процедуры P_i внешнего подпроцесса.

4) Результаты выполнения процедуры P_{n-1} используются как входные для процедур P_k и P_{n-1} .

5) Конечная процедура P_k внешнего подпроцесса не связана непрямой связью с начальной процедурой P_1. Точнее, такая связь в общем случае не гарантируется.

При выполнении рассмотренных пяти условий между процедурами P_i и P_k имеется неявная связь третьего типа.

Приведенные выше условия позволяют идентифицировать конструкцию неявного выбора третьего типа в том случае, если параллельно текущему фрагменту бизнес-процесса выполняется внешний подпроцесс, что приводит к возникновению конструкции с двумя входными процедурами и двумя выходными, определяющими два варианта протекания процесса. Выполнение того или иного варианта зависит от того, по какой цепочке – $<P_2 \dots P_{n-1}>$ или по внешнему подпроцессу $<P_i \dots P_k>$ пойдет реализация бизнес-процесса после выполнения начальных процедур. Тогда, если в журнале регистрации событий отражено выполнение процедуры, то произошло выполнение цепочки $<P_2 \dots P_{n-1}>$ и, следовательно, между процедурами имеется неявная связь.

Неявная связь типа 4: внешний подпроцесс выполняется параллельно и в зависимости от последовательности $<P_2 \dots P_{n-1}>$.

Данный тип связи является детализацией взаимодействий, представленных на рис. 2 и основан на параллельном выполнении рассматриваемого фрагмента и внешнего подпроцесса, причем запуск внешнего подпроцесса определяется текущим фрагментом, а его завершение влияет на выполнение текущего фрагмента (рис. 4):

– выход процедуры P_2 – вход в начальную процедуру внешнего подпроцесса P_i;

– выход конечной процедуры внешнего подпроцесса P_k – вход в последнюю процедуру промежуточной последовательности текущего фрагмента P_{n-1}.

На основании анализа изображенной выше схемы сформулируем набор условий, позволяющих формализовать связи данного типа:

1) Между начальной P_1 и конечной P_n процедурами исследуемого фрагмента отсутствует явная связь.

2) Между начальной P_1 и конечной P_n процедурами исследуемого фрагмента существует непрямая связь через последовательность $<P_2 \dots P_{n-1}>$.

3) У первой процедуры P_2 промежуточной последовательности имеется два выхода:

– в последующую процедуру текущего фрагмента бизнес-процесса;

– в начальную процедуру P_i внешнего подпроцесса.

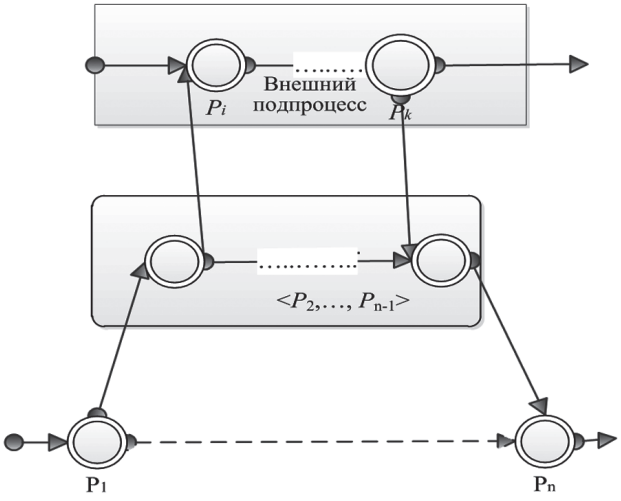


Рис. 4. Неявная связь типа 4: внешний подпроцесс выполняется параллельно

4) Результаты выполнения процедуры внешнего подпроцесса используются как входные для процедуры P_k .

5) Конечная процедура внешнего подпроцесса не связана непрямой связью с начальной процедурой. Точнее, такая связь в общем случае не гарантируется.

При выполнении рассмотренных пяти условий между процедурами P_i и P_k имеется неявная связь четвертого типа.

Полученные в данном подразделе предикатные модели неявных связей между процедурами отражают различные варианты параллельного и последовательного выполнения текущего фрагмента бизнес-процесса, отражающего текущую ситуацию, с другими подпроцессами. Указанные модели позволяют формально представить скрытые знания о взаимосвязях между процедурами и, таким образом, повысить адекватность модели, получаемой в результате анализа журналов регистрации событий бизнес-процесса.

Выводы

Неявные зависимости в бизнес-процессах отражают непрямые причинно-следственные связи между процедурами и обладают свойствами связности, достижимости и не обладают свойством последовательности. В случае неявных зависимостей между рассматриваемыми процедурами бизнес-процесса существует цепочка других процедур (непрямая связь), что и затрудняет выявление таких фрагментов.

На примерах работы построенных логических сетей можно видеть, что в большей части тактов вычисляется только один предикат, что говорит о том, что эти логические сети работают практически в последовательном режиме и не имеют преимущества над моделями этих же процессов в виде многоместных предикатов. Однако если учесть, что в рассмотренных примерах охватывались только небольшие ключевые фрагменты реальных бизнес-процессов, а полные модели часто содержат много процедур, которые можно выполнять одновременно, то тогда преимущество моделей в виде систем бинарных уравнений станет очевидным.

Практический аспект полученных результатов заключается в следующем. Выполнение логической сети, реализующей конструкции неявного выбора, обеспечивает возможность для получения журнала регистрации событий с отражением неявных взаимосвязей между процедурами, что создает условия для разработки методов выявления конструкций неявного выбора.

Список литературы: 1. *R. Agrawal, D. Gunopulos, and F. Leymann.* Mining Process Models from Workflow Logs [Текст] / In Sixth International Conference on Extending Database Technology. – 1998. – 469-483 pages. **2.** *W.M.P. van der Aalst, B.F. van Dongen, J. Herbst, L. Maruster, G. Schimm, and A.J.M.M. Weijters* [Текст] / Workflow Mining: A Survey of Issues and Approaches. Data and Knowledge Engineering, 47(2):237–2003. – 267 pages. **3.** *A.K.A. de Medeiros, W.M.P. van der Aalst, and A.J.M.M. Weijters.* Workflow Mining: Current Status and Future Directions. In R. Meersman, Z. Tari, and D.C. Schmidt, editors [Текст] // On The Move to Meaningful Internet Systems 2003: CoopIS, DOA, and ODBASE. Volume 2888 of Lecture Notes in Computer Science. Pages 389 -406. Springer-Verlag, Berlin, 2003. **4.** *W.M.P. van der Aalst, A.J.M.M. Weijters, and L. Maruster.* Workflow Mining: Discovering Process Models from Event Logs. [Текст]/ IEEE Transactions on Knowledge and Data Engineering. – 2004. – 16(9):1128-1142 pages.

Поступила в редколлегию 20.09.2011

УДК 65.012.23 + 519.766.2

Предикативні моделі неявних зв'язків між процедурами бізнес-процесу / Н.В. Голян, Ю.П. Шабанов-Кушнаренко // Біоніка інтелекту: наук.-техн. журнал. – 2011. – № 3 (77). – С. 46-49.

Розроблено ефективні методи інтелектуального аналізу бізнес-процесів, зокрема, методи виявлення фрагментів таких процесів. Також проведено аналіз витягнутої інформації з журналів реєстрації подій бізнес-процесу з тим, щоб формалізувати реальну поведінку БП. Такий аналіз даних особливо важливий у тих випадках, коли реєструється відбувається послідовність подій, тобто виконавці мають можливість приймати рішення про порядок подальшого проходження процесу

Іл.: 4. Бібліогр.: 4 найм.

УДК 65.012.23 + 519.766.2

Predicate models of non-obvious cjnnections between procedures of business prosses / N.V. Golyan, Ю.П. Шабанов-Кушнаренко// Bionics of Intelligense: Sci. Mag. – 2011. – № 3 (77). –P. 46-49.

The effective methods of intellectual analysis of business processes are worked out, in particular, methods of exposure of fragments of such processes. The analysis of prolate information is also conducted from the magazines of registration of events to the business process with that, to formalize the real behavior of БП. Such analysis of data is especially important in those cases, when registered there is a sequence of events, id est performers have the opportunity to make decision about the order of the further passing of process.

Fig.: 4. Ref.: 4 items.

УДК 519.7

Н. Е. Русакова
ХНУРЭ, г. Харьков, Украина

О МЕТОДЕ РАССЛОЕНИЯ КОНЕЧНОГО ПРЕДИКАТА

Анализ уже имеющихся достижений в области моделирования мозгоподобных структур показывает, что многие актуальные задачи в этой области еще ждут своего решения. В связи с этим в статье ставится задача разработки метода преобразования математической модели произвольной конечной мозгоподобной структуры в ее техническую эффективно действующую модель, называемую реляционной сетью.

РАССЛОЕНИЕ ПРЕДИКАТА, КЛАССИФИКАТОРЫ, ПЕРЕКЛЮЧАТЕЛЬНАЯ ЦЕПЬ, РЕЛЯЦИОННАЯ СЕТЬ, МОЗГОПОДОБНАЯ СТРУКТУРА

Введение

Как известно, отношение – это универсальное средство формального описания структуры любых объектов, их свойств, связей между ними, действий над нами, а также любых информационных процессов. В формульном виде отношения записываются с помощью предикатов [1]. Любой предикат, описывающий конечный процесс в вычислительной технике, можно записать в виде таблицы. В статье решается вопрос перехода от таблицы предиката к логической схеме, электронная реализация которой будет решать уравнения вида $P(x_1, x_2,..., x_m)=1$.

Ниже представлен алгоритм расслоения предиката $P(x_1, x_2,..., x_m)$, заданного на $A_1 \times A_2 \times ... \times A_m$. При этом размерность m области $A_1, A_2,..., A_m$ задания аргументов $x_1, x_2,..., x_m$ предиката P и сам предикат могут быть выбраны произвольно. Расслоение предиката ценно тем, что приводит к построению реляционной сети любой математической модели и к обоснованию ее универсальности и эффективности. Кроме того, расслоение предиката служит фундаментом для обобщения метода нулевого прибора [2], который является основным средством моделирования психофизических процессов. Такое обобщение приводит к существенному расширению сферы применения метода нулевого прибора, превращая его из метода моделирования процессов классификации объектов в универсальное средство формального описания произвольных психофизических процессов.

Описание алгоритма расслоения

Приведенный алгоритм сопровождается универсальным примером. При решении этого примера должны присутствовать все детали, которые могут возникнуть в более сложном примере, а также он должен быть наиболее простым.

Предикат $P(x_1, x_2, x_3)=t$ задан табл. 1. Имеем три предметных переменных x_1, x_2, x_3, каждая задана на своей области A_1, A_2, A_3. Если записать уравнение $P(x_1, x_2,..., x_m)=1$, то это уравнение задает отношение, соответствующее предикату P.

Таблица 1

Таблица задания предиката P

x_1 \ x_2x_3	00	01	10	11	20	21	
0	1	1	1	1	1	0	
1	1	1	1	1	1	0	t
2	1	0	1	0	0	0	
3	1	0	1	0	0	0	

Определяем области изменения A_1, A_2, A_3 исходных переменных x_1, x_2, x_3 предиката P:

$$A_1(x_1)=x_1^0 \vee x_1^1 \vee x_1^2 \vee x_1^3;$$

$$A_2(x_2)=x_2^0 \vee x_2^1 \vee x_2^2;$$

$$A_3(x_3)=x_3^0 \vee x_3^1.$$

Следовательно:

$$A_1=\{0, 1, 2, 3\},\ A_2=\{0, 1, 2\},\ A_3=\{0, 1\}.$$

Совершенная дизъюнктивная нормальная форма исходного предиката P равна:

$$P(x_1, x_2, x_3)=x_1^0x_2^0x_3^0 \vee x_1^0x_2^0x_3^1 \vee x_1^0x_2^1x_3^0 \vee x_1^0x_2^1x_3^1 \vee$$
$$\vee x_1^0x_2^2x_3^0 \vee x_1^1x_2^0x_3^0 \vee x_1^1x_2^0x_3^1 \vee x_1^1x_2^1x_3^0 \vee x_1^1x_2^1x_3^1 \vee \quad (1)$$
$$\vee x_1^1x_2^2x_3^0 \vee x_1^2x_2^0x_3^0 \vee x_1^2x_2^1x_3^0 \vee x_1^3x_2^0x_3^0 \vee x_1^3x_2^1x_3^0.$$

Предикат $P(x_1, x_2, x_3)=t$ можно кратко изобразить в виде, представленном на рис. 1.

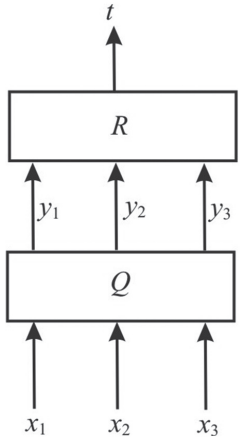


Рис. 1. Расслоение предиката

Здесь Q – классифицирующий слой, который преобразовывает классы x_1, x_2, x_3 в классы

y_1, y_2, y_3 ; R – ассоциирующий слой, в котором все переменные соединяются в единое целое.

Алгоритм состоит из восьми шагов, опишем каждый из них.

Шаг 1. Вводим предикаты эквивалентности $E_i(x_i, \dot{x}_i)$ $(i = \overline{1, m})$, характеризующие классификаторы предиката P. Вычисление производится по формуле, которая может использоваться для любого конечного числа переменных x_m:

$$E_i(x_i, \dot{x}_i) = \forall x_1 \in A_1 \ \forall x_2 \in A_2 ... \forall x_{i-1} \in A_{i-1}$$

$$\forall x_{i+1} \in A_{i+1} ... \ ... \forall x_m \in A_m \ (P(x_1, x_2, ..., x_{i-1}, \quad (2)$$

$$x_i, x_{i+1}, ... x_m) \sim P(x_1, x_2, ..., x_{i-1}, \dot{x}_i, x_{i+1}, ... x_m)).$$

Находим эквивалентность $E_1(x_1, \dot{x}_1)$ для заданного предиката (1) согласно формуле (2):

$$E_1(x_1, \dot{x}_1) = \forall x_2 \in \{0, 1, 2\} \forall x_3 \in \{0, 1\}$$

$$(P(x_1, x_2, x_3) \sim P(\dot{x}_1, x_2, x_3)) =$$

$$= (T_1(x_1) \sim T_1(\dot{x}_1)) \cdot (T_2(x_1) \sim T_2(\dot{x}_1)) \cdot (T_3(x_1) \sim T_3(\dot{x}_1)) =$$

$$= (1 \sim 1) \cdot (T_2(x_1) \sim T_2(\dot{x}_1)) \cdot (0 \sim 0) =$$

$$= \overline{T_2(x_1)} \cdot \overline{T_2(\dot{x}_1)} \vee T_2(x_1) T_2(\dot{x}_1) =$$

$$= \overline{(x_1^0 \vee x_1^1)} \, \overline{(\dot{x}_1^0 \vee \dot{x}_1^1)} \vee (x_1^0 \vee x_1^1)(\dot{x}_1^0 \vee \dot{x}_1^1) =$$

$$= (x_1^0 \vee x_1^1)(\dot{x}_1^0 \vee \dot{x}_1^1) \vee (x_1^2 \vee x_1^3)(\dot{x}_1^2 \vee \dot{x}_1^3).$$

Аналогично находим эквивалентности $E_2(x_2, \dot{x}_2)$, $E_3(x_3, \dot{x}_3)$ для переменных x_2, x_3:

$$E_2(x_2, \dot{x}_2) = \forall x_1 \in \{0, 1, 2, 3\} \forall x_3 \in \{0, 1\}$$

$$(P(x_1, x_2, x_3) \sim P(x_1, \dot{x}_2, x_3)) = (x_2^0 \vee x_2^1)(\dot{x}_2^0 \vee \dot{x}_2^1) \vee x_2^2 \dot{x}_2^2;$$

$$E_3(x_3, \dot{x}_3) = \forall x_1 \in \{0, 1, 2, 3\} \forall x_2 \in \{0, 1, 2\}$$

$$(P(x_1, x_2, x_3) \sim P(x_1, x_2, \dot{x}_3)) = x_3^0 \dot{x}_3^0 \vee x_3^1 \dot{x}_3^1.$$

Шаг 2. Получаем классификаторы предиката P в неявном $F_i(x_i, y_i)$ и в явном $f_i(x_i) = y_i$ видах $(i = \overline{1, m})$ для исходного предиката:

$$F_1(x_1, y_1) = (x_1^0 \vee x_1^1) y_1^0 \vee (x_1^2 \vee x_1^3) y_1^1;$$

$$F_2(x_2, y_2) = (x_2^0 \vee x_2^1) y_2^0 \vee x_2^2 y_2^1;$$

$$F_3(x_3, y_3) = x_3^0 y_3^0 \vee x_3^1 y_3^1.$$

$$f_1(x_1) = y_1 : \begin{cases} x_1^0 \vee x_1^1 = y_1^0; \\ x_1^2 \vee x_1^3 = y_1^1; \end{cases} \quad f_2(x_2) = y_2 : \begin{cases} x_2^0 \vee x_2^1 = y_2^0; \\ x_2^2 = y_2^1; \end{cases}$$

$$f_3(x_3) = y_3 : \begin{cases} x_3^0 = y_3^0; \\ x_3^1 = y_3^1. \end{cases}$$

Шаг 3. Строим классифицирующий слой переключательной цепи предиката P (рис. 2).

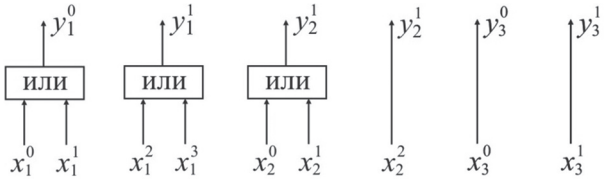


Рис. 2. Переключательная цепь предиката P

Шаг 4. Находим предикат $Q(y_1, y_2, ..., y_m)$, ассоциирующий слои переключательной цепи предиката $P(x_1, x_2, x_3)$. Его вычисление производится по формуле:

$$Q(y_1, y_2, ..., y_m) = \exists x_1 \in A_1 \ \exists x_2 \in A_2 ... \exists x_m \in A_m$$

$$(P(x_1, x_2, ..., x_m) \wedge F_1(x_1, y_1) \wedge F_2(x_2, y_2) \wedge ... \wedge F_m(x_m, y_m)).$$

Для предиката $P(x_1, x_2, x_3)$ получаем следующий результат:

$$Q(y_1, y_2, ..., y_m) = \exists x_1 \in \{0, 1, 2, 3\} \ \exists x_2 \in \{0, 1, 2\}$$

$$\exists x_3 \in \{0, 1\} (P(x_1, x_2, x_3) \wedge F_1(x_1, y_1) \wedge F_2(x_2, y_2) \wedge$$

$$\wedge F_3(x_3, y_3)) = y_1^0 y_2^0 y_3^0 \vee y_1^0 y_2^0 y_3^1 \vee y_1^0 y_2^1 y_3^0 \vee y_1^1 y_2^0 y_3^0.$$

Таблица отношения, соответствующего предикату $Q(y_1, y_2, ..., y_m)$, наборы значений переменных в которой пронумерованы значениями вспомогательной переменной z, представлена в табл. 2.

Таблица 2

Отношение, соответствующее предикату Q

y_1	0	0	0	1
y_2	0	0	1	0
y_3	0	1	0	0
z	0	1	2	3

$Q(y_1, y_2, y_3)$

$R(y_1, y_2, y_3, z)$

Шаг 5. Производим бинаризацию предиката $Q(y_1, y_2, ..., y_m)$. Используем его представление следующей формулой:

$$Q(y_1, y_2, ..., y_m) = \exists z \in C(G_1(y_1, z) \wedge$$

$$\wedge G_2(y_2, z) \wedge ... \wedge G_m(y_m, z)),$$

где

$$G_i(y_i, z) = \exists y_1 \in B_1 \exists ... y_{i-1} \in B_{i-1}$$

$$\exists y_{i+1} \in B_{i+1} ..... \exists y_m \in B_m \ R(y_1, y_2, ..., y_m, z), \ (i = \overline{1, m});$$

$R(y_1, y_2, y_3)$ – предикат, получаемый из предиката $Q(y_1, y_2, ..., y_m)$ нумерацией его конституэнт единицы.

Следовательно для $P(x_1, x_2, x_3)$ имеем:

$$R(y_1, y_2, y_3, z) = y_1^0 y_2^0 y_3^0 z^0 \vee y_1^0 y_2^0 y_3^1 z^1 \vee$$

$$\vee y_1^0 y_2^1 y_3^0 z^2 \vee y_1^1 y_2^0 y_3^0 z^3;$$

$$G_1(y_1, z) = y_1^0 z^0 \vee y_1^0 z^1 \vee y_1^0 z^2 \vee y_1^1 z^3 =$$

$$= y_1^0 (z^0 \vee z^1 \vee z^2) \vee y_1^1 z^3;$$

$$G_2(y_2,\, z) = y_2^0 z^0 \vee y_2^0 z^1 \vee y_2^1 z^2 \vee y_2^0 z^3 =$$
$$= y_2^0 (z^0 \vee z^1 \vee z^3) \vee y_2^1 z^2;$$
$$G_3(y_3,\, z) = y_3^0 z^0 \vee y_3^1 z^1 \vee y_3^0 z^2 \vee y_3^0 z^3 =$$
$$= y_3^0 (z^0 \vee z^2 \vee z^3) \vee y_3^1 z^1.$$

Шаг 6. Строим ассоциирующий слой переключательной цепи предиката P (рис. 3).

$$Q(y_1,\, y_2, y_3) = t$$

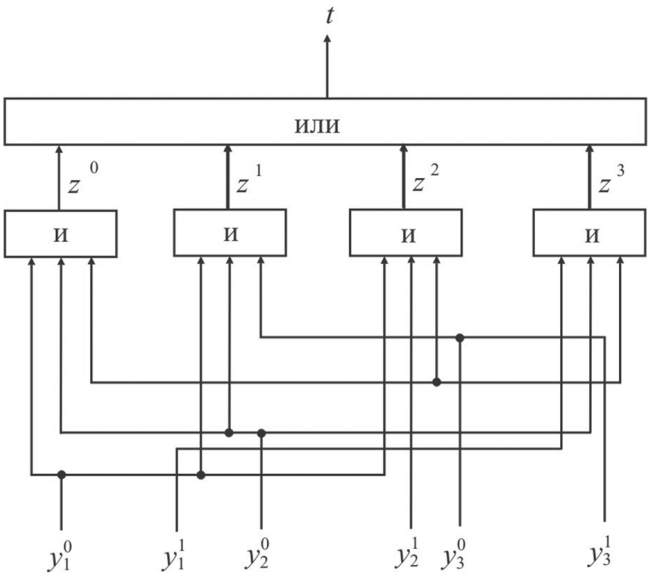


Рис. 3. Ассоциирующий слой переключательной цепи предиката P

Шаг 7. Задаем $t = 1$ и превращаем схему предиката P в реляционную сеть, реализующую отношение P (рис. 4).

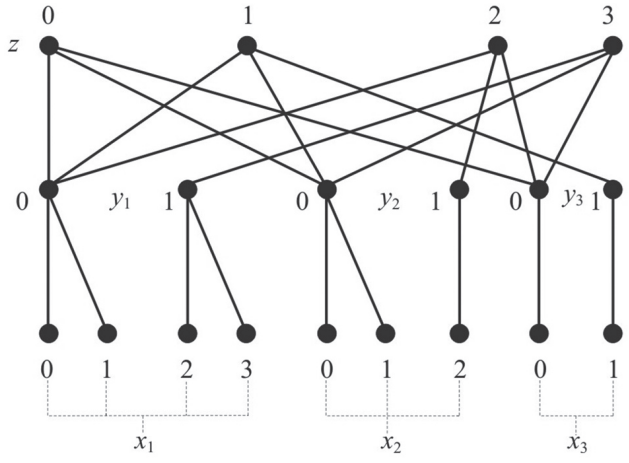


Рис. 4. Реляционная сеть, реализующая отношение $P(x_1,\, x_2,\, x_3)$

Шаг 8. Представляем реляционную сеть в виде многополюсника [3] (рис. 5).

По ветвям реляционной сети происходит преобразование информации при помощи вычисления дизъюнктивного и конъюктивного образов множества по формулам:

$$\exists x \in A(P(x) \cdot K(x,\, y)) = Q_{\max}(y)\,,$$

$$\forall x \in A(P(x) \supset K(x,\, y)) = Q_{\min}(y)\,.$$

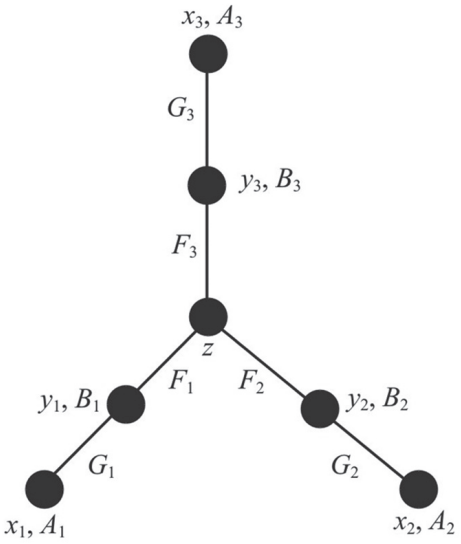


Рис. 5. Реляционная сеть, представленная в виде многополюсника

Сеть отыскивает во всех случаях неискаженные множества корней уравнения $P(x_1,\, x_2, ...,\, x_m) = 1$, а значит она эффективна. Сеть решает уравнения для любых предикатов $P(x_1,\, x_2, ...,\, x_m)$ при любом числе m предметных переменных x. Это означает, что она обладает свойством универсальности. Наличие свойств универсальности и эффективности позволяет рекомендовать так определенную реляционную сеть к применению в качестве решателя мозгоподобной структуры. Универсальная и эффективная реляционная сеть называется совершенной. Любая же рациональная модификация совершенной реляционной сети называется просто реляционной сетью.

Выводы

В статье разработан алгоритм синтеза реляционной сети на основе метода расслоения предиката, который был использован при построении реляционных сетей для моделей различных объектов. Метод расслоения произвольного многоместного предиката позволяет расчленить его в набор одноместных [4] и двуместных предикатов [3]. С помощью предметных переменных исходного предиката все полученные одно- и двуместные предикаты единстсвенным образом соединятся в реляционную сеть исходного многоместного предиката.

При помощи процедуры расслоения предиката существенно усилен метод нулевого прибора, являющийся основным инструментом моделирования психофизических процессов. За счет этого теперь становится возможным моделировать влияние всевозможных внешних факторов, меняющих способ классификации предметов. За счет такого усиления метод нулевого прибора приобретает универсальность при моделировании психофизических процессов.

Список литературы: 1. *Бондаренко, М.Ф.* Теория интеллекта [Текст]: Учебник / М.Ф. Бондаренко, Ю.П. Шабанов-Кушнаренко. – Харьков. Изд-во СМИТ. – 2007. – 576 с. **2.** *Бондаренко, М.Ф.* О методе нулевого прибора [Текст] / М.Ф. Бондаренко, Н.П. Кругликова, Н.Е. Русакова, Ю.П. Шабанов-Кушнаренко // Бионика интеллекта. – 2010. – № 2. – С. 111-115. **3.** *Бондаренко, М.Ф.* О реляционных сетях [Текст] / М.Ф. Бондаренко, Н.П. Кругликова, И.А. Лещинская, Н.Е. Русакова, Ю.П. Шабанов-Кушнаренко // Бионика интеллекта. – 2010. – № 3. – С. 8-13. **4.** *Бондаренко, М.Ф.* Об алгебре одноместных предикатов [Текст] / М.Ф. Бондаренко, Н.П. Кругликова, И.А. Лещинская, Н.Е. Русакова, Ю.П. Шабанов-Кушнаренко // Бионика интеллекта. – 2010. – № 2. – С. 62-67.

Поступила в редколлегию 11.09.2011

УДК 519.7

Про метод розшарування кінцевого предикату/ Н. Є. Русакова // Біоніка інтелекту: наук.-техн. журнал. – 2011. – № 3 (77). – С. 50-53.

У статті ставиться задача розробки методу перетворення математичної моделі довільної кінцевої мозкоподібної структури у технічну эффективно діючу модель, що зветься реляційною мережею.

Табл. 2. Іл. 5. Бібліогр.: 4 найм.

UDC 519.7

About method of stratification of eventual predicate / N.E. Rusakova // Bionics of Intelligense: Sci. Mag. – 2011. – № 3 (77). –P. 50-53.

In the article the task of development of method of transformation of mathematical model of arbitrary eventual braine-like structure is put in its technical effectively operating model, urgent a relation network.

Tabl. 2. Fig. 5. Ref.: 4 items.

УДК 004.942+271.47.15.14.21.21

Yu. Bespalov[1], I. Gorodnyanskiy[2], G. Zholtkevych[3], I. Zaretskaya[4], K. Nosov[5], T. Bondarenko[6], K. Kalinovskaya[7], Y. Carrero[8]

[1]V. N. Karazin Kharkiv National University, Kharkiv, Ukraine, yuri.g.bespalov@univer.kharkov.ua;
[2]V. N. Karazin Kharkiv National University, Kharkiv, Ukraine, biolog@univer.kharkov.ua;
[3]V. N. Karazin Kharkiv National University, Kharkiv, Ukraine, zholtkevych@univer.kharkov.ua;
[4]V. N. Karazin Kharkiv National University, Kharkiv, Ukraine, zar@univer.kharkov.ua;
[5]V. N. Karazin Kharkiv National University, Kharkiv, Ukraine, k-n@nm.ru;
[6]Institute for Problems of Cryobiology and Cryomedicine of the National Academy of Sciences of Ukraine, Kharkiv, Ukraine, cryo@online.kharkov.ua;
[7]Kharkiv Zoo, Kharkiv, Ukraine, kharkovzoo2010@gmail.com;
[8]College of Mount Saint Vincent, New York, USA, yafreisycarrero@ymail.com.

DISCRETE DYNAMICAL MODELING OF SYSTEM CHARACTERISTICS OF A TURTLE'S WALK IN ORDINARY SITUATIONS AND AFTER SLIGHT STRESS

In the paper a class of discrete dynamical models, based on the intra- and between-specific relationships (interactions), which adopted in biology and ecology, is suggested. The relevance of the models in the form of a convergence in probability of sample correlation coefficients is grounded. One of the introduced models, applied to the analysis of a turtle's walk under two states, allowed to reveal deep systemic factors of biomechanics of animal's walking.

DYNAMICAL SYSTEMS, BIOMECHANICS OF WALK, DISCRETE MODELS, SYSTEM IDENTIFICATION, INTERSPECIFIC INTERACTIONS

Introduction

There are concepts that describe an animal's walk as a complex process of system character, which has been known at least since the first half of 20-th century, and are used to analyze the walking characteristics of different animals from human beings to dinosaurs [1, 2, 6, 7, 8]. Beginning with reptiles, the relation between factors guaranteeing stability of an animal's body along with the motion rate in the process of motion is of very importance. Considerable difficulties arise when it is impossible to trace the entire sequence of an animal's motion during each phase. To get over these difficulties, in this paper we used a mathematical apparatus of discrete modeling and dynamic systems with feedback (DMDS), to develop with some of the authors participation [3, 4, 5], a way to obtain the sequence of phases of a cycle of system changes (system trajectory) based on the partial information, when only separate phases are known but not their sequence. The objective of this paper is to investigate the relation between the rate of motion and body stability factors, in a rather simple case, of a turtle's walk using the DMDS method.

1. Theory

In this paper, the authors suggest an approach to determine the relationships between biological objects (say, species) in the framework of some discrete dynamical model. In brief, this approach is presented in illustration [3], which we will discuss in more details.

Let a biological or ecological system be described by N components $A_1, A_2, \ldots, A_N$. These components can have different representations, for example, they can be express the numbers of animals or the amount of biomass of different species. We assume that components only take discrete values $1, 2, \ldots, K$, i. e. K values. The value 1 means a minimum amount of a component, the value K means its maximum value, i. e. a component varies from 1 to K. Indeed, the real range of component's varying may differ from the range $[1, K]$, but for our model the only important thing is that the component varies in some quantitative scale from minimum to maximum.

The value of each of the components is observed and measured at discrete instants of time $t = 0,1,\ldots$. Thus, the values of the component A_i (i. e. i-th component) at the instants of time $t = 0,1,\ldots$ are numbers $A_i(0)$, $A_i(1)$, ….

The trajectory of the system is described by an infinitive-right matrix

$$\begin{pmatrix} A_1(0) & A_1(1) & A_1(2) & \ldots \\ A_2(0) & A_2(1) & A_2(2) & \ldots \\ \vdots & \vdots & \vdots & \ldots \\ A_N(0) & A_N(1) & A_N(2) & \ldots \end{pmatrix}. \quad (1)$$

This trajectory, as always, includes all states of the system at the instants of time $t = 0,1,\ldots$. Hence the state of the system at the instant of time t is the vector $(A_1(t), A_2(t), \ldots, A_N(t))^T$, where T means the matrix transposition. We suppose that the system is strictly determined, and its state at the instant of time t is fully determined by the state at the moment $t-1$. According to the theory of mathematical systems [9], such a system is called a free dynamical system with discrete time, but in our paper we shall use own terminology. Since the

system has only finite number of states (namely, K^N), there exists a positive integer $\mathcal{T}$, for which the conditions of periodicity hold

$$A_j(s) = A_j(s+\mathcal{T}),\ \forall s \geq s_0,$$

for some integer $s_0 > 0$.

It is natural to call a number $\mathcal{T}$ the period of the system. Let us extract the minor

$$\begin{pmatrix} A_1(s) & A_1(s+1) & \dots & A_1(s+\mathcal{T}-1) \\ A_2(s) & A_2(s+1) & \dots & A_2(s+\mathcal{T}-1) \\ \vdots & \vdots & \ddots & \vdots \\ A_N(s) & A_N(s+1) & \dots & A_N(s+\mathcal{T}-1) \end{pmatrix} \quad (2)$$

from the matrix (1) ($s \geq s_0$), which gives full description of the behavior of the system.

Let us introduce a concept of relationships between components. Let $\Omega = \{-,0,+\}$, i. e. the set Ω consists of three elements. We determine a relationship between components A_i and A_j as an entry from the set $\Omega \times \Omega$ and denote it as $\Lambda(A_i, A_j) = (\omega_1, \omega_2)$, where $\omega_1 \in \Omega$, $\omega_2 \in \Omega$. If $\Lambda(A_i, A_j) = (\omega_1, \omega_2)$, this means of this relationship following:

1. If $\omega_1 = \{-\}$ then large values of the component A_j implies decreasing the value of the component A_i.

2. If $\omega_1 = \{0\}$ then the value of the component A_j doesn't influence the value of the component A_i.

3. If $\omega_1 = \{+\}$ then the large values of the component A_j implies increasing the value of the component A_i.

The relationship Λ is antisymmetric, i. e. if $\Lambda(A_i, A_j) = (\omega_1, \omega_2)$, then $\Lambda(A_j, A_i) = (\omega_2, \omega_1)$. It is obvious that all combinations (ω_1, ω_2) correspond to relationships (interspecific interactions) of neutralism, competition, amensalism, predation, commensalism and mutualism, widely used in ecology and biology. We assume, that each component A_j can have with itself only following relationships — $(0,0)$, $(-,-)$ and $(+,+)$, i. e. symmetric relationships.

Assume that all relationships $\Lambda(A_j, A_i)$ between all pairs (A_j, A_i) of components $A_1, A_2, \dots, A_N$ are fixed. Let us define for each A_j the set of components, for which A_j has the relationship (s,u), $s,u \in \Omega$, i. e. (s,u) is some fixed relationship from the set $\Omega \times \Omega$

$$L_j(s,u) = \{A_i \mid \Omega(A_j, A_i) = (s,u)\}.$$

The sets $L_j(+,+)$, $L_j(-,-)$, $L_j(0,0)$ can have from 0 to N entries, other sets ($L_j(\omega_1, \omega_2)$, $\omega_1 \neq \omega_2$) can have from 0 to $N-1$ entries. It is convenient to express relationships by a relationships matrix. If we have N components $A_1, A_2, \dots, A_N$, the relationships matrix is called the following table

$$\begin{bmatrix} & A_1 & A_2 & \dots & A_N \\ A_1 & (\omega_1,\omega_1) & & & \\ A_2 & (\omega_2,\omega_1) & (\omega_2,\omega_2) & & \\ \vdots & \vdots & \vdots & \ddots & \\ A_N & (\omega_N,\omega_1) & (\omega_N,\omega_2) & \dots & (\omega_N,\omega_N) \end{bmatrix}. \quad (3)$$

The relationships above the main diagonal are omitted, since they are recovered by the relationships below the diagonal (according to the antisymmetric property).

Let $\varkappa = \{1,2,\dots,K\}$ and $N_j(s,u)$ is the number of components in the set $L_j(s,u)$, $j = 1,2,\dots,N$. A transition from the state $(A_1(t), A_2(t), \dots, A_n(t))^T$ to the state $(A_1(t+1), A_2(t+1), \dots, A_n(t+1))^T$ is described by N transition functions F_j. Each function defines the mapping

$$\varkappa^{N_j(+,+)+N_j(+,0)+N_j(+,-)+N_j(-,+)+N_j(-,0)+N_j(-,-)} \to \varkappa.$$

This mapping in symbolic form may be expressed by the formula

$$\begin{aligned} A_j(t+1) = F_j(&A_k(t) \in L_j(+,+), A_k(t) \in L_j(+,0), \\ &A_j(t) \in L_k(+,-), A_k(t) \in L_j(-,+), \\ &A_k(t) \in L_j(-,0), A_k(t) \in L_j(-,-)),\ j = 1,2,\dots,N, \end{aligned} \quad (4)$$

where $A_k(t) \in L_j(+,+)$, $A_k(t) \in L_j(+,0)$, … are the values $A_k(t)$ of all A_k, belonging to $L_j(+,+)$, $L_j(+,0)$, … correspondingly.

The transition function, introduced by the above formula, is quite natural in its structure. The given component A_j is influenced only by those components, which indeed influence A_j, i. e. the components from the sets $L_j(+,\omega)$ and $L_j(-,\omega)$ for any $\omega \in W$.

Now, let us describe types of relationships, inherent to real biological and ecological systems.

The formula (4) expresses a general form of transition of the system from the state at the moment t to the moment $t+1$. For a more detailed description of the behavior of biological or ecological system, we have to specify an explicit form of transitional functions, which express dynamical properties of the system.

We suggest two approaches to such a dynamics, which are based on concepts of biological interactions.

Let us introduce the following functions defined on the set $\varkappa$

$$\mathrm{I}nc(A) = \min\{K, A+1\},$$
$$\mathrm{D}ec(A) = \max\{1, A-1\}.$$

First we define a type of relationships, which takes into account the weighted sum of all $A_j(t)$ (inclusive $A_i(t)$) for calculating the value of component A_i at the instant of time $t+1$. We call this type of relationships a weight functions' approach. Now, this is the exact definition.

For each j ($j = 1,2, \dots, N$) we introduce a set of functions $\varphi_{j,1}^{\langle s,u\rangle}(\cdot)$, $\varphi_{j,2}^{\langle s,u\rangle}(\cdot)$, …, $\varphi_{j,N_j}^{\langle s,u\rangle}(\cdot)$. These are the functions of interactions of those components, where A_j has relationships (s,u), $s \in \{+,-\}$, $u \in W$. The properties of these functions are the following:

1. The functions are defined on the discrete set $\varkappa$.

2. $\varphi_{j,k}^{\langle +,+\rangle}(\cdot)$, $\varphi_{j,k}^{\langle +,0\rangle}(\cdot)$, $\varphi_{j,k}^{\langle +,-\rangle}(\cdot)$ are increasing functions.

3. $\varphi_{j,k}^{\langle -,+\rangle}(\cdot)$, $\varphi_{j,k}^{\langle -,0\rangle}(\cdot)$, $\varphi_{j,k}^{\langle -,-\rangle}(\cdot)$ are decreasing functions.

4. $\varphi_{j,k}^{\langle +,+\rangle}(1) = \varphi_{j,k}^{\langle +,0\rangle}(1) = \varphi_{j,k}^{\langle +,-\rangle}(1) =$
$= \varphi_{j,k}^{\langle -,+\rangle}(1) = \varphi_{j,k}^{\langle -,0\rangle}(1) = \varphi_{j,k}^{\langle -,-\rangle}(1) = 0$.

Now define a set of numbers $\delta_j > 0$ ($j = 1,2,\ldots,N$) and call them thresholds of sensivity. For the system's state at the instant of time t the following value is calculated

$$d_j = \sum_{A_k \in L_j(+,+)} \varphi_{j,k}^{\langle +,+\rangle}(A_k(t)) + \sum_{A_k \in L_j(+,0)} \varphi_{j,k}^{\langle +,0\rangle}(A_k(t)) + \sum_{A_k \in L_j(+,-)} \varphi_{j,k}^{\langle +,-\rangle}(A_k(t)) + \sum_{A_k \in L_j(-,+)} \varphi_{j,k}^{\langle -,+\rangle}(A_k(t)) + \sum_{A_k \in L_j(-,0)} \varphi_{j,k}^{\langle -,0\rangle}(A_k(t)) + \sum_{A_k \in L_j(-,-)} \varphi_{j,k}^{\langle -,-\rangle}(A_k(t)), \quad (5)$$

(it is clear that d_j depends on t, but t is omitted for short in the left side).

The value of the component A_j changes according to the value d_j by the following rules

1. if $d_j \geq \delta_j$, then $A_j(t+1) = \mathrm{Inc}(A_j(t))$;
2. if $d_j \leq -\delta_j$, then $A_j(t+1) = \mathrm{Dec}(A_j(t))$;
3. if $-\delta_j < d_j < \delta_j$, then $A_j(t+1) = A_j(t)$.

Now we can explain the mean of introduced transition functions. For example, the functions $\varphi_{j,k}^{\langle -,+\rangle}(\cdot)$ ($k = 1,2,\ldots,N_j(-,+)$) reflects the influence upon component A_j of those components, which related with A_j by relationship $(-,+)$, i. e. the components from the set $L_j(-,+)$. The greater this influence (i. e. the greater values of $A_i(t)$ from the set $L_j(-,+)$), the less values of d_j. An influence of other components, where A_j has other relations, are ``weighted'' in similar way. If cumulative influence of the components, interacting with A_j and expressed by (5), exceeds the threshold value δ_j, then the value of A_j changes by unit.

From the rules for definition of transitional function one can observe, that an increment of the value of A_j is less or equal to 1 ($|A_j(t+1) - A_j(t)| \leq 1$). This means, that the rate of changes of A_j is invariable. It is possible to avoid such an unnatural restriction, for example, by introducing a dependence of the increment on $|d_j|/\delta_j$. However, in this paper we do not consider such extensions.

It is clear that the threshold δ_j influences the dynamics of the system in following way: the greater δ_j, the greater absolute value of the weighted sum d_j required for overcoming δ_j in changing the value of A_j. So, if δ_j is very large, the system becomes very inert. When

$$\delta_j > \max\{\sum_{A_k \in L_j(+,+)} \varphi_{j,k}^{\langle +,+\rangle}(K) + \sum_{A_k \in L_j(+,0)} \varphi_{j,k}^{\langle +,0\rangle}(K) + \sum_{A_k \in L_j(+,-)} \varphi_{j,k}^{\langle +,-\rangle}(K), -(\sum_{A_k \in L_j(-,+)} \varphi_{j,k}^{\langle -,+\rangle}(K) + \sum_{A_k \in L_j(-,0)} \varphi_{j,k}^{\langle -,0\rangle}(K) + \sum_{A_k \in L_j(-,-)} \varphi_{j,k}^{\langle -,-\rangle}(K))\},$$

the value of A_j never changes ($A_j(t) = const$, $t = 0,1,\ldots$). If we wish to avoid this trivial case, the value of δ_j should be not very large.

A second approach, proposed here, is based on the famous Justus von Liebich's law of limiting factors. This concept was originally applied to plant or crop growth. This approach is described in brief [3] and now we give detailed description of this approach. Our following results are based on it.

Assume, that the system of relationships between A_1, A_2, …, A_N is given. Let us introduce two constant matrices C and C^* of size $N \times N$. The transition functions are based on the following algorithm.

Let the system in the instant of time t has the state $(A_1(t), A_2(t), \ldots, A_N(t))^T$ and A_j is an arbitrary fixed component. Let i runs from 1 to N, by u we denote any entry from the set W.

1. If for the current i the equality $\Lambda(A_j, A_i) = (-,u)$ holds, we assume

$$f_i = \begin{cases} -1, & \text{if } A_i(t) \geq c_{ji}^*, \\ 0, & \text{if } c_{ji} + 1 \leq A_j(t) \leq c_{ji}^* - 1, \\ 1, & \text{if } A_j(t) \leq c_{ji}. \end{cases}$$

Note, that no matter the specific value of u, only the influence on A_j from the side of A_i plays the role.

2. If for the current i the equality $\Lambda(A_j, A_i) = (+,u)$ holds, we assume

$$f_i = \begin{cases} -1, & \text{if } A_j(t) \leq c_{ji}, \\ 0, & \text{if } c_{ji} + 1 \leq A_j(t) \leq c_{ji}^* - 1, \\ 1, & \text{if } A_j(t) \geq c_{ji}^*. \end{cases}$$

3. If for the current i the equality $\Lambda(A_j, A_i) = (0,u)$ holds, we assume $f_i = 1$.

After the cycle termination, we obtain the sequence f_1, f_2, …, f_N. Then we can calculate the value $A_j(t+1)$ according to the following rule:

$$A_j(t+1) = \begin{cases} \mathrm{Dec}(A_j(t)), & \text{if } \min\limits_{1 \leq i \leq N}\{f_i\} = -1, \\ A_j(t), & \text{if } \min\limits_{1 \leq i \leq N}\{f_i\} = 0, \\ \mathrm{Inc}(A_j(t)), & \text{if } \min\limits_{1 \leq i \leq N}\{f_i\} = 1. \end{cases} \quad (6)$$

Applying this algorithm for each $j = 1, 2, \ldots, N$, we shall obtain the system's state at the instant of time $t+1$.

Now we can explain the mean of an introduced transition from t to $t+1$. E. g., stating that the given component A_j has the relationship $(+,-)$, which is the current component A_i (see algorithm). According to the mean of relation $(+,-)$, large values of A_i lead to decreasing A_j. Indeed, according to the item 1 of the algorithm, if $A_i(t) \geq c_{ji}^*$ (i. e. $A_i(t)$ is "large enough"), $f_i = -1$ and, according to (6), A_j will decrease if $A_j(t) > 1$. Other cases of this transition works analogically.

When we investigate real data, we do not observe the dynamics, described by relationships (3), by the matrix of the trajectory (1) or by its minor (2).

The result of this observation is the following table of cases

$$\tilde{M} = \begin{pmatrix} C_{11} & C_{12} & \dots & C_{1B} \\ C_{21} & C_{22} & \dots & C_{2B} \\ \vdots & \vdots & \ddots & \vdots \\ C_{N1} & C_{N2} & \dots & C_{NB} \end{pmatrix},$$

where columns correspond to cases and rows correspond to components (N components and B cases).

We propose an algorithm that reveals the system relationships of above mentioned type, on the base of transition functions F_j, $j = 1,2,\dots,N$ and the observation table $\tilde{M}$.

This algorithm allows us to determine between- and intra-components relationships, which are as close as possible relationships that form matrix (2) in a certain mean. Assume, that a number K and transition functions are given. In this case, for initial will be $(A_1(0), A_2(0)$, …, $A_N(0))^T \in \varkappa^N$ and the given sets $L_1(u,s)$, $L_2(u,s)$, …, $L_N(u,s)$, $u \in \{-,0,+\}$, $s \in \{-,0,+\}$, which makes it possible to calculate the matrix (1) or the minor (2). Let

$$P = \begin{pmatrix} 1 & r_{12} & \dots & r_{1N} \\ r_{21} & 1 & \dots & r_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ r_{N1} & r_{N2} & \dots & 1 \end{pmatrix}$$

be a Pearson correlation matrix between rows of minor (2). Now for the matrix $\tilde{M}$ let us calculate Pearson correlation matrix of its rows

$$\tilde{P} = \begin{pmatrix} 1 & \rho_{12} & \dots & \rho_{1N} \\ \rho_{21} & 1 & \dots & \rho_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ \rho_{N1} & \rho_{N2} & \dots & 1 \end{pmatrix}.$$

Let us introduce the measure of distance between correlation matrices P and $\tilde{P}$

$$D(P,\tilde{P}) = \sum_{j=1}^{N-1} \sum_{j=i+1}^{N} (r_{ij} - \rho_{ij})^2. \qquad (7)$$

Consider the task of minimization $D(P,\tilde{P})$ by all possible vectors of initial states $(A_1(0)$, $A_2(0)$, …, $A_N(0))^T \in \varkappa^N$ and all allowable sets $L_j(s,u)$, $s,u \in \mathrm{W}$ for all j

$$D(P,\tilde{P}) \mapsto \min$$

by all initial states and by all allowable $L_j(s,u)$.

Now, we can explain the mean of the stated task. Suppose, that a process in some real system is described by cyclical trajectory (2). One cannot the possibility to observe the dynamic of this trajectory, i. e. a full length cycle. The observation are taken from the system at random instants of time t from s to $s+\mathcal{T}-1$ with equal probability. When an observation is fixed, the column $(A_1(t)$, $A_2(t)$, …, $A_N(t))^T$ from (2) is attached to table of observations. In other words, the columns of table of observations $\tilde{M}$ are obtained from (2) by an equiprobable choice of columns. The stated task means a search of such relationships between components, that the minor (2) is to be as close as possible to the table of observations in the mean of the measure (7).

The following theorem shows, that this task is well-grounded in some mean.

If the table of observations $\tilde{M}$ is obtained from the minor (2) by equiprobable choice of columns, then the correlation matrix of the observations table $\tilde{P}$ converges to the correlation matrix of minor P (in probability)

$$\lim_{B\to\infty} \rho_{ij} = r_{ij},\ i = 1,2,\dots,N,\ j = 1,2,\dots,N.$$

Proof. Since the Pearson coefficient is a pairwise characteristics (between two variables), it is enough to prove the theorem for the case $N = 2$. Let

$$\begin{pmatrix} x_{1,1} & x_{1,2} & \dots & x_{1,\mathcal{T}} \\ x_{2,1} & x_{2,2} & \dots & x_{2,\mathcal{T}} \end{pmatrix}$$

be the minor (2), where we use the notation $x_{i,j}$ instead $A_i(j+s-1)$ for convenience. Let $\bar{x}_i = \frac{1}{\mathcal{T}} \sum_{j=1}^{\mathcal{T}} x_{i,j}$, $i = 1,2$, are to be the means of rows. If a sample variance of both rows is not 0, the Pearson correlation coefficient (between rows) is equal to

$$R_0 = \frac{\sum_{j}^{\mathcal{T}} (x_{1,j} - \bar{x}_1)(x_{2,j} - \bar{x}_2)}{\sqrt{\sum_{j=1}^{\mathcal{T}} (x_{1,j} - \bar{x}_1)^2} \sqrt{\sum_{j=1}^{\mathcal{T}} (x_{2,j} - \bar{x}_2)^2}}.$$

Now we can present the observation matrix $\tilde{M}$ as a frequency table

$$\begin{bmatrix} x_{1,1} & x_{1,2} & \dots & x_{1,\mathcal{T}} \\ x_{2,1} & x_{2,2} & \dots & x_{2,\mathcal{T}} \\ m_1 & m_2 & \dots & m_{\mathcal{T}} \end{bmatrix},$$

where m_j is a frequency of the column $(x_{1,j}, x_{2,j})^T$, which is taken from the minor (2) and placed into the observation matrix $\tilde{M}$.

The means of the rows of the observation matrix $\tilde{M}$ are

$$\bar{C}_i = \frac{1}{B} \sum_{k=1}^{B} C_{i,j} = \frac{1}{B} \sum_{j=1}^{\mathcal{T}} m_j x_{i,j} = \sum_{j=1}^{\mathcal{T}} \frac{m_j}{B} x_{i,j}, i = 1,2.$$

According to the Bernoulli theorem [10], $\frac{m_j}{B} \to \frac{1}{\mathcal{T}}$ (in probability, when $B \to \infty$) for all $j = 1,2,\dots,\mathcal{T}$. From this it follows $\bar{C}_i \to \bar{x}_i$ (in probability, $B \to \infty$), $i = 1,2$.

The Pearson correlation coefficient (between rows) of the observation matrix $\tilde{M}$ equals

$$R_B = \frac{\sum_{k=1}^{\mathcal{T}} \frac{m_k}{B} (x_{1,k} - \bar{x}_1)(x_{2,k} - \bar{x}_2)}{\sqrt{\sum_{k=1}^{\mathcal{T}} \frac{m_k}{B} (x_{1,k} - \bar{x}_1)^2} \sqrt{\sum_{k=1}^{\mathcal{T}} \frac{m_k}{B} (x_{2,k} - \bar{x}_2)^2}}.$$

Then,

$$\lim_{B\to\infty} R_B =$$

$$\lim_{B\to\infty} \frac{\sum_{k=1}^{\mathcal{T}} \frac{m_k}{B}(x_{1,k}-\bar{C}_1)(x_{2,k}-\bar{C}_2)}{\sqrt{\sum_{k=1}^{\mathcal{T}} \frac{m_k}{B}(x_{1,k}-\bar{C}_1)^2}\sqrt{\sum_{k=1}^{\mathcal{T}} \frac{m_k}{B}(x_{2,k}-\bar{C}_2)^2}} =$$

$$\lim_{B\to\infty} \frac{\sum_{k=1}^{\mathcal{T}} \frac{m_k}{B}(x_{1,k}-\bar{x}_1)(x_{2,k}-\bar{x}_2)}{\sqrt{\sum_{k=1}^{\mathcal{T}} \frac{m_k}{B}(x_{1,k}-\bar{x}_1)^2}\sqrt{\sum_{k=1}^{\mathcal{T}} \frac{m_k}{B}(x_{2,k}-\bar{x}_2)^2}} =$$

$$\frac{\sum_{k=1}^{\mathcal{T}} \frac{1}{\mathcal{T}}(x_{1,k}-\bar{x}_1)(x_{2,k}-\bar{x}_2)}{\sqrt{\sum_{k=1}^{\mathcal{T}} \frac{1}{\mathcal{T}}(x_{1,k}-\bar{x}_1)^2}\sqrt{\sum_{k=1}^{\mathcal{T}} \frac{1}{\mathcal{T}}(x_{2,k}-\bar{x}_2)^2}} = R_0 .$$

(The convergence holds in probability).

The theorem is proven.

This result means that if the real dynamical periodic process is described by the minor (2), we can expect that the correlation matrix of the observations table will be close to correlation matrix of real process, i. e. to the correlation matrix of the minor.

So, as mentioned above, we can consider the task (7) as the task of system identification. Initial states $(A_1(0)$, $A_2(0)$, …, $A_N(0))^T$ and sets $L_j(s,u)$ in (7) are parameters, which should be identified.

2. Results

By using digital photography, we captured separate walking phases (which by no means form a complete cycle) of a two year-old male *Emys orbicularis* along the bottom of an enameled pool located in a terrarium. In the case of using a stressor, a turtle was kept lying on its back for two minutes. Then another turtle was immediately put into the pool in a back up position (without using a stressor). Using the images, we calculated the ratio of the following distances to the length of the turtle's shell: from the tail head to the ankle of each of the four legs — the right foreleg (**rf**), the right hand leg (**rb**), the left foreleg (**lf**), the left hand leg (**lb**). We considered such a distance to reflect the degree of a leg straightening.

Using DMDS, the structures of relationships and sets of states of the four-component system were obtained for all four parameters **rf**, **rb**, **lf**, **lb** (the components of the system are the turtle's legs). The results correspond to the observations both for the stressed and non-stressed cases in the outmost degree (in the conventional sense for DMDS).

For modeling with DMDS, we used the approach based of the principles of the von Liebig law and proposed that $K = 3$ (three levels of components' values). For these two cases, the system trajectories which showed the sequence of walking phases (the cycle including different combinations of contracting and straightening of each of four legs) were constructed. On these trajectories, the system factors analyzed the body's stability for the stressed and the non-stressed turtles walk.

A marked difference between the system trajectories for each case with and without the stressor was recovered. In the case of the turtle without the stressor, the DMDS obtained the following data in the system trajectory. This is represented in the table 1.

Table 1. System trajectory for the case without stressor.

rf	3	3	3	3	2	1	1	**1**	**2**	**3**	3
rb	1	**1**	**2**	3	3	3	2	1	1	1	1
lf	1	1	1	**1**	**2**	2	1	1	1	1	1
lb	2	1	1	1	1	1	**1**	**2**	**3**	3	2
Conditional time, steps	1	2	3	4	5	6	7	8	9	10	11

The bold font highlights the sequences of monotone increase of **rf**, **rb**, **lf** and **lb** values which corresponds to apparent motion of extremities (in this paper we neglect differences in the directions of motion). This system trajectory shows that the extremities start their motion in patterns so that only one leg keeps moving, and only at the end of each phase they start to move the other one. Thus, most of the time the turtle's body has three supporting points which guarantees stability of its motion.

In the case of the turtle with the stressor, the DMDS data obtained the following data in the system trajectory. This is represented in the table 2.

Table 2: System trajectory for the case with stressor.

rf	**1**	**2**	2	1	1	1	1	2	3	3	3	2	1	1	1
rb	1	1	**1**	**2**	**3**	3	3	2	1	1	1	1	1	1	1
lf	**1**	**2**	**3**	3	3	2	1	1	1	2	2	1	1	1	1
lb	1	1	1	1	1	1	1	1	**1**	**2**	**3**	3	3	2	1
Conditional time, steps	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15

This system trajectory shows that in contrast to the case of a non-stressed turtle the motion of a stressed turtle has phases where two legs are simultaneously moving so there is no guarantee that the body has three supporting points to maintain stability of its motion.

The results of the DMDS allows us to put forward a working hypothesis: that even under very slight stress in the motion of a turtle there occurs a deviation from the optimal relation between factors providing motion rate, and stability to the deterioration of conditions for stable motion. This deviation can be registered by systems of remote sensing (say, by digital photography) and further analyzed by DMDS with the use of an animal's motion trajectory based on the images representing the separate phases of motion. Information about the time sequence of these phases is not required, so images representing any incomplete fragments of a motion cycle can be used. In the case with the turtle, the requirements to refine the images are not so high. The images can be obtained by digital photography from any altitude, which allows us to alter the resolution degree and recognize the position of the animal's legs at a position with respect to its body silhouette.

Discussion

One should keep in mind that DMDS is mainly the method that we used to form the working hypotheses. The working hypothesis in this paper as well as system trajectories illustrating it, might present some theoretical interest to study rather simple cases of an animals' motion and some practical interest, for example, for the development of systems of ecological monitoring based on the remote (aerospace) methods of photographing animals in nature with further computer image processing of their silhouettes. Deliberately, the important feature is possibly to use the DMDS process and analyze the arrays of incomplete information with data of fragmentary observations for separate phases of an animals' motion, in conditions when it is impossible to observe the time sequence of all motion phases (say, presence of animals' shelters, limited time of photographing etc.), but on the base of which the whole cycle of their motion can be restored.

There are well known papers on mathematical modeling in the biomechanics of animal, reptiles, dinosaurs in particular, and other fossil of animals that have incomplete information [11, 2, 6, 7, 12, 13]. In all of these cases the working hypothesis about the structure of relations between components of a modeled multicomponent system is needed (in the case of dinosaurs components are parts of legs taking different but interrelated positions in the process of motion). This working hypothesis should be constructed independently from the results of numerical experiments on the given mathematical model, say, based on the data about the structure and position of the tail of a two-legged dinosaur and analogies with biomechanics of the recent animals. We speak about a structure which contains a certain set of relations from the following list of possible pairs of influences on each other component in a multi-component system of any kind: $(+,+)$, $(-,-)$, $(-,+)$, $(-,0)$ ", $(+,0)$ and $(0,0)$. (The procedures of DMDS explains that if the previous value of an component, which is a subject to influence, is high then "minus" the influence leads to decreasing and a "plus" influence leads to the increasing of the current value of an component, which is an object of influence; "zero" influence stabilizes the current values of an component, which is an object of influence, on the level of previous ones). In the case of DMDS, the working hypothesis mentioned above appears as a result of modeling, and this provided successful results.

References: 1. Poinar, G.Jr., Poinar R.:What Bugged the Dinosaurs?: Insects, Disease, and Death in the Cretaceous, Princeton University Press (2007). 2. Spotilla, J.R., Lommen, P.W., Bakken G.S., Gates, D.M.: A mathematical model for body temperature of large reptiles: Implications for dinosaur endothermy. Am. Nat. 107, 391-404 (1973). 3. Bespalov, Yu.G., Derecha, L.N., Zholtkevych, G.N., Nosov, K.V. Discreet model of the system with negative feedback. Bulletin of V. N. Karazin Kharkiv National University. Series "Mathematical Modeling. Information Technology. Automated Control Systems'', 833, 27-38 (2008). 4. Bespalov, Yu.G., Zholtkevych, G.N., Nosov, K.V., Marchenko, V.S., Marchenko, G.P., Psarev, V.A., Utevsky, A. Yu.: Contribution to heliobiological effects investigation with the help of Discrete Modeling of Dynamical Systems with feedback. In: 8th Gamow Summer School "Astronomy and beyond: Astrophysics, Cosmology, Radioastronomy and Astrobiology'', pp. 12-13. Odessa, (2008). 5. Zorya, A.V., Nosov, K.V., Bespalov, Yu.G.: On mathematical methods for forecasting population dynamics of murine rodents of Kharkiv region. In: Proceedings of 12th Kharkiv final regional theoretical and practical conference. Kharkiv, (2009). 6. Hutchinson, J.R., Ng-Thow-Hing, V., Anderson, F.C.: A 3D interactive method for estimating body segmental parameters in animals: application to the turning and running performance of Tyrannosaurus rex. Journal of Theoretical Biology, 246, 660-680 (2004). 7. Hutchinson, J.R.: Biomechanical modeling and sensitivity analysis of bipedal running ability. II. Extinct taxa. Journal of Morphology, 262, 441-461 (2004). 8. Christian, A., Horn, H.-G., Preuschoft, H.: Biomechanical reasons for bipedalism in reptiles. Amphibia-Reptilia, 15, 275-284 (1994). 9. Kalman, R.E., Falb P.L., Arbib, M.A.: Topics in mathematical system theory. McGraw-Hill, New York, (1969). 10. Gnedenko, B.V.: Theory of Probability. CRC Press (1998). 11. Pontzer, H., Allen, V., Hutchinson, J.R.: Biomechanics of Running Indicates Endothermy in Bipedal Dinosaurs. PLoS ONE, 11, e7783 (2009). 12. Hart, R.T., Hennebel, V.V., Thongpreda, N., Buskirk, W.C.V., Anderson, R.C.: Journal of Biomechanics, 25, 261-286 (1992). 13. Papageorgiou, N.S., Kyritsi-Yiallourou, S.T.: Handbook of Applied Analysis. Springer (2009).

Поступила в редколлегию 20.06.2011

УДК 004.942+271.47.15.14.21.21

Дискретне динамічне моделювання системних характеристик ходи черепахи у звичайних обставинах та після легкого стресу / Ю.Г. Беспалов, І.Д. Городнянський, Г.М. Жолткевич, І.Т. Зарецька, К.В. Носов, Т.П. Бондаренко, К.М. Калиновська, Я. Карреро // Біоніка інтелекту: наук.-техн. журнал. – 2011. – № 3 (77). – С. 54-59.

У статті пропонується дискретна динамічна модель, що дозволяє виразити структуру внутрішньо- та зовнішньокомпонентних відносин динамічної системи у термінах міжвидової взаємодії, прийнятої у біології та екології. Запропонована модель застосовується до вивчення системних аспектів біомеханіки ходи черепахи у спокійному стані та після легкого стресу.

Табл. 2. Бібліогр.: 13 найм.

УДК 004.942+271.47.15.14.21.21

Дискретное динамическое моделирование системных характеристик ходьбы черепахи в обычных условиях и после легкого стресса / Ю.Г. Беспалов, И.Д. Городнянский, Г.Н. Жолткевич, И.Т. Зарецкая, К.В. Носов, Т.П. Бондаренко, Е.М. Калиновская, Я. Карреро // Бионика интеллекта: наук.-техн. журнал. – 2011. – № 3 (77). – С. 54-59.

В статье предлагается дискретная динамическая модель, которая позволяет выразить структуру внутри- та внешнекомпонентных отношений динамической системы в терминах межвидовых взаимодействий, принятых в биологии та экологии. Предложенная модель применяется для изучения системных аспектов биомеханики ходьбы черепахи в спокойном состоянии и после легкого стресса.

Табл. 2. Библиогр.: 13 назв.

УДК 519.81

Э.Г. Петров[1], Е.В. Губаренко[2]
[1] ХНУРЕ, м. Харків, Україна, ST@kture.kharkov.ua;
[2] ХНУРЕ, м. Харків, Україна, vergeley@mail.ru

РОЛЬ, ЗАДАЧИ И МЕТОДЫ ГОСУДАРСТВЕННОГО УПРАВЛЕНИЯ ПРИ РЕАЛИЗАЦИИ КОНЦЕПЦИИ УСТОЙЧИВОГО РАЗВИТИЯ

Описан процесс восприятия государством социальных ресурсов и методы их описания. Проанализирован процесс обмена живого труда на универсальный ресурс. Сформулированы целевые требования к государственным органам управления. Подчеркнута необходимость ориентации государства на системы здравоохранения и образования

СОЦИАЛЬНЫЕ РЕСУРСЫ, ГОСУДАРСТВО, ФОРМИРОВАНИЕ БЮДЖЕТА, ЖИВОЙ ТРУД, УНИВЕРСАЛЬНЫЙ РЕСУРС, ЗДРАВООХРАНЕНИЕ, СИСТЕМА ОБРАЗОВАНИЯ

Введение

Последнее десятилетие ознаменовалось все более частыми и глубокими финансово-экономическими кризисами, которые в настоящее время превратились в перманентные. Этот процесс сопровождается расслоением общества, увеличением разрыва между благополучными и развивающимися странами, ростом социальной напряженности и экстремизма, общей политической нестабильностью. Неустойчивость социально-экономической системы (СЭС) сопровождается обострением экологической обстановки, возрастающей угрозой изменения климата, общей деградацией среды обитания. Все это в целом стало логическим итогом, как было отмечено на саммите ООН в Рио-де-Жанейро в 1992, концепции экономического роста, при которой экономические показатели доминируют в ущерб социальным и экологическим. В условиях стихийного рынка, неограниченной конкуренции, прибыли «любой ценой» возникает ситуация неконтролируемого, зачастую хищнического, использования всех видов ресурсов: природных, экологических, трудовых. Подобный подход привел цивилизацию к критическому состоянию. В качестве альтернативы была сформулирована концепция устойчивого развития мировой СЭС, предусматривающая гармоничное, сбалансированное, научно и морально обоснованное развитие человечества [1].

Концепция устойчивого развития получила декларативную поддержку большинства стран мира. Она обсуждалась на саммите Тысячелетия (Нью-Йорк, 2000 г.), всемирном саммите по Устойчивому развитию (Йоханнесбург, 2002 г.), конференциях в Киото (Япония) и Брюсселе. Однако, практические итоги двух прошедших десятилетий, особенно в социальной сфере близки к нулевым. В экологической сфере дальше деклараций о готовности принимать меры, ситуация так же не продвинулась. В результате СЭС в измерении «устойчивого развития» продолжает деградировать возрастающими темпами.

Большинство государств и представляющие их политические деятели понимают необходимость изменения ситуации, но хотят остаться в стороне, сохранив идеологию концепции экономического роста, рассчитывая, что другие государства компенсируют их безучастность собственными ресурсами. Такая ситуация обусловлена тем, что правительства всех стран, которые хотя бы декларативно заявляют о демократии, являются заложниками общественного мнения, а изменение взглядов и переориентирование их на новую концепцию требуют проявления сильной политической воли, что обществом всегда воспринималось крайне негативно.

Вторым аспектом, сдерживающим практическую реализацию концепции устойчивого развития, является тот факт, что политическая жизнь во всех странах превратилась в бизнес-проект, когда группа лиц, финансирует продвижение некоторого индивида, который впоследствии становится выразителем их воли и защитником их интересов. Другими словами, демократия практически во всех странах выродилась в разновидности олигархии. Государство при таком способе правления выражает интересы не всего общества как целого, а только узкой социальной группы.

Вместе с этим, из-за роста недовольства в обществе, которое выражается в виде протестов, забастовок, митингов, акций неповиновения вплоть до вооруженных столкновений и революций, государства вынуждены переориентировать свою политику на более полное удовлетворение потребностей общества в целом и составляющих его индивидов.

Однако изменения должны быть системно аргументированы, проявлять признаки эволюционной наследственности, создавать условия для удовлетворения потребностей всех членов общества, сохраняя устойчивость СЭС. Реализация этих задач требует существенного изменения роли государственного управления, разработки новых моделей,

методов и средств их реализации, особенно в социальной сфере.

Цель настоящей статьи заключается в системном анализе общественных и личных потребностей социума как системообразующего элемента государства, а также обосновании роли и задач государственного управления национальной социальной системой и определении возможных способов их решения в условиях перехода к концепции устойчивого развития.

1. Роль и место социального ресурса в государстве

Государство является организационной формой СЭС, которая является сложной целенаправленной системой, элементами которой выступают социум (социальные группы), субъекты экономических процессов и природно-экологическая среда. Особая роль социального элемента заключается в том, что он единственный является активным, т.е. способен формировать цели СЭС.

Концептуальная (общая) цель СЭС заключается в том, что она должна как можно полно удовлетворять материальные и духовные потребности социума в целом и каждого индивида в отдельности.

В свою очередь, все три элемента СЭС являются сложными метасистемами. Элементами социальной системы выступают индивидуумы, которые на основе межличностных отношений образуют сложную многоуровневую социальную структуру, порождающую основное свойство СЭС – ориентацию на удовлетворение потребностей общества. Реализация указанного свойства основана на производстве благ путем трансформации природно-экологических ресурсов с помощью живого труда в материальные и духовные блага посредством экономической системы. Это означает, что именно экономическая система является средством достижения цели СЭС как целостной метасистемы. В таком контексте живой труд является уникальным незаменимым ресурсом, а его количество и качество определяют эффективность СЭС.

Трудность реализации цели СЭС обусловлена сложностью структуры социальной системы. Эта сложность заключается в том, что все элементы социальной системы являются активными целенаправленными элементами с уникальными потребностями по различным характеристикам и показателям. Это означает, что непосредственное удовлетворение потребностей каждого индивида невозможно. Поэтому в процессе развития цивилизации сложился следующий механизм удовлетворения индивидуальных потребностей: индивидуум посредством экономической системы обменивает свой живой труд на универсальный ресурс – деньги, которые затем путем обмена трансформирует в любые материальные и духовные блага.

Такой механизм в процессе развития СЭС привел к тому, что на некотором этапе произошла подмена целей: цель удовлетворения потребностей каждого индивида была заменена целью максимизации прибыли (денежных средств) социальных групп, что вылилось в концепцию экономического роста и привело цивилизацию к границе устойчивости за счет хищнического, неконтролируемого, стихийного использования всех видов ресурсов, в том числе живого труда (трудовых ресурсов).

Государство является бюрократической надстройкой, которой общество делегирует полномочия управлять СЭС для эффективной реализации стратегии (траектории) достижения цели. При этом, так как с одной стороны индивидуальные потребности являются ненасыщаемой функцией, а с другой, часто не совпадают и даже противоречивы, необходима концепция управления, гармонизирующая социальные отношения.

В этих условиях под управлением понимается не принудительное подчинение одних социальных групп другим, а осознанный выбор социальных решений.

Многие исследователи в настоящий момент развивают социально-экономический подход к анализу и учету социальных ресурсов. Появились новые индексы и индикаторы в мировой практике (индекс устойчивого развития, индекс развития человека) [2], которые разносторонне учитывают аспекты функционирования СЭС. Разрабатываются новые научные направления (экономика труда и прочее), где стараются объединить практику социологии и экономики.

Труд с социальной точки зрения – деятельность человека, результатом которой является производство общественно полезных (или, хотя бы, потребляемых обществом) продуктов – материальных (товаров, услуг или средств их производства) или духовных [3]. Благодаря трудовой деятельности общество создает элементы культурного наследия, формирует производственный потенциал, накапливает знания и опыт, а также формирует характер взаимоотношений между человеком и природой. Ни один процесс искусственных целенаправленных систем, будь то общественные, организационные, экономические, технические и прочие системы, не может протекать без непосредственного участия человека, в частности без некой трудовой деятельности.

С экономической точки зрения труд – процесс сознательной, целесообразной деятельности людей, с помощью которой они видоизменяют вещество и силы природы, приспосабливая их для удовлетворения своих потребностей [3].

Как отдельному индивидууму, так и группе в целом свойственны социальные и индивидуально-психологические особенности. К социальным

особенностям могут быть отнесены потребности, мотивы и мотиваторы, ценностные ориентации, цели и ожидания, межличностные отношения, в том числе формальные и неформальные, социально-психологический климат в коллективе, уровень профессиональной подготовленности и т.д.

Предметы труда (природное вещество, на которое человек воздействует в процессе труда) и средства труда (вещь или совокупность вещей, которые служат для реализации воздействия) не функционируют, если они не включены в процесс живого труда. Другими словами, человек, как единственный источник трудовой деятельности, является не просто частью системы производства, накопления, перераспределения, обмена и потребления благ, но ключевой составляющей. Без наличия человека в такой системе сама система теряет смысл. Из этого следует, что на успешное функционирование экономических и производственных систем достаточно сильное воздействие оказывают социальный и психологический аспекты.

В социологи используют пять основных подходов при изучении и объяснении различных фактов [4].

1. *Демографический* – изучение населения: рождаемости, смертности, миграции и связанной с этим деятельности людей. Долгое время основной подход к изучению демографических процессов был количественным, основные принципы которого были сформулированы Томасом Мальтусом еще в XVIII в. Современный взгляд на данные процессы связан более с системным подходом, восприятием демографической структуры общества как целостной мировой системы с уникальными законами и закономерностями. Одним из ярчайших представителей этого направления является С.П. Капица [5].

2. *Психологический* – объясняет поведение человека с точки зрения его восприятия, как собственного, так и окружающих. Изучаются такие понятия как мотивы, мысли, навыки, представления человека о самом себе. Социальные психологи исследуют формирование социальных установок, взаимодействие общества и личности в процессе социализации, формирование и распространение настроений.

3. *Коллективистический подход* применяется, когда изучаются два или более индивида, образующих группу. При помощи данного подхода анализируются конкуренция между политическими партиями, конфликты на расовой основе, соперничество между группами, бюрократические системы. Он также помогает понять, каким образом люди, принадлежащие к одному общественному классу, расе или связанные одинаковым этническим происхождением, возрастом, полом, образуют группы с целью защиты своих интересов (достижения целей). Кроме того, этот подход важен при изучении коллективного поведения, например действий толпы, реакции аудитории, а также таких общественных движений, как борьба за гражданские права, феминизм, массовые акции протеста.

4. *Взаимоотношения* – общественная жизнь рассматривается не через определенных участвующих в ней людей, а через их взаимодействие друг другом, обусловленное их ролями. В большинстве случаях социальный статус (роль) оказывает непосредственное влияние на выбор цели и траектории ее достижения. Более подробно данный аспект будет описан ниже при описании особенностей формирования обобщенных целей.

5. *Культурологический* – он применяется при анализе поведения на основе таких элементов культуры, как общественные морально-этические правила (гласные или негласные) и общественные ценности (вытекающие из религиозных, политических и социальных интересов). Очень важно осознавать, что цели и траектория достижения целей выбираются индивидом из множества не только возможных альтернатив, но и приемлемых по этическим, религиозным и культурным ограничениям. Менталитет, этнические и национальные особенности, привычки, определяют предрасположенность различных социальных групп к выбору схожих целей и траекторий их достижения.

Следует отметить, что при выборе государственной социальной политики (управляющих решений) необходимо учитывать все перечисленные выше подходы к анализу и идентификации социума, его групп и отдельных индивидов, так как полнота исходной информации является необходимым условием эффективности принимаемых организационных решений [6]. Практическая реализация указанного процесса является задачей социального мониторинга как составной части системы общего мониторинга СЭС.

Государство должно обеспечить максимизацию уровня удовлетворения потребностей каждого индивида. Но достичь полного удовлетворения потребностей невозможно по двум причинам: во-первых, государство не может идентифицировать все потребности каждого индивида, а во-вторых, ресурсы государства ограничены. В сложившейся системе социально-экономических отношений целевая задача решается опосредованно путем обмена личного живого труда на универсальный ресурс – деньги, которые затем индивид трансформирует в личные потребности.

При всей кажущейся самостоятельности процесса обмена живого труда на универсальный ресурс, именно государство должно обеспечивать потенциальную возможность каждому индивидууму максимизировать уровень личного потребления. При этом в условиях рыночных отношений госу-

дарство не всегда может непосредственно устанавливать цены на рынке труда. Стоимость живого труда, как и любого другого ресурса, определяется его количеством и качеством. В этих условиях средством достижения глобальной государственной цели является создание потенциальных условий для максимизации личного потенциала индивидов (т.е. максимизация количества и качества труда). При этом количество производимого труда определяется уровнем здоровья индивидуума, а качество – уровнем его интеллектуального развития. Поэтому именно система здравоохранения и система образования являются основными рычагами государственного управления в социально-ориентированном государстве.

2. Формирование бюджета

Как уже отмечалось, государство непосредственно не производит материальных и духовных благ, но располагает консолидированным денежным бюджетом, пропорциональным в общем случае эффективности экономики государства. Так как деньги можно трансформировать в любой ресурс, на основе бюджета реализуются любые управляющие воздействия. При этом при выборе управления, статей и объемов расхода должны выполняться следующие условия:

– должен отсутствовать дефицит бюджетных средств, объем расходов не должен превышать объем исходного бюджета. К сожалению, на данный момент далеко не каждая страна на нашей планете может похвастать выполнением этого условия;

– расширенное воспроизводство бюджета или поступательный рост наполняющих его статей. Это означает, что государство вынуждено в первую очередь финансировать бюджетообразующие процессы с целью увеличить объем поступлений в бюджет и тем самым расширить собственные возможности;

– эффективность использования бюджетных средств. Государство не зарабатывает денежных средств, а лишь заимствует их у общества с целью интеграции и более эффективного целенаправленого использования путем перераспределения. В случае, когда эффективность использования денежных средств государством ниже предполагаемой эффективности использования обществом, целесообразность существования государства как органа управления ставится под сомнение.

3. Подход к решению целевых задач государства

При решении этих формальных задач будем исходить из следующих соображений: объем бюджета пропорционален объему валового внутреннего продукта (ВВП), так как формируется как его процентная норма, следовательно, нужно принимать решения, стимулирующие его рост. В свою очередь, объем ВВП является неубывающей функцией численности активного населения, производительности труда и уровня его потребления, другими словами, чем выше численность населения и разнообразней структура потребления, тем выше ВВП. Например, первые три места по объему ВВП занимают ЕС, США и Китай, первые два известны своим потреблением, превышающим среднемировой уровень в разы, Китай же известен численностью населения.

Следовательно, для выполнения ранее описанных условий государство должно обеспечить:

– естественный прирост населения;

– максимизацию количества живого труда при фиксированном уровне активного населения за счет снижения потерь по болезням, эпидемическим и хроническим заболеваниям, инвалидности и прочее;

– производительность (качество) труда;

– рост личного потребления и разнообразие его структуры.

Существует два основных аспекта, на которые государство способно воздействовать, чтобы решить целевые задачи: здравоохранение (продолжительность жизни, устойчивость к заболеваниям) и образование (воспитание, фактологические знания, квалификационные умения). Здоровье состоит из физического и психологического. К физическому относят:

– здоровье родителей до зачатия и во время беременности – основной путь увеличения численности населения – это естественный прирост, и чем здоровее родители, тем выше вероятность здорового потомства;

– здоровье детей – во время развития очень важно сохранить здоровье молодого поколения по двум причинам, во-первых, именно в этот момент формируется способность человека противостоять болезням и обеспечивается долголетие; во-вторых, чем здоровее индивид, тем больше благ он способен произвести для общества;

– здоровье взрослого населения – это основной производитель общественных благ, чем дольше индивиды сохраняют свое здоровье, тем больше и качественнее производят блага; каждая профессия и род занятий сопряжен с рисками для различных заболеваний, поэтому на государство возлагается груз разработки правовых и нормативных требований к трудовой деятельности.

К психологическому здоровью относят:

– профилактику и лечение душевных расстройств;

– обеспечения уровня удовлетворения и счастья;

– повышения производительности труда, посредством мотивационной стимуляции.

Человек не способен контактировать, производить блага или просто выжить в современном

обществе без определенных знаний и умений. Поэтому государство посредством образовательных систем (дошкольной и школьной) предоставляет индивиду определенный набор знаний и умений, которыми необходимо обладать, дабы стать полноправным и полноценным членом общества. Также такие понятия как воспитание, этические нормы, правила поведения, социальные роли и нормы тоже прививаются в результате образования.

Следует подчеркнуть, что вложения в указанные нематериальные сферы являются косвенными инвестициями в развитие материального производства.

Выводы

Государство как орган управления обществом было создано с целью удовлетворения личных целей индивидов, путем обеспечения роста объемов и разнообразия общественных благ. Чтобы удовлетворить максимально возможное количество личных целей индивидов, государство должно повышать эффективность производства общественных благ, а так как единственным производителем благ является человек, то речь идет о повышении эффективности живого труда. Такие показатели как производительность труда, эффективность, уровень производства являются немаловажными, и их рост достигается за счет повышения квалифицированности индивидов посредствам систем образования.

Вторым принципиально важным направлением является долговечность производителей благ. Причем система здравоохранения, призванная решить эту проблему, должна обеспечить не просто выживание, а сохранить здоровье человека на таком уровне, чтобы он мог максимально долго и эффективно производить общественные блага.

Государство неспособно напрямую обеспечивать удовлетворение каждой личной цели индивида, как и не может определить правомерность предоставления индивиду доступа к общественным благам. Поэтому была разработана схема, по которой человек обменивает свой живой труд на вознаграждение (универсальный ресурс), который впоследствии может обменять на общественное благо и достичь личной цели.

Проведенный в статье системный анализ целей и задач государственного управления позволяет наметить основные направления и принципы формирования расходной части государственного и региональных бюджетов СЭС. Однако конкретизация и детализация намеченных принципов требует разработки решения комплекса оптимизационных задач по распределению ограниченного объема бюджетных ресурсов по критерию максимизации дохода в бюджет на следующем плановом интервале. В свою очередь это требует синтеза социально-экономических моделей, устанавливающих связь величины бюджетных расходов с ожидаемым бюджетообразующим эффектом. Государственная политика должна стать принятием строго аргументированных, научно обоснованных социально-экономических решений.

Список литературы: 1. *Коптюг*, В.А. Конференция ООН по окружающей среде и развитию (Рио-де-Жанейро, июнь 1992 года). Информационный обзор [Текст] / В.А. Коптюг. – Новосибирск: СО РАН, 1992. – 63 с. **2.** *Згуровский*, М.З. Роль инженерной науки и практики в устойчивом развитии общества [Текст] / М.З. Згуровский, Г.А. Статюха // Системні дослідження та інформаційні технології. – 2007. – №1. – С. 19-38. **3.** Экономика труда: учебник. 2-е изд., перераб. И доп. / под ред. проф. Ю.П. Кокина, проф. П.Э. Шлендера. – М.: Магистр, 2010. – 686 с. **4.** *Смелзер*, Н. Социология [Текст] / Н. Смелзер: пер. с англ. – М.: Феникс, 1994. – 688 с. **5.** *Капица*, С.П. Парадоксы роста. Законы развития человечества [Текст] / С.П. Капица. – М.: АНФ. – 2010. – 192 с. **6.** *Глушков*, В.М. Введение в АСУ [Текст]: изд.2-е, исправленное и дополненное / В.М. Глушков. – К.: Техніка, 1974, – 320 с.

Поступила в редколлегию 29.06.2011

УДК 519.81

Роль, завдання та методи державного управління при реалізації концепції сталого розвитку / Е.Г. Петров, Є.В. Губаренко // Біоніка інтелекту: наук.-техн. журнал. – 2011. – № 3 (77). – С. 60-64.

Держава, як орган управління суспільством, має створювати умови для максимізації рівня особистого споживання індивідів. Для цього необхідно відмовитися від сприйняття населення як пасивної частини економічних процесів і сконцентрувати увагу на соціальних аспектах суспільства. Держава покликана підтримувати суспільні процеси, перерозподіляючи і стимулюючи виробництво суспільних благ. За допомогою формування кількісних і якісних характеристик індивіда, як джерела живої праці, держава має можливість підвищувати його потенційні можливості задоволення рівня особистого споживання. Інструментами такого впливу є системи охорони здоров'я і освіти.

Бібліогр.: 6 найм.

UDK 519.81

The role, tasks and methods of state management in implementing the concept of sustainable development / E.G. Petrov, E.V. Gubarenko // Bionics of Intelligense: Sci. Mag. – 2011. – № 3 (77). –P. 60-64.

The state, as the management body of the company, should create conditions to maximize the level of personal consumption of the individual. For this it is necessary to abandon the prevailing perceptions of the population as a passive part of the economic processes and focus on the social aspects of society. The state should support social processes, redistributing and stimulating the production of public goods. Through the formation of a quantitative and qualitative characteristics of the individual as the source of living labour, the state has the possibility to increase the potential of the satisfaction level of personal consumption. Instruments of such influence are the health and education systems.

Ref.: 6 items.

УДК 519.72;681.326

С. В Кузьменко
ХНУВД, м. Харьков, Украина, unlaudt@kture.kharkov.ua

ТЕОРЕТИЧЕСКИЕ ОСНОВЫ ИДЕОГРАФИЧЕСКОГО МЕТОДА

Проведен анализ современного состояния использования идеографического подхода для решения задач моделирования информационных систем. Приведено математическое описание теоретических основ идеографического метода, а также структура информационной технологии идеографического описания систем.

ИНФОРМАЦИОННЫЕ СИСТЕМЫ, НОМОТЕТИЧЕСКИЙ ПОДХОД, ИДЕОГРАФИЧЕСКИЙ ПОДХОД, ТЕОРЕТИЧЕСКИЕ ОСНОВЫ, ИДЕОГРАФИЧЕСКОЕ МОДЕЛИРОВАНИЕ, ИДЕОГРАММА, ИНФОРМАЦИОННАЯ ТЕХНОЛОГИЯ ИДЕОГРАФИЧЕСКОГО ОПИСАНИЯ СИСТЕМ

Введение

Развитие возможностей средств вычислительной техники, появление средств визуализации различных процессов позволяют перейти от номотетического способа представления информационных систем к их идеографическому представлению. Если номотетический подход – это способ познания, целью которого является установление общего, имеющего форму закона, то идеографический подход – это способ познания, целью которого является изображение объекта как единого уникального целого. Главной особенностью идеографического метода является постижение индивидуального в его однократности, уникальности и неповторимости.

Номотетический и идеографический подходы различаются по нескольким основаниям. Во-первых, по-разному понимается объект измерения. Если в рамках номотетического подхода присутствует понятие объекта как набора свойств, то идеографический подход представляет объект как целостную систему. Во-вторых, для каждого из подходов характерна разная направленность измерений: выявление и измерение общих для всех объектов их свойств – для номотетического подхода и распознание индивидуальных особенностей объекта – для идеографического подхода. И, в-третьих, характер методов измерения в каждом из подходов различен по процессу и содержанию: стандартизованные методы измерения с одной стороны, и проективные методики и качественные техники – с другой.

Индивидуализация, переход к конкретным особенностям каждой из информационных систем требуют их представления в терминах идеографического подхода в виде индивидуальных образов. Однако решение подобной задачи невозможно без разработки некоторой индивидуальной модели, которая позволит представить как общие, так и личностные свойства информационных систем. В связи с тем, что человеческое мышление оперирует образами, а не формулами и цифрами, при разработке индивидуальной модели необходим переход к широкому использованию идеографического метода в сочетании с номотетическим.

За последние 10-15 лет появился ряд работ, в которых сделана попытка разработки такой модели. В работах [1, 2] предложен идеографический метод представления систем имитационного моделирования, что позволило разработать ряд программных средств оперативного построения имитационных программ [3, 4]. В работах [5, 6] эти результаты были развиты для 2-уровневых имитационных моделей информационно-технологических систем, взаимосвязи между уровнями которых осуществляются на информационном уровне. В настоящее время получены результаты [7], которые позволяют распространить их на более широкий класс систем.

1. Постановка задачи

Проведенный анализ позволяет поставить следующую задачу: разработать теоретические основы идеографического метода и информационную технологию идеографического описания систем, которые, с одной стороны, позволят представлять различные системы с учетом их индивидуальности, а с другой – объединить идеографическое описание с их номотетическим представлением.

2. Теоретические основы метода

Основным методологическим предположением метода является предположение о том, что любая система может быть с определенной степенью точности представлена ее идеографической моделью.

Определение 1. Любая система S может быть представлена ее идеографической моделью Mg, т.е. $Mg \equiv S$, что определяет ее индивидуальность в однократности, уникальности и неповторимости.

Любая система S имеет определенную структуру и описывается своими неповторимыми характеристиками. Исходя из этого, идеографическая модель Mg системы S строится на основе идеографических элементов Ig_i.

Определение 2. Система S, которая моделируется, состоит из множества объектов $Ob_i, (i = \overline{1,n})$.

При этом во множестве $Ob = \{Ob_i\}$ нет равнозначных объектов:

$$S = \cup Ob_i, \quad (i = \overline{1,n});$$
$$\forall i \in n \exists! Ob_i (Y_i = F(X_i)).$$

Определение 3. Идеографическая модель Mg системы S является объединением идеологических элементов Ig_i:

$$Mg = \bigcup_{i=1}^{n} Ig_i, (i = \overline{1,n}).$$

Определение 4. Идеографический элемент Ig_i описывает только один определенный объект Ob_i реальной системы, т.е.

$$\forall Ig_i \in Ig \vee \forall Ob_i \in Ob \exists! Ig_i \equiv Ob_i.$$

Определение 5. Идеографические элементы Ig_i подразделяются на два вида: структурный идеографический элемент Ig_i^{c}; функциональный идеографический элемент Ig_i^{Φ}.

Определение 6. Структурный идеографический элемент Ig_i^{c} идеографической модели Mg описывает только один определённый структурный объект Ob_i^{c} обработки материального потока моделируемой системы S.

Определение 7. Функциональный идеографический элемент Ig_i^{Φ} идеографической модели Mg описывает методы обработки данных и их визуализации в моделируемой системе S.

Определение 8. Структурный Ig_i^{c} и функциональный Ig_i^{Φ} идеографические элементы имеют трёхуровневую структуру, которая состоит из логического (l), концептуального (k) и физического (f) уровней (рис. 1).

Логический уровень Ig_i – это идеографическое представление объекта Ob_i системы S, т.е. графическое, видеографическое или иное изображение, которое является легко узнаваемым. Физический уровень Ig_i – это физический элемент объекта Ob_i системы S, который может быть представлен в виде устройства, программы, схемы или другого элемента. Концептуальный уровень Ig_i – это уровень преоб-разования идеографического описания Ig_i в его физическое описание. Этот уровень может быть представлен математическими моделями, функциональными зависимостями и другим.

Определение 9. Функция $F^{\Phi}: X^{\Phi} \to Y^{\Phi}$ или $Y^{\Phi} = F^{\Phi}(X^{\Phi})$ есть отображение множества входных значений X^{Φ} во множество элементов выходных результатов Y^{Φ} каждого из функциональных идео-графических элементов Ig_i^{Φ} идеографической модели Mg на каждом из уровней отображения: $l^{\Phi}, k^{\Phi}, f^{\Phi}$. Пространство входных переменных определяется областями определения A^{Φ} переменных $x_1^{\Phi}, x_2^{\Phi}, ..., x_{A^{\Phi}}^{\Phi}$, а элементом входных данных являются точки $x_1^{\Phi}, x_2^{\Phi}, ..., x_{A^{\Phi}}^{\Phi}$. Пространство выходных переменных определяется областями значений B^{Φ} переменных $y_1^{\Phi}, y_2^{\Phi}, ..., y_{B^{\Phi}}^{\Phi}$, а элементом выходных результатов являются точки $y_1^{\Phi}, y_2^{\Phi}, ..., y_{B^{\Phi}}^{\Phi}$.

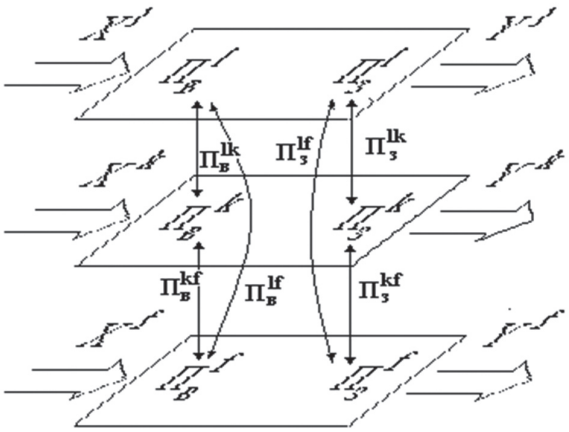


Рис. 1. Структурная схема идеографического элемента Ig_i

Определение 10. Функция $F^c: X^c \to Y^c$ или $Y^c = F^c(X^c)$ есть отображение множества входных значений X^c во множество элементов выходных результатов Y^c каждого из структурных идеографических элементов Ig_i^c идеографической модели Mg на каждом из уровней отображения: l^c, k^c, f^c. Пространство входных переменных определяется областями определения A^c переменных $x_1^c, x_2^c, ..., x_{A^c}^c$, а элементом входных данных являются точки $x_1^c, x_2^c, ..., x_{A^c}^c$. Пространство выходных переменных определяется областями значений B^c переменных $y_1^c, y_2^c, ..., y_{B^c}^c$, а элементом выходных результатов являются точки $y_1^c, y_2^c, ..., y_{B^c}^c$.

Аксиома 1. Функция $F^j: X^j \to Y^j$ или $Y^j = F^j(X^j)$, где $j = (l, k, f)$ для каждого из уровней идеографического элемента Ig_i (структурного или функционального) идеографической модели Mg, имеет свои, свойственные только ей области определения входных A^l, A^k, A^f и выходных переменных B^l, B^k, B^f, а также соответствующие элементы входных x_i^j и выходных y_i^j результатов.

Теорема 1. Идеографическая модель Mg системы S имеет трехуровневую структуру, состоящую из логического (l), концептуального (k) и физического (f) уровней.

Доказательство. Исходя из *определения 8,* идеографический элемент Ig_i имеет трехуровневую структуру, а исходя из *определения 3* идеографическая модель Mg является объединением идеографических элементов, поэтому идеографическая модель Mg наследует структуру идеографического элемента и имеет трехуровневую структуру.

Теорема 2. Идеографический элемент Ig_i (структурный или функциональный) на каждом уровне описания состоит из: плана отображения (Π_0^j) – графического отображения функции преобразования $Y^j = F^j(X^j)$ во множество переменных P^j, которые состоят из переменных входа, выхода и внутренних переменных $P^j = \{x^j, y^j, v^j\}$;

плана содержания $\left(\Pi_c^j\right)$, математического выражения, которое задает преобразование множества переменных $P^j = \left\{x^j, y^j, v^j\right\}$.

Доказательство. На основании *определений 4, 9 и 10* идеографический элемент Ig_i имеет, во-первых, многоуровневую структуру, а во-вторых, описывается функцией преобразования $Y = F(X)$ множества входных элементов данных X во множество выходных результатов Y, что, в общем случае, является математическим выражением, а в-третьих, свое отображение в соответствии с восприятием различными категориями пользователей.

Теорема 3. Между планом отображения Π_o и планом содержания Π_c уровней описания идеогра-фического элемента Ig_i (структурного или функцио-нального) есть взаимнооднозначное соответствие, которое описывается с помощью плана отображения (Π_o^{lk}, Π_o^{lf}, Π_o^{kf}) и плана содержания (Π_c^{lk}, Π_c^{lf}, Π_c^{kf}) уровней.

Доказательство. Исходя из того, что каждый идеографический элемент Ig_i описывает один определенный объект моделируемой системы (*определение 4*), а также из того, что каждый уровень описания имеет свой план отображения $\left(\Pi_o^l, \Pi_o^k, \Pi_o^f\right)$ и содержания $(\Pi_c^l, \Pi_c^k, \Pi_c^f)$ – *теорема 2*, выходит, что между планами отображения $\left(\Pi_o\right)$ и планами содержания $\left(\Pi_c\right)$ уровней можно найти математическое выражение – план отображения уровней $(\Pi_o^{lk}, \Pi_o^{lf}, \Pi_o^{kf})$ и план содержания уровней $(\Pi_c^{lk}, \Pi_c^{lf}, \Pi_c^{kf})$:

$$\Pi_c^l = \Pi_c^{lk}\left(\Pi_c^k\right);$$

$$\Pi_c^l = \Pi_c^{lf}\left(\Pi_c^f\right);$$

$$\Pi_c^k = \Pi_c^{kf}(\Pi_c^f).$$

При этом считается, что расположение кодов уровней в верхнем индексе определяет порядок отображения: *lk* – логического в концептуальный, *lf* – логического в физический, *kf* – концептуального в физический.

Теорема 4. Выходные результаты y_i^j идеографиического элемента Ig_i (структурного или функционального) не могут быть входными x_i^j данными того же элемента.

Доказательство. Если в идеографической модели *Mg* существует $y = f(y, x)$, то это означает, что объект Ob_i моделируется системой, передает выходные результаты на свой вход, что невозможно в классе последовательно связанных объектов системы.

Аксиома 2. Количество и степень отношения между входными и выходными переменными идеографического элемента Ig_i (структурного или функционального) определяет аксиома вычисления $e(Ig_i)$.

Аксиома 3. Количество входов идеографического элемента Ig_i (структурного или функционального) определяется порогом вычисления $p(Ig_i)$.

Аксиома 4. Каждый функциональный идеографический элемент Ig_i^{Φ} имеет связь по информации со структурным идеографического элемента Ig_i^{c}, что позволяет персонифицировать данные каждого функционального идеографического элемента Ig_i^{Φ} с конкретным структурным идеографическим элементом Ig_i^{c}.

Аксиома 5. Структурный идеографический элемент Ig_i^{A} имеет несколько планов отображения (Π_o^j), которые используются для его отображения на υ–х уровнях иерархической структуры идеографической модели *Mg*.

Теорема 5. Количество соединений, которые объединяют два смежных идеографических элемента $\left(Ig_i, Ig_{i+1}\right)$, не должно превышать их пороги вычисления: $p = \min\left(Ig_i, Ig_{i+1}\right)$.

Доказательство. Превышение порогов вычисления при объединении двух смежных идеографических элемента $\left(Ig_i, Ig_{i+1}\right)$ привело бы к нарушению *аксиомы 3*, то есть к тому, что пороги вычисления идеографических элементов установлены неверно.

Теорема 6. Выходные результаты идеографиического элемента Ig_i (структурного или функционального) являются входными данными другого идеографического элемента Ig_{i+1} только в том случае, если они имеют совместную аксиому вычисления $e\left(Ig_i, Ig_{i+1}\right)$.

Доказательство. На основании *аксиомы 4* два идеографических элемента $\left(Ig_i, Ig_{i+1}\right)$ могут объединиться в ансамбль при существовании между их переменными совместной аксиомы вычисления $e\left(Ig_i, Ig_{i+1}\right)$. Поэтому, если такая аксиома не существует, то выходные результаты $Y\left(Ig_i\right)$ i-го идеографического элемента не могут быть входными переменными $X\left(Ig_{i+1}\right)$ $(i+1)$-го идеографического элемента.

Теорема 7. Комплексирование идеографических элементов Ig_i (структурных и функциональных) в идеографическую модель *Mg* выполняется на основании их планов отображения $\Pi_o = \left\{\Pi_o^l, \Pi_o^k, \Pi_o^f\right\}$, а проверка правильности комплексирования – на основании их планов содержания $\Pi_c = \left\{\Pi_c^l, \Pi_c^k, \Pi_c^f\right\}$.

Доказательство. Согласно *теореме 2* план отображения $\left(\Pi_o^j\right)$ идеографического элемента – это графическое отображения функции преобразования $Y^j = F^j\left(X^j\right)$, то есть некоторое визуальное представление объекта Ob_i моделируемой системы S, которое используется для построения визуального представления системы S, а план содержания $\left(\Pi_c^j\right)$ – это математическое выражение, которое описывает функцию преобразования $Y^j = F^j\left(X^j\right)$. Следовательно, построение визуального представления моделируемой системы S возможно только

на основании визуальных представлений объектов Ob_i этой системы, то есть их планов отображения $\Pi_{\text{o}} = \left\{\Pi_{\text{o}}^{l}, \Pi_{\text{o}}^{k}, \Pi_{\text{o}}^{f}\right\}$, а проверка правильности комплексирования – на основании совместимости их функций преобразования $F^j : X^j \to Y^j$, то есть планов содержания $\Pi_{\text{c}} = \left\{\Pi_{\text{c}}^{l}, \Pi_{\text{c}}^{k}, \Pi_{\text{c}}^{f}\right\}$

Теорема 8. Пороги вычисления уровней идеографического элемента Ig_i (структурного или функционального) одинаковы: $p^l = p^k = p^f$.

Доказательство. Так как каждый уровень идеографического элемента Ig_i (структурного или функционального) представляет одну и ту же функции преобразования $Y^j = F^j\left(X^j\right)$, но в разном виде (логическом, концептуальном и физическом), пороги вычисления уровней идеографического элемента Ig_i (структурного или функционального) не могут быть не одинаковы.

Теорема 9. Между входными переменными уровней идеографического элемента Ig_i (структурного или функционального) $X = \left\{X^l, X^k, X^f\right\}$ и выходными результатами $Y = \left\{Y^l, Y^k, Y^f\right\}$ существует взаимно однозначное соответствие.

Доказательство. Так как каждый уровень идеографического элемента Ig_i (структурного или функционального) представляет одну и ту же функцию преобразования $Y^j = F^j\left(X^j\right)$, но в разном виде (логическом, концептуальном и физическом), между входными переменными уровней идеографического элемента Ig_i (структурного или функционального) $X = \left\{X^l, X^k, X^f\right\}$ и выходными результатами $Y = \left\{Y^l, Y^k, Y^f\right\}$ не может не существовать взаимно однозначное соответствие.

Теорема 10. Множество идеографических элементов Ig_i (структурных и функциональных), которые описывают идеографическую модель Mg системы S, не содержит двух идеографических элементов, которые имеют одинаковые планы отображения $\Pi_{\text{o}} = \left\{\Pi_{\text{o}}^{l}, \Pi_{\text{o}}^{k}, \Pi_{\text{o}}^{f}\right\}$, планы содержания $\Pi_{\text{c}} = \left\{\Pi_{\text{c}}^{l}, \Pi_{\text{c}}^{k}, \Pi_{\text{c}}^{f}\right\}$ и множества переменных X, Y, то есть:

$$\forall i \in n \; \exists! \; Ig_i,$$

$$\Pi_{\text{o}}\left(Ig_i\right) \neq \Pi_{\text{o}}\left(Ig_{(-i)}\right) \vee \Pi_{\text{c}}\left(Ig_i\right) \neq \Pi_{\text{c}}\left(Ig_{(-i)}\right) \vee$$

$$\vee A\left(Ig_i\right) \neq A\left(Ig_{(-i)}\right) \vee B\left(Ig_i\right) \neq B\left(Ig_{(-i)}\right).$$

Доказательство. Согласно *определению 4* идеографический элемент Ig_i описывает только один определенный объект Ob_i реальной системы, то есть $Ig_i \equiv Ob_i$. Поэтому если предположить, что имеется больше одного идеографического элемента Ig_i, которые имеют одинаковые планы отображения $\Pi_{\text{o}} = \left\{\Pi_{\text{o}}^{l}, \Pi_{\text{o}}^{k}, \Pi_{\text{o}}^{f}\right\}$, планы содержания $\Pi_{\text{c}} = \left\{\Pi_{\text{c}}^{l}, \Pi_{\text{c}}^{k}, \Pi_{\text{c}}^{f}\right\}$ и множества переменных X, Y, то это означало бы, что нарушено *определения 4* и неправильно сформирован состав идеографических элементов Ig_i идеографической модели Mg системы S.

3. Информационная технология идеографического описания систем

Представленные в работе теоретические основы идеографического метода позволяют реализовать информационную технологию описания широкого класса систем на основе идеограмм (рис. 2). Основными объектами этой технологии являются:

– пользователь (*a*) идеографического описания системы;

– проектировщик (*в*) идеографических элементов описания системы;

– идеографическая модель системы (*c*);

– инструментальное средство (*d*), используемое при создании идеографических элементов;

– реестр идеографических элементов (*e*).

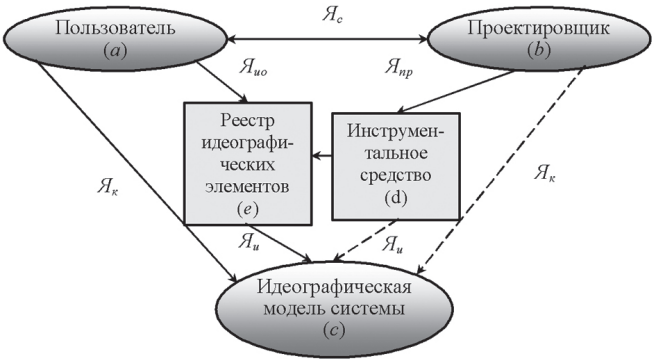


Рис. 2 . Контекстная схема информационной технологии идеографического моделирования систем

Информационная технология реализуется следующим образом: пользователь (*a*) и проектировщик (*в*), используя язык спецификаций $Я_c$, определяют требования к идеографическому моделированию (описанию) той предметной области (системы), в которой работает пользователь. Проектировщик (*в*), используя язык проектирования $Я_{пр}$ и инструментальное средство (*d*), выполняет создание идеографических элементов описания системы и размещает их в реестре (*e*). Проектировщик (*в*) при отладке идеографических элементов описания системы использует язык $Я_и$ создания идеографической модели системы (*c*) и язык команд $Я_к$ для управления процессом использования идеографической модели системы (*c*).

Пользователь (*a*), используя язык идеографического описания $Я_{ио}$, имеет возможность анализировать состав реестра идеографических элементов (*e*) и, используя язык описания идеографической модели $Я_и$ системы (*c*), создает идеографическую модель (*c*), а используя язык команд $Я_к$ управляет процессом использования идеографической модели системы (*c*).

Выводы

Разработанные теоретические основы идеографического описания систем позволяют оперировать с системой на основе образов (плана отобра-

жения), а не математических зависимостей, схем, диаграмм и другого (плана содержания), что более естественно для пользователя. Однако реализация такого взаимодействия требует разработки информационной технологии идеографического описания систем (рис. 2) для конкретной предметной области, что является направлением дальнейшего развития работ. При разработке такой информационной технологии необходимо:

– определиться с объектами информационной технологии (рис. 2);

– разработать соответствующие языки информационной технологии с учетом особенностей выбранной предметной области;

– апробировать адаптивную информационную технологию для решения задач идеографического моделирования элементов и объектов системы выбранной предметной области;

– провести анализ использования адаптивной информационной технологи для решения задач идеографического моделирования элементов и объектов систем различного назначения.

Список литературы: 1. *Кузьменко, В.М.* Спеціальні мови програмування. Програмні та інструментальні засоби моделювання складних систем: [навч. посібник] [Текст] / В.М. Кузьменко – Х.: ХНУРЕ, 2000. – 324 с. **2.** *Кузьменко, В.М.* Інформаційна технологія імітаційного моделювання на основі ідеографічного підходу [Текст] / В.М. Кузьменко // Надійність інструменту та оптимізація технічних систем. – 1999. – Вип. 9. – С. 64-70. **3.** *Кузьменко, В.М.* Метод имитационного моделирования сложных систем на основе идеографического подхода [Текст] / В.М. Кузьменко, Л.Ф.Ненько // Проблемы бионики. – 2000. – № 52. – С. 67-72. **4.** *Ненько, Л.Ф.* Методы и средства оперативного управления интегрированными организационно-технологическими системами: автореф. дис. ... канд. техн. наук: спец. 05.13.06 «АСУ и прогрессивные информационные технологии» / Л.Ф.Ненько. – Х., 2001. – 21 с. **5.** *Шульга, Ю.В.* Модели автоматизированного проектирования технологических систем обработки отправлений: автореф. дис. ... канд. техн. наук: спец. 05.13.12 «Системы автоматизации проектирования» / Ю.В. Шульга. – Х., 2002. – 21 с. **6.** *Кузьменко, В.М.* Инструментальное средство имитационного моделирования информационно–технологических систем [Текст] / В.М. Кузьменко, Ю.В. Шульга // АСУ и приборы автоматики. – 2002. – № 120. – С. 51 – 55. **7.** *Кузьменко, С.В.* Модели и информационная технология контроля в распределенных организационно-технологических системах: автореф. дис. ... канд. техн. наук: спец. 05.13.06 «Информационные технологии» / С.В. Кузьменко. – Х., 2008. – 21 с.

Поступила в редколлегию 05.07.2011

УДК 519.72;681.326

Теоретичні основи ідеографічного методу / С.В. Кузьменко // Біоніка інтелекту: наук.-техн. журнал. – 2011. – № 3 (77). – С. 65-69.

Розглядається сучасний стан моделювання інформаційних системах з використанням засобів ідеографічного методу. Наведено нові результати щодо теоретичного обґрунтування ідеографічного моделювання інформаційних систем. Як практичний результат отриманих теоретичних досліджень приведена інформаційна технологія інформаційного опису систем з використанням ідеографічного методу.

Іл. 2. Бібліогр.: 7 найм.

UDS 519.72;681.326

Theoretical bases of an ideographic method / S.V. Kuzmenko // Bionics of Intelligense: Sci. Mag. – 2011. – № 3 (77). –P. 65-69.

The current state of modeling information systems with use of means of an ideographic method is considered. New results concerning a theoretical substantiation of ideographic modeling of information systems are resulted. As practical result of the received theoretical researches the information technology of the information description of systems with use of an ideographic method is resulted.

Fig. 2. Ref.: 7 items.

УДК 004.42:519.688:519.81

Н.И. Калита, А.В. Гурина
ХНУРЭ, г. Харьков, Украина, kalita.nik@gmail.com

СИНТЕЗ МОДЕЛИ МНОГОФАКТОРНОГО ОЦЕНИВАНИЯ ЭФФЕКТИВНОСТИ КОМАНДЫ ПРОЕКТА

Предложена математическая модель определения оптимального состава проектной команды и оценки ее эффективности. Критерием оптимальности является максимальная полезность (привлекательность) вариантов команды с учетом ограничений на личностные и функциональные требования к проектным ролям. Приведен пример решения задачи.

КОМАНДА ПРОЕКТА, ФУНКЦИОНАЛЬНЫЕ РОЛИ, ЛИЧНОСТНЫЕ ЧЕРТЫ, ЭФФЕКТИВНОСТЬ КОМАНДЫ ПРОЕКТА, ТЕОРИЯ ПОЛЕЗНОСТИ, МОДЕЛЬ ОЦЕНКИ ЭФФЕКТИВНОСТИ КОМАНДЫ

Введение

Для успешной реализации проекта в какой-либо сфере деятельности первостепенное значение имеет эффективная команда проекта, которая представляет собой специфическую организационную структуру, формируемую на время жизненного цикла проекта [1]. Команда проекта является основным элементом его структуры, так как именно она обеспечивает реализацию замысла проекта. Под командой понимается объединение людей, осуществляющих совместную деятельность и обладающих общими интересами, которое способно достигать цели автономно и согласованно при минимальных управляющих воздействиях [2]. Для команды проекта необходимо наличие у ее членов комбинации взаимодополняющих навыков, которые составляют три категории:

– функциональные (или профессиональные);

– навыки по решению проблем и принятию решений;

– навыки межличностного общения.

Степень выраженности указанных качеств предопределяет те или иные роли членов команды в проекте [3[. Каждый из них выполняет роли двух типов: функциональную (проектную) - как определенный набор функций и полномочий в проекте, и командную, в основе которой лежат личностные особенности.

Основными характеристиками команды являются состав, структура, групповые процессы. Состав – совокупность характеристик членов команды, важных для анализа ее как единого целого (например, численность, возрастной, половой состав). Структура рассматривается с точки зрения функций, выполняемых отдельными членами команды, а также с точки зрения межличностных отношений в ней. К групповым процессам относятся процессы развития, сплочения, группового давления, выработки решений.

Команды могут быть однородные и неоднородные. В однородной команде члены команды выполняют однотипные функции, и при формировании такой команды основной задачей является распределение объемов работ между исполнителями. В неоднородной команде ее члены выполняют различные функции, причем каждый член команды в общем случае характеризуется определенными эффективностями реализации тех или иных функций. Поэтому при формировании неоднородной команды основной задачей является распределение ролей и видов деятельности между исполнителями, а потом уже распределяются объемы работ [2]. Проектные команды относятся к неоднородным командам, где члены команды выполняют различные роли.

Эффективную команду можно охарактеризовать общепринятыми критериями эффективности любой организационной структуры, однако специфические черты, присущие только команде, обусловливают эффективность с позиций профессиональной деятельности по проекту и организационно-психологического климата деятельности.

Проблема формирования команды сначала была предметом исследования в психологии, социологии и менеджменте [4-7], однако с развитием и внедрением информационных технологий возникла необходимость в использовании не только неформальных методов для исследования и формирования команды, но и основанных на математических моделях и методах.

Целью статьи является разработка математической модели оценки эффективности команды проекта, которая учитывает наличие требуемых профессиональных и личностных качеств у претендентов в команду проекта.

1. Состояние проблемы и постановка задачи

В настоящее время при формировании команд используются методы, основанные [2, 4, 6, 7]: 1) на исследовании межличностных отношений (социометрия, деловые игры); 2) на математических моделях различных классов. Первая группа методов позволяет количественно оценить такие показатели вариантов команды как взаимные

симпатии и антипатии, групповой статус, ролевой состав и на основе полученных показателей выработать рекомендации по персональному составу команды. При этом, однако, не учитывается распределение функциональных ролей между претендентами. Анализ второй группы методов показал, что с помощью моделирования решаются задачи как на этапе формирования команды, так и на этапе ее работы над проектом. Используются задачи о назначении, рефлексивные и теоретико-игровые модели, модель Маршака-Раднера, репутации и норм деятельности. Для формирования неоднородных команд вначале предлагается решить задачу распределения объемов работ, а затем распределить между претендентами функции. В [8] для формирования эффективной команды учитывается прошлый опыт работы над проектом. Несмотря на разнообразие моделей и методов формирования команды, нерешенной остается задача комплексного учета ролевых особенностей проектной команды, личностных и профессиональных качеств претендентов в команду, которые обеспечивают ее наибольшую эффективность.

Рассмотрим задачу формирования неоднородной команды проекта в такой постановке. Для выполнения проекта необходима команда, в которой членам команды определено множество различных функциональных ролей $Y = \{y_j\}$, $j = \overline{1,m}$. Количественный состав команды определяется объемом работ, предусмотренным проектом. Будем полагать, что объемы работ распределены между функциональными ролями. Ролевой состав команды и количество исполнителей каждой роли определяется менеджером проекта для каждого конкретного проекта. Например, IT-проект предполагает роли: руководитель разработки, разработчик, администратор ИС, тестировщик, менеджер по качеству, системный аналитик, технический писатель. Одна роль может исполняться несколькими специалистами. Но, в свою очередь, один и тот же специалист может совмещать различные роли.

Известны также множества претендентов $X_j = \{x_i\}$, $i = \overline{1,n}$ на каждую *j*-ю роль (должность). При этом некоторые элементы x_i могут быть включены в несколько множеств претендентов (претендуя на различные роли в команде проекта). Каждый претендент x_i из множеств X_j описывается множеством профессиональных $K(x_i) = \{k_u(x_i)\}$, $u = \overline{1,U}$ и личностных $L(x_i) = \{l_v(x_i)\}$, $v = \overline{1,V}$ качеств (характеристик). Для определенности будем полагать, что значения $k_u(x_i)$, $u = \overline{1,U}$ и $l_v(x_i)$, $v = \overline{1,V}$ известны и заданы в виде количественных оценок. Эти значения могут быть получены специалистами по персоналу методами, принятыми в данной предметной области, например, в результате обработки психологических тестов, тестов на профессиональные знания и навыки и т.п.

Каждая функциональная роль y_j характеризуется множеством профессиональных $K^*(y_i) = \{k_u^*(y_i)\}$ и личностных $L^*(y_i) = \{l_v^*(y_i)\}$ требований, которым должны удовлетворять претенденты x_i, и в соответствии с которыми осуществляется выбор оптимальных исполнителей на роли y_j, $j = \overline{1,m}$. Профессиональные требования описывают уровень знаний, умений и навыков для функциональной роли. Личностные качества включают: степень ответственности, способность самостоятельно принимать решения, умение работать в условиях плотного графика, системное мышление, способность работать в условиях рисков и неопределенностей, стратегическое мышление, лидерские качества, организационные и мотивационные навыки, дисциплинированность и организованность, навыки работы в команде, аналитический подход, способность к эффективному общению (устному и письменному), понимание технологий и трендов (тенденций), креативность. Для различных командных ролей (например, лидер, исполнитель, контролер) требования к личностным качествам различны.

Под оптимальным исполнителем x_{ij}^* роли y_j будем понимать такого претендента x_i из множества X_j, который имеет максимальную привлекательность (полезность) $P_j(x_i)$ [9]:

$$x_{ij}^* = \arg\max_{x_i \in X_j} P_j(x_i).$$

Необходимо разработать математическую модель оценки эффективности команды проекта, которая формируется из множеств претендентов X_j на роли y_j, $j = \overline{1,m}$.

2. Математическая модель

Для синтеза модели оценивания эффективности команды проекта используем модель многофакторного оценивания [9, 10]. Привлекательность $P_j(x_i)$ каждого претендента x_i на *j*-ю роль определим как аддитивную функцию полезности, которая учитывает наличие и степень развития его профессиональных $K(x_i)$ и личностных $L(x_i)$ характеристик, требуемых для рассматриваемой должности y_j. Если претендент x_i на роль y_j обладает множеством характеристик $K(x_i) = \{k_u(x_i)\}$ и $L(x_i) = \{l_v(x_i)\}$ большим, чем требуется в $K^*(y_i) = \{k_u^*(y_i)\}$ и в $L^*(y_i) = \{l_v^*(y_i)\}$, он участвует в отборе, если же у него отсутствует хотя бы одна характеристика, он исключается из претендентов на исполнение данной роли.

Для каждой роли y_j, $j = \overline{1,m}$ при формировании команды на выполнение конкретного проекта значения критериев из требований $K^*(y_i)$ и $L^*(y_i)$ известны и могут быть заданы количественными оценками.

Для различных функциональных ролей y_j профессиональные и личностные качества претенден-

тов имеют различный вес, поэтому введем для них весовые коэффициенты a_1 и a_2 соответственно. Кроме того, и профессиональные качества $k_u(x_i)$, $u=\overline{1,U}$ и личностные $l_v(x_i)$, $v=\overline{1,V}$ при оценке претендентов в различных проектах могут иметь разную важность. Следовательно, этот факт также необходимо учесть, используя соответствующие весовые коэффициенты важности q_u и g_v. Тогда привлекательность i-го претендента на j-ю функциональную роль определяется как

$$P_j(x_i)=a_1\sum_{u=1}^{U}q_u k_u(x_i)+a_2\sum_{v=1}^{V}g_v l_v(x_i), \qquad (1)$$

и многофакторная модель оценивания эффективности команды проекта имеет вид:

$$P(x)=\sum_{j=1}^{m}P_j(x_i)\to\max_{x_i\in X_j,\ j}$$

при ограничениях

$$k_u(x_i)\ge k_u(y_j),\ j=\overline{1,m},$$

$$l_v(x_i)\ge l_v(y_j),\ j=\overline{1,m}.$$

Функция полезности вида (1) требует, чтобы значения частных критериев $k_u(x_i)$ и $l_v(x_i)$ имели одинаковый интервал изменения, были безразмерны, и большее значение было бы наилучшим. Поэтому в общем случае значения указанных частных критериев в приведенной модели пронормированы по формуле вида [9]:

$$p[k_i(x)]=\left(\frac{k_i(x)-k_{i\text{нх}}}{k_{i\text{нл}}-k_{i\text{нх}}}\right)^{\alpha_i},$$

где $k_{i\text{нх}}$, $k_{i\text{нл}}$ – соответственно наихудшее и наилучшее значения частного критерия $k_u(x_i)$, $u=\overline{1,U}$, и $l_v(x_i)$, $v=\overline{1,V}$ на всем множестве претендентов X_j; α_i - показатель нелинейности; $p[k_i(x)]$ имеет смысл полезности частного критерия. Также весовые коэффициенты q_u и g_v удовлетворяют требованиям $\sum_{u=1}^{U}q_u=1$, $0\le q_u\le 1$, $\sum_{v=1}^{V}g_v=1$, $0\le g_v\le 1$, и их значения могут быть определены различными методами [9, 10].

3. Вычислительные эксперименты

Предложенная математическая модель апробирована с помощью разработанного программного средства на примере формирования команды для выполнения IT-проекта. Для такой команды определено множество функциональных и личностных ролей, а также минимальные значения критериев $K^*(y_i)$ и $L^*(y_i)$. Сформированы множества претендентов на каждую функциональную роль $X_j=\{x_i\}$. Значения весовых коэффициентов q_u и g_v приняты одинаковыми, $a_1=0{,}65$, $a_2=0{,}35$. В таблице представлены результаты решения задачи в развернутом виде, т.е. по каждой роли приведены оценки эффективности претендентов.

Таблица

Роль	Наименование	x_i	$P_j(x_i)$	Желаемая должность
y_1	Системный аналитик	x_8	0,8148	Системный аналитик
		x_{12}	0,80988	Системный аналитик
y_2	Руководитель разработки	x_{17}	0,93	Руководитель разработки
y_3	Разработчик	x_9	0,7193	Руководитель разработки
		x_8	0,4972	Системный аналитик
		x_{18}	0,4097	Разработчик
		x_{11}	0,0825	Разработчик
y_4	Системный администратор	x_6	0,0656	Системный администратор
y_5	Тестировщик	x_{13}	0,1065	Тестировщик
		x_6	0,1026	Менеджер по качеству
y_6	Менеджер по качеству	x_{19}	0,08666	Манеджер по качеству
y_7	Технический писатель	x_{15}	0,1253	Технический писатель
		x_{11}	0,039	Технический писатель

Из таблицы видно, что для команды $Y=\{y_1,y_2,y_3,y_4,y_5,y_6,y_7\}$, в которой требуется по одному исполнителю на каждую функциональную роль, наилучшим вариантом является $\{x_8,x_{17},x_9,x_6,x_{13},x_{19},x_{15}\}$. При этом претенденты x_6,x_{13},x_{19},x_{15} имеют достаточно низкие оценки привлекательности $P_j(x_i)$, что должен учесть руководитель проекта при принятии решений. На основании приведенной информации формируется также команда, в которой на одну функциональную роль необходимо подобрать несколько исполнителей.

Выводы

Построенная в работе математическая модель определения оптимального состава команды проекта и оценки ее эффективности, а также программное средство могут быть использованы в составе системы поддержки принятия решений по управлению проектом в любой предметной области. Вычислительные эксперименты показали эффективность предложенного подхода в смысле минимизации времени на обработку информации о претендентах на работу специалистами по персоналу за счет автоматизации и комплексного учета всевозможных выдвигаемых требований.

Предложенную модель целесообразно использовать при формировании команд в случае, когда

претенденты незнакомы друг с другом и не могут высказать суждений об отношениях, определяющих психологическую совместимость, что является одним из важнейших факторов сплоченности команды. Дальнейшее развитие модели предполагает учет социометрических параметров претендентов и команды.

Список литературы: **1.** *Управление* проектами: учебное пособие [Текст] / под ред. И.И. Мазура. — 2-е изд. — М.: Омега-Л, 2004. — 664 с. **2.** *Новиков*, *Д.А.* Математические модели формирования и функционирования команд [Текст] / Д.А. Новиков. – М.: ПМСОФТ, 2007. – 140 с. **3.** *Белбин, Р.М.* Типы ролей в командах менеджеров [Текст]: Пер с англ / Р.М. Белбин. – М.: Гиппо, 2003. – 216 с. **4.** *Паркер, Г.* Формирование команды [Текст] / Г. Паркер, З. Кролл – СПб.: Питер, 2003. – 560 с. **5.** *Мескон, М.* Основы менеджмента [Текст]: Пер. с англ. / М. Мескон:– М.: Дело, 1999. – 800 с. **6.** *Минцберг*, *Г.* Структура в кулаке: создание эффективной организации [Текст]: Пер. с англ. / под ред. Ю. Н. Каптуревского / Г. Минцберг:– СПб.: Питер, 2004. – 512 с. **7.** *Карякин*, *А.М.* Командная работа: основы теории и практики [Текст] / А.М. Карякин; Иван. гос. энерг. ун-т. – Иваново, 2003. – 136 с. **8.** *Харитонов*, *В.Н.* Формирование команды проекта реконструкции системы теплоснобжения [Текст] / В.Н. Харитонов // Весник инженерной академии Украины. – 2008 № 3-4. – С. 263. **9.** *Петров*, *Э.Г.* Методы и средства принятия решений в социально-экономических системах [Текст] / Э.Г. Петров, М.В. Новожилова, И.В. Гребенник, Н.А. Соколова. – Херсон: ОЛДИ-плюс, 2003. – 380 с. **10.** *Петров, К.Э.* Компараторная структурно-параметрическая идентификация моделей скалярного многофакторного оценивания [Текст] / К.Э. Петров, В.В. Крючковский. – Херсон: Олди-плюс, 2009. – 294 с.

Поступила в редколлегию 30.08.2011

УДК 004.42:519.688:519.81

Синтез моделі багатофакторного оцінювання ефективності команди проекту / Н.І. Калита, А.Є. Гуріна // Біоніка інтелекту: наук.-техн. журнал. – 2011. – № 3 (77). – С. 70-73.

Аналізується одна із задач управління проектами – формування оптимальної команди виконання проекту. Для її розв'язання запропоновано математичну модель, яка враховує як професійні вимоги до претендентів, так і особистісні. Розглянуто результати обчислювальних експериментів.

Табл. 1. Бібліогр.: 10 найм.

UDK 004.42:519.688:519.81

Synthesis of multifactor model of evaluating the project team effectiveness / N.I. Kalita, A.V. Gurina // Bionics of Intelligense: Sci. Mag. – 2011. – № 3 (77). –P. 70-73.

One of the tasks of project management - formation of an optimal project team is analyzed. A mathematical model for it solving is offered that takes into account both professional and personality requirements for candidates. The results of computational experiments are considered.

Tab. 1. Ref.: 10 items.

УДК 518.81

О.А. Писклакова, Д.Е. Шмидт, В.С. Алексенко
ХНУРЕ, м. Харків, Україна, st@kture.kharkov.ua

МОДЕЛЬ ВЫБОРА ОПТИМАЛЬНОГО ПАРТНЕРА ПРЕДПРИЯТИЯ В УСЛОВИЯХ МНОГОКРИТЕРИАЛЬНОСТИ И НЕОПРЕДЕЛЕННОСТИ

Рассмотрена модель выбора оптимального партнёра предприятия в условиях многокритериальности и неопределенности. Проанализирован способ задания весовых коэффициентов и частных критериев в виде нечетких чисел.

ВЫБОР ОПТИМАЛЬНЫХ ПОСТАВЩИКОВ, МНОГОКРИТЕРИАЛЬНОСТЬ, ИНТЕРВАЛЬНАЯ НЕОПРЕДЕЛЕННОСТЬ, ФУНКЦИЯ ПОЛЕЗНОСТИ, МЕТОД РЕЙТИНГОВЫХ ОЦЕНОК

Введение

Рынок – система экономических отношений, складывающихся в процессе производства, обращения и распределения товаров и денежных средств, для которого характерна свобода субъектов в выборе покупателей и продавцов, определении цен, формировании и использовании ресурсных источников. [1].

Рыночные отношения заставляют всех участников системы производства, транспортировки и потребления объединить свои усилия таким образом, чтобы производить ровно столько товара, сколько требуется на рынке, и доставлять этот товар к моменту, определенному спросом в сфере потребления.

В настоящий момент времени перед большинством промышленных, а также коммерческих предприятий, стоит основная задача – снижение суммарных затрат на изготовление и реализацию продукции. При этом необходимо достигать заданного уровня качества товаров. Наиболее современным подходом, позволяющим оптимизировать стратегию развития предприятия, является логистический, рассматривающий взаимоувязанный комплекс проблем. Необходимость логистического подхода в практике хозяйственной деятельности предприятия обусловлена, прежде всего, переходом от рынка продавца к рынку покупателя, который требует гибкого реагирования производственных систем на быстро изменяющиеся приоритеты потребителей. Любое предприятие, не-зависимо от формы собственности, размера и вида деятельности, участвует в качестве логистического звена системы «поставщик – производитель – потребитель».

По мере усложнения рыночных условий функционирования предприятий повышается актуальность логистического подхода к управлению предприятием. В той мере, в какой предприятие базирует свое конкурентное преимущество на логистике, оно обладает теми уникальными свойствами, которые трудно воспроизвести другим предприятиям.

Цель логистической деятельности считается достигнутой, если выполнены следующие правила: нужный продукт необходимого качества доставлен с требуемым уровнем затрат нужному потребителю в необходимом количестве, в нужное время и в нужное место [2].

Одной из наиболее важных задач логистики является выбор поставщика

Важность ее объясняется не только тем, что на современном рынке функционирует большое количество поставщиков одинаковых товаров, но главным образом тем, что поставщик должен быть надежным партнером фирмы в реализации ее логистической стратегии.

Тенденции покупок вместо собственного производства ориентированы на улучшение качества, снижение уровня запасов, интеграцию систем поставщиков и покупателей в единую логистическую систему взаимодействия и координации в логистических каналах совместного бизнеса, обусловив тем самым потребность повысить эффективность работы с поставщиками. В настоящее время наблюдается повышение внимания к тщательному выбору поставщиков и предъявлению к ним более высоких требований.

Процесс выбора поставщика включает в себя поиск и отбор предприятием потенциальных поставщиков сырья, материалов, комплектующих изделий.

Оценивание поставщика проводится с точки зрения обеспечения поставок продукции требуемого качества, в требуемые сроки и по приемлемой цене.

Эффективные решения по выбору источников снабжения являются основой создания устойчивой базы снабжения любой компании. Обычно решение покупателя зависит от его оценки способности поставщика удовлетворять критериям качества, объема, условий доставки, цены и обслуживания.

На практике задача выбора оптимального поставщика решается в условиях неполноты знаний о взаимосвязи показателей поставщиков и, как следствие, неточного ее описания, невозможности

или неточности измерения некоторых показателей, случайных внешних и внутренних воздействий и т.д. Дополнительная сложность заключается в том, что показатели разнородны и могут быть представлены в виде случайных величин, нечетких множеств или просто интервальных величин.

Целью настоящей статьи является анализ особенностей и обоснование модели выбора оптимального партнера предприятия в условиях многокритериальности и неопределенности исходных данных.

1. Постановка задачи

На практике выделяют следующие этапы решения задачи выбора поставщика:

1. Поиск потенциальных поставщиков.

При этом могут быть использованы следующие методы:

– объявление конкурса;

– изучение рекламных материалов: фирменных каталогов, объявлений в средствах массовой информации и т. п.;

– посещение выставок и ярмарок;

– переписка и личные контакты с возможными поставщиками.

В результате перечисленных мероприятий формируется список потенциальных поставщиков, который постоянно обновляется и дополняется.

2. Анализ потенциальных поставщиков.

Составленный перечень потенциальных поставщиков анализируется на основании специальных критериев, позволяющих осуществить отбор приемлемых поставщиков. Количество таких критериев может составлять несколько десятков. Однако зачастую ограничиваются ценой и качеством поставляемой продукции, а также надежностью поставок, под которой понимают соблюдение поставщиком обязательств по срокам поставки, ассортименту, комплектности, качеству и количеству поставляемой продукции.

К другим критериям, принимаемым во внимание при выборе поставщика, относят следующие:

– удаленность поставщика от потребителя;

– сроки выполнения текущих и экстренных заказов;

– наличие резервных мощностей;

– организация управления качеством у поставщика;

– психологический климат у поставщика (возможности забастовок);

– способность обеспечить поставку запасных частей в течение всего срока службы поставляемого оборудования;

– финансовое положение поставщика, его кредитоспособность и др.

Также следует обращать внимание на характеристики поставщика, которые свидетельствуют о его способности соответствовать заявленным требованиям покупателя: предыдущая история компании, развитость инфраструктуры, финансовое положение, организация и управление, репутация, местонахождение. С точки зрения управления отношениями с поставщиками ценность приобретенного потребителем товара или услуги – это окончательная (в долгосрочной перспективе) полезность этого приобретения. Это необязательно самая низкая цена покупки, минимальные инвестиции в запасы, самое быстрое время доставки, или самый быстрый жизненный цикл, или самое высокое качество; это оптимальное сочетание всего перечисленного.

В результате анализа потенциальных поставщиков формируется перечень конкретных поставщиков, с которыми проводится работа по заключению договорных отношений.

3. Оценка результатов работы с поставщиками.

На выбор поставщика существенное влияние оказывают результаты работы по уже заключенным договорам. Для этого разрабатывается специальная шкала оценок, позволяющая рассчитать рейтинг поставщика [3].

В общем случае задачу выбора оптимальных партнеров можно свести к функции полезности (1).

$$P(x) = \sum_{i=1}^{n} a_i k_i^{\text{Н}}(x), \qquad (1)$$

где $P(x)$ – функция полезности; a_i – относительные безразмерные весовые коэффициенты, для которых выполняются ограничения (2)

$$\begin{cases} 0 \le a_i \le 1 \\ \sum\limits_{i=1}^{n} a_i = 1, \forall i = \overline{1,n} \end{cases}, \qquad (2)$$

а $k_i^{\text{Н}}(x)$ – нормализованные, т.е. приведенные к изоморфному виду частные критерии [4]. Нормализация критериев проводится по формуле

$$k_i^{\text{Н}}(x) = \left(\frac{k_i(x) - k_i^{\text{НХ}}}{k_i^{\text{НЛ}} - k_i^{\text{НХ}}} \right)^{\alpha_{\text{Ш}}}, \qquad (3)$$

где $k_i(x)$ – значение частного критерия; $k_i^{\text{НЛ}}$, $k_i^{\text{НХ}}$ – соответственно наилучшее и наихудшее значение частного критерия, которое он принимает на области допустимых решений $x \in X$.

В зависимости от вида экстремума (направления доминирования)

$$k_i^{\text{НЛ}} = \begin{cases} \max\limits_{x \in X} k_i(x), \text{ если } k_i(x) \to \max \\ \min\limits_{x \in X} k_i(x), \text{ если } k_i(x) \to \min \end{cases}, \qquad (4)$$

$$k_i^{\text{НХ}} = \begin{cases} \min\limits_{x \in X} k_i(x), \text{ если } k_i(x) \to \max \\ \max\limits_{x \in X} k_i(x), \text{ если } k_i(x) \to \min \end{cases}. \qquad (5)$$

общая модель определения полезности решения $x \in X$ имеет вид

$$P(x) = G\left[J(a_i), k_i(x)\right],\ i = \overline{1,n}, \qquad (6)$$

где $J(a_i)$ – информация о значениях коэффициентов относительной важности.

Крайними ситуациями являются случаи, когда:

1) весовые коэффициенты a_i заданы в виде точных точечных количественных значений;

2) информация о предпочтительности частных критериев полностью отсутствует.

Как правило, между этими крайностями имеется множество ситуаций с различной степенью неопределенности задания весовых коэффициентов.

В инженерной практике часто встречаются ситуации принятия решения многокритериальных решений в условиях, когда предпочтения частных критериев (весовые коэффициенты a_i) заданы в виде интервалов возможных значений $[a_{i\min},\ a_{i\max}],\ \forall_i = \overline{1,n}$. При этом возможны следующие случаи:

– задание числовых значений a_i в виде интервалов, без указания предпочтений внутри интервала;

– задание распределения a_i на интервале возможных значений в виде вероятностных характеристик;

– задание возможных значений a_i с помощью лингвистических переменных.

Общим для всех перечисленных выше случаев условием корректности задания интервалов возможных значений a_i является одновременное выполнение условий

$$\sum_{i=0}^{n} a_{i\max} > 1;\quad \sum_{i=0}^{n} a_{i\min} < 1, \qquad (7)$$

Таким образом, экстремальное решение $x^\circ \in X$ (оптимальный поставщик) определяется по формуле

$$x^\circ = \arg\max_{x_i \in X} P(x_i),\ i = \overline{1,n}. \qquad (8)$$

Необходимо рассмотреть два случая: когда a_i заданы с помощью лингвистических переменных и когда весовые коэффициенты заданы в виде нечетких множеств.

2. Дефазификация весовых коэффициентов, заданных нечеткими числами

Для конструктивного решения задачи многокритериальной оптимизации необходимо определить численные значения весовых коэффициентов a_i. Эта задача может быть решена двумя способами: методом экспертного оценивания или методом компараторной идентификации [5]. Однако в обоих случаях можно определить не точечную, а интервальную оценку значений a_i, $i = \overline{1,n}$. Это обусловлено тем, что оба метода базируются на определении, хотя и разными способами, некоторого ограниченного множества индивидуальных, субъективных оценок и последующей их обработке путем усреднения [4]. Таким образом, исходную информацию можно представить в виде

$$a_j^{\min} \le a_j \le a_j^{\max},\ \forall j = \overline{1,n}, \qquad (9)$$

а точечную оценку, как

$$a_j^{\text{ср}} = \frac{a_j^{\min} + a_j^{\max}}{2};\ \forall j = \overline{1,n}. \qquad (10)$$

При этом в общем случае

$$\sum_{j=1}^{n} a_j^{\min} < 1,\ \sum_{j=1}^{n} a_j^{\max} > 1,\ \sum_{j=1}^{n} a_j^{cp} \ne 1, \qquad (11)$$

т.е., не выполняется ограничение (2).

Необходимо устранить возникающую в процессе идентификации параметров a_j, $j = \overline{1,n}$ интервальную неопределенность, т.е. детерминировать параметры модели многокритериального оценивания (1).

Один из возможных подходов к решению сформулированной задачи заключается в интерпретации интервалов (9), как нечетких множеств.

Будем полагать, что кортеж весовых коэффициентов $A = < a_i >, i = \overline{1,n}$ задан в виде интервалов, на которых экспертным путем определены функции принадлежности $\mu(a_i), i = \overline{1,m}$. При этом значения a_i^* соответствуют функции принадлежности $\mu(a_j^*) = 1$. Носителем каждой нечеткой оценки a_j, $\forall j = \overline{1,n}$, является соответствующий интервал $[a_j^{\min}, a_j^{\max}]$, а a_j^* соответствует $\mu(a_j^*) = 1$. Для простоты, но без потери общности рассматриваются функции принадлежности треугольного вида. Необходимо определить детерминированные точечные значения $a_j^{\text{Д}}$, при этом должно выполняться условие

$$\sum_{j=1}^{m} a_j^{\text{Д}} = 1\ . \qquad (12)$$

Если $\sum_{j=1}^{m} a_j^* = 1$, то решение задачи тривиально, т.е. $a_j^{\text{Д}} = a_j^*$. В противном случае, т.е. в случае если $\sum_{j=1}^{m} a_j^* \ne 1$ необходимо определить такой кортеж значений $A^{\text{Д}} = < a_j^{\text{Д}} >$ для которого:

1) удовлетворяется условие $\sum_{j=1}^{n} a_i^* = 1$;

2) минимизируется отклонение

$$\Delta\mu_{\sum a_j} = \mu\Big(\sum_{j=1}^{n} a_j^*\Big) - \mu\Big(\sum_{j=1}^{n} a_j\ \Big) \qquad (13)$$

или с учетом, что $\mu\left(\sum_{j=1}^{n} a_i^*\right) = 1$, максимизируется $\mu\left(\sum_{j=1}^{n} a_i^{\text{Д}}\right)$.

Так как, по определению, значение функции принадлежности суммы нечетких чисел определяется минимальным значением μ_{a_j} на множестве слагаемых (1), то очевидно, что условие (13) будет выполняться при равенстве всех $\mu(a_j)$, $\forall j = \overline{1,n}$. Это означает, что в качестве $a_j^{Д}$, $\forall j = \overline{1,n}$, необходимо выбрать такие значения из интервала неопределенности, для которых

$$\mu(a_j^{Д}) = \mu(\sum_{j=1}^{n} a_j^{Д} = 1), \forall j = \overline{1,n}. \qquad (14)$$

Для треугольных функций принадлежности нетрудно получить следующую формулу

$$a_j^{Д} = a_j^{*} \pm [1 - \mu(\sum_{j=1}^{n} a_i^{Д} = 1)] \cdot \frac{a_j^{\max} - a_j^{\min}}{2}. \qquad (15)$$

Знак (+) соответствует, случаю, когда

$$\sum_{j=1}^{n} a_j^{*} < 1, \qquad (16)$$

а (-) – противоположному случаю. Формула достаточно хорошо аппроксимирует и нелинейные (колоколообразные, гауссовы) функции принадлежности [6].

3. Учет частных критериев, заданных нечеткими множествами

На практике частные критерии чаще всего задаются в интервальном виде, а статистическая информация о характере распределения значений внутри интервала неизвестна. Эксперт в таком случае может назначить функцию принадлежности внутри интервала. Тогда значение частного критерия будет представлено в виде нечеткого числа с функцией принадлежности, при этом то значение функции принадлежности, при котором она равна единице, необязательно будет находиться на середине интервала.

Для вычисления функции полезности, когда частные критерии заданы в виде НЧ, а весовые коэффициенты – в виде детерминированных значений, необходимо дефазифицировать частные критерии и найти точечные значения оценки альтернатив. Для этого необходимо выделить модальные значения нечетких чисел, т.е. значения, при которых функция принадлежности равна 1, и вычислить детерминированную оценку функции полезности. При этом значения весовых коэффициентов должны удовлетворять условию (2).

Выводы

В статье был выполнен анализ особенностей и обоснование модели выбора оптимального партнера предприятия в условиях многокритериальности и неопределенности исходных данных.

Было разработано программное средство, которое позволяет осуществлять выбор оптимальных поставщиком методом рейтинговых оценок. При этом критерии поставщиков могут быть заданы либо детерминированно, либо лингвистически.

Разработанное программное средство призвано повысить эффективность предприятия при выборе партнеров. Благодаря этому должны снизиться затраты, время выполнения, а также возможные риски данного бизнес-процесса, что приведет к снижению себестоимости конечной продукции.

Программное средство предусматривает для решения данной задачи эффективный набор аналитических средств, автоматизирующих и упрощающих подбор оптимального поставщика по заданным критериям.

Список литературы: 1. *Троицкая, Н.А.* Единая транспортная система: учеб. пособие [текст] / Н. А. Троицкая, А. Б. Чубуков. – М.: AKADEMIA, 2003. – 240 с. **2.** *Миротин, Л.Б.* Логистика: обслуживание потребителей: учеб. пособие [текст] / Л.Б. Миротин, И.Э. Ташбаев, А.Г. Касенов – М.: ИНФРА-М, 2002. – 190 с. **3.** *Гаджинский, А.М.* Логистика [текст] / А.М. Гаджинский. – М.: Информационно-внедренческий центр "Маркетинг", 1999. – 228 с. **4.** *Катулев, А.Н.* Современный синтез критериев в задачах принятия решений [текст] / А.Н. Катулев, Л.С. Виленчук, В.Н. Михно. – М.: Радио и связь, 1992. – 119 с. **5.** *Петров, Е.Г.* Методи і засоби прийняття рішень в соціально-економічних системах [текст] / Е.Г. Петров, М.В. Новожилова, І.В. Гребеннік. – К.: Техніка, 2004. – 256 с. **6.** *Бодров, В.И.* Математические методы принятия решений: учеб. пособие [текст] / В.И. Бодров, Т.Я. Лазарева, Ю.Ф. Мартемьянов. – Тамбов: Изд-во Тамб. гос. тех. ун-та, 2004. – 124 с. **7.** *Петров, Э.Г.* Детерминизация нечетких параметров модели многокритериального оценивания [текст] / Э.Г. Петров, О.А. Писклакова, Н.А. Брынза // Вестник ХГТУ. – 2008. – №2(31). – С. 71-75.

Поступила в редколлегию 07.09.2011

УДК 518.81

Модель вибору оптимального партнера підприємства в умовах багатокритеріальності і невизначеності / О.О. Писклакова, Д.Є. Шмідт, В.С. Алексенко // Біоніка інтелекту: наук.-техн. журнал. – 2011. – № 3 (77). – С. 74-77.

Запропонована модель вибору оптимального партнера підприємства в умовах багатокритеріальності і невизначеності. Розглянуто випадки, коли вагові коефіцієнти та часткові критерії задані у вигляді нечітких чисел.

Бібліогр.: 7 найм.

UDK 518.81

Model for selecting the best partner of a company in a multi-criteria and uncertainty / O.A. Pisklakova, D.E. Schmidt, V.S. Aleksenko// Bionics of Intelligense: Sci. Mag. – 2011. – № 3 (77). – P. 74-77.

The model for selecting the best partner of a company in a multi-criteria and uncertainty was proposed. The cases were considered when the weights and the particular criteria are defined as fuzzy numbers.

Ref.: 7 items.

УДК 519.81

В.П. Писклакова[1], О.А. Писклакова[2], А.А. Пряничникова[3]

[1, 2] ХНУРЭ, г. Харьков, Украина, ST@ kture.kharkov.ua

[3] ХНУРЭ, г. Харьков, Украина, stdep@kture.kharkov.ua

СОЗДАНИЕ РЕГИОНАЛЬНОГО МОНИТОРИНГА КАК СРЕДСТВА РЕАЛИЗАЦИИ КОНЦЕПЦИИ УСТОЙЧИВОГО РАЗВИТИЯ СОЦИАЛЬНО-ЭКОНОМИЧЕСКИХ СИСТЕМ

В данной работе рассматриваются цели и задачи информационного мониторинга как средства получения полной, актуальной, достоверной информации о состоянии социально-экономических систем (СЭС) как объекта комплексного управления в рамках концепции устойчивого развития.

КОНЦЕПЦИЯ, УСТОЙЧИВОЕ РАЗВИТИЕ, СОЦИАЛЬНО-ЭКОНОМИЧЕСКИЙ МОНИТОРИНГ, РЕГИОНАЛЬНОЕ УПРАВЛЕНИЕ

Введение

Высокие темпы развития экономики и технологического прогресса повлекли за собой необходимость гармонизации социальной, культурной, экологической жизни населения. На современном этапе человечество столкнулось со всё обостряющимися противоречиями между своими растущими потребностями и неспособностью биосферы обеспечить их, не разрушаясь. В результате социально-экономическое развитие приняло характер ускоренного движения к глобальной экологической катастрофе. При этом ставится под угрозу не только удовлетворение жизненно важных потребностей и интересов будущих поколений людей, но и сама возможность их существования. Возникла идея разрешить это противоречие путем перехода к такому развитию общества, которое не разрушает своей природной основы, гарантируя человечеству возможность выживания и дальнейшего управляемого и устойчивого развития.

Идеи и принципы, концепция и стратегия устойчивого развития изложены в решениях конференции ООН по охране окружающей среды и развитию (Рио-де-Жанейро, 1992 г.) [1]. Для перехода на путь устойчивого развития каждое государство должно разрабатывать собственную программу действий, с учетом собственных, существующих в данный период, тенденций в социальной, экономической и экологической сфере. Министерство экологии и природных ресурсов Украины выступило инициатором создания такой стратегии и в 2004 году она должна быть разработана [2].

1. Условия реализации концепции устойчивого развития общества

Разработка стратегии устойчивого развития для Украины должна быть основана на тщательном, всестороннем изучении существующего состояния нашего общества, а также на анализе потенциальных возможностей его улучшения [2].

Состояние современного украинского общества характеризуется обилием разнообразных проблем, большое количество которых требует безотлагательного разрешения. Все эти проблемы тесно взаимосвязаны, и поэтому решать их необходимо комплексно. Но для этого необходимо преодолеть стереотипы мышления, например о противоречии экологии и экономики, то любые экологические мероприятия затратны для производства. На самом деле существует множество примеров, когда деньги, вложенные в экологизацию производства, приносят прибыль[2].

Устойчивое развитие подразумевает создание условий, обеспечивающих удовлетворение потребностей сегодняшнего дня, не подвергая существование последующих поколений большему риску, чем нынешний. Устойчивое развитие – это гармоничное регулируемое развитие: целенаправленный контроль над происходящими изменениями в СЭС всех уровней, прогнозирование и компенсация наиболее опасных неустойчивостей и диспропорций развития.

Мировое сообщество уже осознало пагубность концепции «экономического роста» и переходит к так называемой концепции «устойчивого развития». Концепция «экономического роста» предполагает принятие решения всего по одному скалярному критерию – «максимуму ожидаемой прибыли», в то время как концепция «устойчивого развития» при решении любых задач опирается на следующие три критерия: экономический, социальный и экологический.

Концепция устойчивого развития – модель развития цивилизации, которая исходит из необходимости обеспечить мировой баланс между решением социально-экономических проблем и сохранением окружающей среды. Это модель развития общества, при которой удовлетворение жизненных потребностей нынешнего поколения людей достигается не за счет лишения такой возможности будущих поколений.

В концепции устойчивого развития делается акцент на том, что экономический рост сам по себе не может рассматриваться как основной критерий

развития. Ведь этот рост по своей сути осуществляется за счет уничтожения природных ресурсов, и тогда последующие поколения вообще не будут иметь к ним доступа. Сюда относятся не только нефть, газ и полезные ископаемые, но и питьевая вода, воздух и т.п. Чтобы этого не допустить, сторонники концепции устойчивого развития предлагают гармонизировать экономическую, экологическую и социальную составляющие развития общества. Экономическая составляющая — сохранение совокупного капитала, с помощью которого должны производиться в необходимом объеме продукты потребления и услуги. Социальная составляющая устойчивого развития подчеркивает важность сохранения стабильности существующих социальных систем, в том числе, сокращения числа разрушающих конфликтов в обществе, путем справедливого распределения благ между людьми. Экологическая составляющая – сохранение возможности самовосстановления экосистем[3, 4].

Существенное достоинство концепции устойчивого развития в том, что она учитывает не только экологический, но и временной фактор. Она ориентирована не на быстрое получение результата, а на длительную перспективу развития. Модель устойчивого развития основана на идее равенства интересов настоящего и будущих поколений. Таким образом, концепция устойчивого развития является формой регулирования социальной ответственности современного общества и государства за создание условий для будущих поколений удовлетворять разнообразные потребности – физиологические, экономические, духовные и иные – в процессе взаимодействия с природой.

Для успешной реализации идей устойчивого развития в Украине необходимо следующее:

– информирование всех слоев населения о проблемах устойчивого развития. Несмотря на то, что эта концепция в настоящий момент находится в центре внимания мирового сообщества, в Украине многие о ней либо не знают, либо не обращают на нее должного внимания;

– смена приоритетов при разработке экономической, промышленной, энергетической, сельскохозяйственной политики. Необходимо реформировать налоговую систему, в частности перейти на рентную систему налогообложения, поддерживать развитие малого бизнеса, привлекать внутренние и внешние инвестиции;

– формирование эффективной внешней политики (особенно в отношении импорта/экспорта). В настоящее время украинский рынок заполнили импортированные товары, хотя многие из них могут быть заменены товарами намного лучшего качества украинского производителя;

– совершенствование нормативной базы. Необходимо создать подходящие нормативно-правовые условия для устойчивого развития;

– использование новых информационных технологий для обмена информацией по устойчивому развитию, для создания баз данных и моделирования локальных социально-экономических и экологических систем;

– развитие местного самоуправления. Устойчивое развитие предполагает тесное взаимодействие всех секторов общественной жизни, что является затруднительным при централизованном отраслевом управлении [2].

Поэтому необходимо передать часть полномочий местным властям в рамках развития системы местного самоуправления. В Хартии устойчивого развития европейских городов это отражено в принципе субсидиарности, который предполагает наличие у органов местного самоуправления стольких полномочий сколько возможно, а у центральных – сколько необходимо, т.е. полномочия на решение соответствующей проблемы должны предоставляться властным структурам на как можно более низшем уровне, который способен решить эту проблему. Оптимально таким уровнем являются муниципальные власти, поскольку в силу своих полномочий в области планирования, инвестиций, регулирования, контроля и исполнения решений они являются самыми близкими к гражданам органами, то есть органами, которые могут непосредственно реагировать на пожелания и устремления своих избирателей. Процесс устойчивого развития городов позволит обеспечить городским властям обратную связь, показывая, какие виды деятельности ведут к сбалансированности городского развития, а какие, наоборот, препятствуют этому [2].

Устойчивое развитие городов опирается на следующие базовые принципы:

– города представляют собой, с одной стороны, наибольшую территориальную единицу, население которой непосредственно испытывает на себе нарушения социального, архитектурного, экономического, ресурсного и экологического равновесия. С другой стороны, городской уровень – это тот наименьший масштаб, на котором эти проблемы могут найти конструктивное целостное решение в реализуемых стратегиях развития;

– признается важность рационального использования территории и формирования эффективной политики пространственного планирования, включающей стратегическую экологическую оценку всех планов. Преимущество отдается созданию многофункциональных зон, совмещающих жилье, места работы и оказания услуг, чтобы сократить потребность в переездах и, соответственно, снизить уровни загрязнения.

– признается важность партнерства всех секторов общества (власть, бизнес, общественность), обеспечивается доступ всех граждан и заинтересованных групп к необходимой информации и

участие каждого жителя в разработке локальных стратегий развития, осуществляются программы обучения и подготовки по вопросам устойчивого развития как для широкой общественности, так и для работников органов местного самоуправления.

Таким образом, оптимальным уровнем реализации идей устойчивого развития является муниципальный уровень, и именно на этом уровне в Украине наметились положительные сдвиги. Ряд городов подписал Ольборгскую Хартию. Сдвинулся с мертвой точки процесс взаимодействия общественности (в лице общественных организаций) с властью и бизнесом [2].

Для создания системы, основанной на описанной выше концепции, необходимо контролировать, в каком начальном состоянии находится объект, т.е. текущее состояние планетарной СЭС – в глобальном смысле и состояние конкретного объекта исследования в частности, т. е. создать систему комплексного мониторинга.

Мониторинг можно представить как систему, специальным образом организованную и действующую постоянно, которая помогает производить сбор, анализ и обработку информации, позволяет вести необходимую статистическую отчетность, проводить оценку состояния объекта или процесса, а также прогнозировать его дальнейшее развитие.

2. Мониторинг как компонент планирования

Мониторинг является очень важным компонентом при планировании, принятии решений. Своевременное вмешательство в деятельность того или иного объекта иногда позволяет избежать очень серьезных ошибок и негативных результатов. Поэтому мониторинг изменения ситуации на протяжении некоторого определенного промежутка времени необходим для оценки эффективности вмешательства и его влияния на исследуемый объект. Мониторинг также позволяет сопоставить потребности и определить очередность осуществления запланированных действий по отношению к данному объекту.

Основная цель мониторинга — сбор, изучение и подготовка информации для принятия и анализа решений, повышение эффективности принятия управленческих решений. Из этого можно сделать вывод, что мониторинг как система сбора и обработки информации должна удовлетворять нескольким условиям.

Во-первых, мониторинг должен иметь некую целевую направленность, то есть производиться исходя из поставленной задачи.

Во-вторых, мониторинг должен обеспечить максимальную точность и объективность получаемых результатов на каждой стадии обработки данных.

Таким образом, можно выделить следующие основные задачи социально-экономического и экологического мониторинга:

– оценивание текущего состояния объекта и процессов, происходящих в нем;

– анализ данных для установления определенных закономерностей хода процессов;

– кратковременное прогнозирование процессов, которые могут происходить в исследуемом объекте;

– визуализация результатов мониторинга и представление их конкретному пользователю в удобном виде.

Также можно выделить еще несколько задач мониторинга:

– организация наблюдения, получение достоверной и объективной информации о протекании процессов на исследуемом объекте;

– оценка и системный анализ получаемой информации, выявление причин, вызывающих тот или иной характер протекания процессов;

– обеспечение в установленном порядке органов управления, предприятий, учреждений и организаций независимо от их подчиненности и форм собственности, граждан информацией, полученной при осуществлении социально-экономического и экологического мониторинга;

– выявление факторов, которые могут спровоцировать негативные последствия в настоящем времени и в перспективе;

– разработка прогнозов развития ситуации;

– подготовка рекомендаций, направленных на то, чтобы не допустить негативных последствий и обеспечить поддержку позитивных тенденций развития, а также доведение их до соответствующих органов управления.

Рассматривая мониторинг как систему наблюдения за экономическими, социальными и экологическими изменениями в окружающем мире можно прийти к выводу, что все три направления очень тесно связаны между собой.

Социально-экономическая структура является сложной системой, для управления которой органы власти должны точно описывать протекающие в ней процессы, правильно их интерпретировать и вырабатывать соответствующие реакции. Вместе с тем, направление и эффективность разрабатываемой и осуществляемой органами власти разного уровня социально-экономической политики зависит от системы координат (индикаторов), с помощью которой эта политика оценивается. Чтобы быстро и надежно оценить результативность развития территории (области, района, города и т.п.), надо иметь интегральные показатели, которые были бы достаточно чувствительны к изменениям социально-экономической ситуации во времени и в пространстве и которые можно было бы количе-

ственно измерить. Здесь встает вопрос о поведении социально-экономического мониторинга.

А.Е. Когут определяет социально-экономический мониторинг как оценку, прогноз, систему наблюдения и анализа экономической и социальной обстановки, складывающейся на территории, а также выработку рекомендаций по принятию рациональных управленческих решений [5].

Главной целью социально-экономического мониторинга принято считать обеспечение органов государственной власти полной, достоверной и своевременной информацией об изменениях происходящих в обществе, о процессах, протекающих в различных сферах жизнедеятельности населения не только на конкретной территории, но и на территории государства в целом. Также одной из основных целей социально-экономического мониторинга является сбор, изучение и подготовка информации для приятия на различных уровнях оптимальных управленческих решений. Это обусловливает две особенности, которым должен удовлетворять мониторинг как система сбора и обработки информации: целевая направленность информационных процессов и максимальная объективность получаемых выводов на каждой стадии переработки данных.

Схематично систему социально-экономического мониторинга можно представить следующим образом (рис. 1):

Такая система предназначена обеспечивать непрерывное отслеживание информации о социально-экономическом состоянии объектов и муниципальных образованиях, анализ и оценку происходящих в них процессов, среднесрочное и долгосрочное прогнозирование социально-экономического развития объектов и муниципальных образований.

Социально-экономическое состояние населения характеризует базовые аспекты жизни населения страны, которыми определены макроэкономическая и демографическая ситуации, состояние охраны здоровья, уровень образования, уровень занятости и условия труда, жилищные условия, степень социального напряжения, экологические условия проживания населения.

Можно выделить следующие характерные черты социально-экономического мониторинга:

1) мониторинг представляет собой постоянное продолжительное действие;

2) включает в себя систематическое наблюдение и сбор информации о параметрах социально-экономической системы;

3) носит системный характер, что является важным условием его эффективности;

4) сбор мониторинговых данных осуществляется при помощи разнообразных методов в зависимости от изучаемых подсистем;

5) полученные данные подвергаются обработке: анализу и диагностике;

6) мониторинговые данные должны быть сохранены для дальнейшего использования в принятии управленческих решений на основе моделирования и прогнозирования в социально-экономической ситуации в регионе;

7) эффективность мониторинговых исследований связана с правильной постановкой цели, адекватной и своевременной информацией, проведением и использованием результатов анализа;

8) охват всех значимых социальных явлений в регионе;

9) наличие определенного постоянного состава показателей и индикаторов (социологических и статистических);

10) наличие временных показателей, дополняющих основную систему и изменяющихся в за-

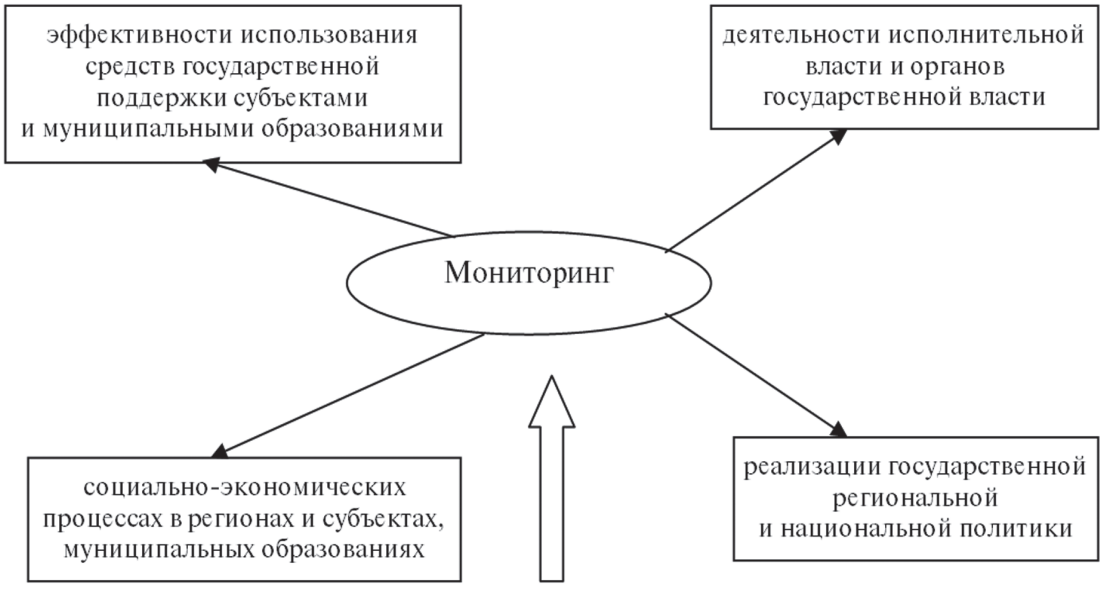


Рис. 1. Система мониторинга

висимости от потребностей пользователя, что обеспечивает гибкость системы мониторинга;

11) передача данных по каналам связи на центральный вычислительный центр, их обработка и хранение;

12) проведение мониторинга из единого организационного центра;

13) организация доступа потребителей к имеющейся информации.

Задачи социально-экономического мониторинга: как правило, сводятся к следующему [6]:

– организация наблюдения, т.е. получение достоверной и объективной информации о протекании на территории муниципального образования социально-экономических процессов;

– оценка и системный анализ имеющейся информации, выявление причин, влияющих на протекание экономических процессов;

– обеспечение в установленном порядке органов управления, предприятий, учреждений и организаций независимо от их подчиненности и форм собственности, а также граждан информацией, полученной при осуществлении социально-экономического мониторинга;

– разработка прогнозов развития социально-экономической ситуации;

– подготовка рекомендаций, направленных на преодоление негативных и поддержку позитивных тенденций.

Основные принципы организации сбора информации по социально-экономической тематике:

– целенаправленность – вся система рационального мониторинга должна быть ориентирована на решение конкретных управленческих задач;

– системный подход – рассмотрение региона как подсистемы более крупной общественной системы , исследование связей его с другими территориальными звеньями;

– комлексность – мониторинг отдельных сфер и направлений развития региона должен осуществляться в их взаимосвязи друг с другом; необходимо осуществлять последовательное решение всей совокупности задач мониторинга по каждому из его направлений;

– непрерывность в наблюдении за объектом;

– периодичность снятия информации о происходящих изменениях;

– сопоставимость применяемых показателей мониторинга во времени и другие.

Основными направлениями социально-экономического мониторинга являются:

а) бюджетный потенциал, определяемый величиной местных налогов и сборов, отчислениями от федеральных и региональных налогов и сборов, поступлениями от сдачи муниципального имущества в аренду и пр.;

б) производственный потенциал, определяемый структурой и объемом производства, величиной и эффективностью использования фондов;

в) инвестиционный потенциал, определяемый объемом привлеченных ресурсов;

г) социально-инфраструктурный потенциал, характеризуемый количеством и качеством объектов инфраструктуры;

д) демографический потенциал, определяемый общей численностью населения, динамикой роста/убыли, миграционными процессами;

е) трудовой потенциал, определяемый образовательными, квалификационными характеристиками, занятостью в разрезе отраслей [7].

Выделяют два основных вида индикаторов:

1) качественные показатели, фиксирующие наличие или отсутствие определенного свойства;

2) количественные показатели, определяющие меру выраженности, развития определенного свойства.

Количественные оценки мониторинга предполагают наличие таких характеристик социально-экономического развития муниципального образования, которые могут корректно выступать в виде совокупности численных значений и (или) их интегрального значения. Качественные оценки вышеназванных процедур исследования предполагают наличие неких эталонов, сравнение с которыми позволяет определить степень приближения или отклонения от заданного критерия. Качественные и количественные оценки эффективно использовать в совокупности [7].

К мониторингу социально-экономических показателей предъявляется ряд требований, среди которых основными являются следующие:

– во-первых, система социально-экономических показателей должна быть взаимоувязана в рамках общей схемы анализа экономической безопасности, допускающей проведение исследований как на федеральном, так и на региональном уровне;

– во-вторых, система социально-экономических показателей должна отвечать перечню основных угроз экономической безопасности;

– в-третьих, перечень социально-экономических показателей, используемых при мониторинге, должен быть минимальным и легко доступным с точки зрения наличия статистической информации;

– в-четвертых, результаты анализа в выбранной системе показателей должны допускать простую и наглядную проверку на непротиворечивость объективной реальности;

– в-пятых, социально-экономические показатели должны быть синхронизированы во времени и характеризовать определенные срезы социально-экономической ситуации [8].

Данные требования обусловливают перечень

показателей, который должен обеспечить, с одной стороны, проведение полного и достоверного анализа, а с другой – возможность быстрого получения информации и проведения необходимых расчетов с применением современных технических и программных средств на базе компьютерной техники.

Для дальнейшего рассмотрения и изучения социально-экономических индикаторов прежде всего необходимо определить требования к ним. Основные требования к индикаторам для мониторинга:

– простота для понимания;

– политическая значимость;

– возможность проведения количественных оценок,

– научная обоснованность и статистическая достоверность;

–возможность выявлять пространственные различия и изменения во времени;

– финансовая оправданность и техническая осуществимость;

– возможность проводить сравнительные оценки между государствами;

– возможность агрегировать значения индикаторов на национальном и межнациональном уровне;

– учет специфических особенностей различных стран;

– удобство для различных категорий пользователей и, прежде всего, лиц, принимающих решения.

Индикаторы могут содержать простую или агрегированную информацию. Различаются единичные и составные индикаторы. Единичные представляют собой отдельные параметры, связанные с эталонной величиной. Эталоном может быть цель (удаленность от цели), исходное состояние (удаленность от исходного состояния), пороговое значение (близость к исчезновению) или эталонный год (изменение с течением времени). Составные индикаторы объединяют разные единичные индикаторы, превращая их в другую общую единицу. Один способ заключается в превращении единичных индикаторов в безразмерные индексы путем деления их на эталонную величину. Другой подход предусматривает взвешенную трансформацию в общую единицу (перевод в эквиваленты). Затем эти единичные индикаторы могут быть обобщены. Менеджеров участков интересует, как правило, статистика и единичные индикаторы; политических деятелей на национальном уровне – составные индикаторы.

Рассмотрим некоторые из этих индикаторов более подробно.

Мониторинг здоровья населения должен осуществляться совместно с экологическим мониторингом, мониторингом уровня медицинского обслуживания, нормальных условий жилья, питания, отдыха.

Мониторинг уровня медицинского обслуживания населения предполагает оценку его обеспеченности амбулаторно-поликлиническими учреждениями и стационарной сетью, соответствие численности медперсонала принятым нормам, наличие и доступность лекарств, медикаментов.

Мониторинг уровня обеспеченности жильем предполагает помимо оценки обеспеченности, оценку уровня благоустройства жилья, характера заселения, соответствия современным планировочным и гигиеническим требованиям.

При мониторинге уровня жизни населения необходимо также оценить состояние и уровень изношенности объектов ЖКХ, обеспеченность населения услугами бытового характера, наличие и доступность этих услуг, которые потребитель сможет приобрести на среднюю заработную плату.

Мониторинг уровня занятости населения включает в себя помимо определения количественных и качественных показателей рынка труда определение структуры занятости по отраслям экономики, выявление динамики структуры отраслей.

Мониторинг обеспечения общественного порядка предполагает определение уровня преступности, числа совершаемых преступлений, числа лиц, совершивших преступления.

Информационный фонд социально-экономического мониторинга должен представлять собой систематизированные многолетние данные об экономической и социальной обстановке, складывающейся в разрезе, например, основных направлений мониторинга, нормативно-справочные материалы, сведенные в статистические регистры и базы данных [4].

Итак, можно сделать вывод, что социально-экономический мониторинг – это разветвленная система получения, обработки и хранения социологической о статистической информации по наиболее актуальным проблемам жизни общества.

Выводы

Таким образом, создание иерархической системы комплексного мониторинга является необходимым условием создания системы поддержки принятия решений (СППР) при реализации концепции устойчивого развития общества. Достаточным условием конструктивности и эффективности такой СППР является дополнение ее комплексом математических моделей, позволяющих формировать эффективные управляющие воздействия для перевода системы из идентифицированного текущего в заданное конечное состояние.

Список литературы: 1. Рио-де-Жанейрская декларация по окружающей среде и развитию. Документ ООН A/CONF.151/26/Rev.1 (Vol. I), Стр. 3–7. **2.** *Лебедев, М.А.*

Устойчивое развитие в Украине: проблемы и возможности [Текст]/ М.А. Лебедев//Проблеми стійкого розвитку України: Зб. доповідей міжнародної наукової конференції студентів. Київ, Всеукраїнська екологічна ліга, 2004. – С.15-18. **3.** *Згуровский, М.З.* Роль инженерной науки и практики в устойчивом развитии общества [Текст]/ М.З. Зауровский, Г.А. Статюха //Системні дослідження та інформаційні технології. – 2007. – №1. – С. 19-38. **4.** *Згуровський, М.З.* Сталий розвиток у глобальному і регіональному вимірах [Текст] /М.З. Згуровский. Київ: Політехніка, 2006. – 83 с. **5.** Основы местного самоуправления в городах России [Текст] / Под ред. А.Е.Когута. // ИСЭП. – СП б, 1995. **6.** *Когут, А.Е.* Информационные основы регионального социально-экономического мониторинга [Текст]/ А.Е. Когут, В.С. Рохчин // ИСЭП РАН. – СПб, 1995. – С.27-32. **7.** *Ушвицкий, Л.И.* Мониторинг социально-экономической безопасности: методические основы [Электронный ресурс] / Л.И. Ушвицкий, В.Д. Протасов // Сборник научных трудов. Серия «Экономика». – СевКавГТУ. Ставрополь, 2002. – № 7. – Режим доступа: http://science.ncstu.ru/articles/econom/7/01.pdf/file_download. **8.** *Кузьмин, М.Н.* Мониторинг как составная часть информационного обеспечения процесса управления [Электронный ресурс] / М.Н. Кузьмин. – Режим доступа: http://sisupr.mrsu.ru/2009-1/pdf/17_Kyzmin.pdf.

Поступила в редколлегию 30.09.2011

УДК 519.81

Створення регіонального моніторингу як засобу реалізації концепції сталого розвитку соціально-економічних систем / В.П. Пісклакова, О.О. Пісклакова, А. А. Пряничнікова // Біоніка інтелекту: наук.-техн. журнал. – 2011. – № 3 (77). – С. 78-84.

Стаття присвячена створенню засобів реалізації стійкого розвитку соціально-економічних систем. Розглянуто проблеми, що виникають при досягненні зазначеної мети, а також шляхи їх вирішення. Передумовою для створення таких систем є наявність повної, достовірної інформації про соціально-економічні об'єкти. Розглянуто види моніторингу, цілі та завдання.

Іл.1. Бібліогр.: 9 найм.

UDK 519.81

Establish a regional monitoring as a means of implementing the concept of sustainable development of socio-economic systems / V.P. Pisklakova // Bionics of Intelligense: Sci. Mag. – 2011. – № 3 (77). – P. 78-84.

The work is devoted to the creation of the implementation of sustainable socio-economic systems. The problems encountered in achieving this goal, as well as their solutions. A prerequisite for developing such systems is the availability of complete, accurate information on the socio-economic targets. Considered monitoring activities, goals and objectives.

Fig. 1. Ref.:9 items.

УДК 004.932.2:004.93'1

В.А. Гороховатский[1], Т.В. Полякова[2]

[1]Харьковский институт банковского дела Университета банковского дела НБУ,
г. Харьков, Украина, e-mail:gorohovatsky-v@rambler.ru
[2]Харьковский национальный автомобильно-дорожный университет,
г. Харьков, Украина, e-mail:assiada@list.ru

ОПТИМАЛЬНЫЕ МЕТОДЫ СОПОСТАВЛЕНИЯ СТРУКТУРНЫХ ОПИСАНИЙ ВИДЕООБЪЕКТОВ

Обсуждаются вопросы применения оптимальных подходов при сопоставлении структурных описаний видео-объектов на примере венгерского метода для поиска максимального паросочетания в двудольном взвешенном графе. Результаты экспериментов с использованием признаков аффинных инвариантов, построенных на множестве координат характерных признаков изображения, подтверждают эффективность предложенных методов в плане более высокой достоверности распознавания по сравнению с традиционными методами на основе голосования.

РАСПОЗНАВАНИЕ ИЗОБРАЖЕНИЙ, ХАРАКТЕРНЫЕ ПРИЗНАКИ, СОПОСТАВЛЕНИЕ СТРУКТУРНЫХ ОПИСАНИЙ, ПАРОСОЧЕТАНИЕ, ВЕНГЕРСКИЙ МЕТОД, ВЗАИМНО-ОДНОЗНАЧНОЕ СООТВЕТСТВИЕ

Введение

В настоящее время для решения интеллектуальных задач компьютерного зрения, связанных с распознаванием объектов в условиях действия фона и помех, эффективны структурные технологии, основанные на анализе особенностей изображения в отдельных точках. Оценка свойств описания видео-объекта в виде конечного числа характерных признаков (ХП) позволяет за приемлемое время качественно разделить информацию об объекте и о фоновых образованиях [1-3]. Основой для формирования ХП служит информация, содержащаяся в некотором смысле «значимых» фрагментах. Распознавание на основе структурных описаний в виде множества ХП имеет следующие преимущества: достаточно простое в вычислительном плане формирование признаков, сжатое признаковое пространство с возможностью управления его размерностью в зависимости от задачи, устойчивость к влиянию фоновых искажений и ложных объектов.

В пространстве ХП распознавание (классификация) видео-объектов сводится к построению мер подобия конечных множеств на основе соответствий элементов. Эффективный способ реализации подобия – голосование ХП [2]. Наряду с устойчивостью принятия решения при неполном или избыточном представлении описаний голосование обладает такими важными достоинствами, как высокая вероятность правильного распознавания и простота построения, что делает его конкурентоспособным среди других подходов.

Особый интерес при сопоставлении конечных множеств признаков в задачах искусственного интеллекта представляет применение оптимальных подходов, в основе которых лежит оптимизация значения некоторого критерия (в данном случае величины подобия описаний) путем комбинаторного перебора на конечном множестве вариантов [4]. Одним из эффективных способов структурного анализа данных, связанных с вычислением оптимального сходства конечных множеств (описаний распознаваемых объектов), является венгерский метод (ВМ), названный комбинаторной оптимизацией ограниченных множеств. Его можно рассматривать как решение задачи о назначениях для наиболее общего случая поиска максимального паросочетания в двудольном взвешенном графе [5-7]. Суть венгерского метода состоит в последовательном формировании оптимального решения за конечное число шагов. Решение задачи о назначениях решает проблему установления степени сходства структурных описаний объектов в условиях несоответствия порядка их следования или нарушения целостности описания из-за помех. Одним из достоинств применения ВМ при сопоставлении описаний является возможность оценки близости промежуточной величины сходства к оптимальному варианту решения на каждой из итераций. Это позволяет контролировать процесс и при необходимости (в целях сокращения временных затрат) прекращать его при достижении заданной точности. Такое свойство особенно важно для задач большой размерности, т.к. число векторов ХП в описаниях достигает двух-трех сотен.

Принцип оптимального сопоставления структурных описаний по сравнению с известными решениями (SIFT, SURF [1]), основанными на голосовании ХП, позволяет реализовать схему построения однозначных соответствий ХП из сравниваемых описаний, что в целом улучшает достоверность распознавания.

Цель исследования – анализ эффективности применения оптимальных подходов на примере венгерского метода для сопоставления описаний в виде множеств ХП.

Задачи работы состоят в модернизации ВМ для сопоставления структурных описаний, сравнительной оценке сложности вычислительных затрат с другими методами, экспериментальных исследованиях оптимальных методов для конкретных систем признаков и баз видеоинформации.

1. Формализация метода оптимального сопоставления описаний

Представим описание U визуального объекта в виде конечного множества $U=\{u_i\}_{i=1}^{m}$ элементов u_i, где m – количество ХП в описании, u_i – ХП, который может представлять как дескриптор (значение признака), так и геометрическую информацию (координаты ХП, значения аффинных инвариантов на основе этих координат и т.д.) [3].

Критерием при сопоставлении элементов является значение некоторой метрики $\rho(u_i,u_j)$, оценивающей различия u_i,u_j, взятых из разных описаний $u_i\in U_1$, $u_j\in U_2$. Методы с использованием голосования определяют число (сумму) голосов элементов, исходя из величин $\rho(u_i,u_j)$ путем оптимизации на конечном множестве или через оценку с помощью некоторого порога. Оптимальные методы на основе $\rho(u_i,u_j)$ формируют некоторое оптимальное решение, например, минимизирующее сумму расстояний между описаниями U_1,U_2 с учетом однозначности соответствий элементов сравниваемых множеств.

Применим ВМ для оптимального установления величины соответствия между двумя структурными описаниями с учетом действия пространственных помех. Результатом выполнения ВМ является формирование максимального паросочетания для элементов двух множеств с минимизацией общей стоимости (весов), которую можно оценить в виде суммы расстояний между парами ХП из сравниваемых описаний. Сопоставление U_1,U_2 на основе ВМ формально сводится к решению оптимизационной задачи:

$$R(x)=\sum_{i=1}^{m}\sum_{j=1}^{n}\rho(u_i,u_j)x_{ij}\to\min, \qquad (1)$$

где $u_i\in U_1$, $u_j\in U_2$, m и n – количества точек в описаниях U_1,U_2; $x_{ij}\in\{0,1\}$ – бинарный признак, отражающий соответствие i-го и j-го элементов из U_1,U_2. Решение задачи (1) при ограничении на однозначность соответствия признаков из сопоставляемых множеств

$$\begin{cases}\sum_{j=1}^{m}x_{ij}=1\ \forall i=\overline{1,n},\\ \sum_{i=1}^{n}x_{ij}=1\ \forall j=\overline{1,m},\end{cases} \qquad (2)$$

минимизирует расстояние между U_1,U_2. Для прикладных задач, где необходима максимизация целевой функции в (1), переход к задаче (1) осуществляют путем вычитания значений $\rho(u_i,u_j)$ из максимального значения метрики.

Как оптимальный, так и традиционный (основанный на голосовании) подходы используют в качестве базовой информации матрицу расстояний

$$\mathrm{P}=\begin{bmatrix}\rho_{11} & \rho_{12} & \cdots & \rho_{1n}\\ \rho_{21} & \rho_{22} & \cdots & \rho_{2n}\\ \cdots & \cdots & \cdots & \cdots\\ \rho_{m1} & \rho_{m2} & \cdots & \rho_{mn}\end{bmatrix}$$

между всевозможными парами ХП $u_i\in U_1$, $u_j\in U_2$.

Реализация принципа однозначного голосования элементов u_i связана с анализом отдельных строк матрицы P на предмет формирования голоса, равного 0 или 1. По полученному набору соответствий (голосов) можно вычислить значение критерия подобия двух описаний в виде суммы $S_1=\sum_i L(i)$, где $L(i)$ – предикат, равный 1, если голос отдается (соответствие установлено).

Реализация ВМ предполагает целенаправленный выбор такой последовательности из $\{\rho_{ij}\}$, сумма значений которой $S_2=\sum\rho_{ij}$ будет минимальна на множестве возможных соответствий элементов описаний, и при этом из каждой строки и столбца матрицы P будет выбран только один элемент, что в результате обеспечивает выбор оптимального по критерию S_2 однозначного соответствия множеств элементов описаний (паросочетание).

Реализация ВМ осуществляется путем целенаправленного эквивалентного преобразования матрицы P с последовательным анализом строк и столбцов для получения матрицы с неотрицательными элементами и системой m независимых нулей, из которых никакие два не принадлежат одной и той же строке или одному и тому же столбцу. При достижении ситуации с m независимыми нулями проблема выбора считается решенной, оптимальный вариант назначений определяется позициями независимых нулей в преобразованной матрице [5]. Пример формирования паросочетания показан на рис. 1.

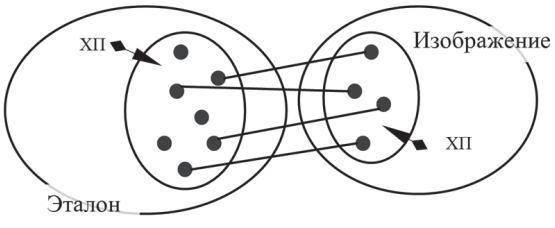


Рис. 1. Паросочетания для множеств ХП

Известно, что задача поиска паросочетаний с использованием ВМ решается за время, пропорциональное величине d^3, где d – это размер сравниваемых описаний. Видим, что требуемые затраты вычислений существенно возрастают с увеличением d. В то же время для метода с голосованием затраты прямо пропорциональны d, что не так критично к росту объема описаний.

При применении ВМ для сопоставления описаний U_1, U_2 их размерности m и n в общем случае могут отличаться. В этом случае в традиционном ВМ размер матрицы P увеличивают до максимального среди m и n, а возникшие элементы заполняют нулями [5-7]. Если исходная матрица не является квадратной, то дополнительно вводят необходимое количество строк (или столбцов), а их элементам присваивают значения, определяемые условиями задачи. Для задачи определения подобия описаний полученным за счет расширения элементам необходимо присвоить некоторое максимальное значение среди возможных значений $\rho(u_i, u_j)$. Заметим, что для нашей задачи нецелесообразно усекать матрицу весов до квадратной традиционным способом в виде $\min(m,n)$, т.к. при этом может быть потеряна ценная информация об видео-объекте. С другой стороны, дальнейшее усложнение обработки за счет применения ВМ требует минимизации размеров матрицы на этапе предварительной обработки. Это достигается посредством применения специальных методов сжатия структурных описаний [2]. Другим способом снижения затрат есть отбор в некотором смысле «значимых» строк и столбцов, например, содержащих большие по значению расстояния.

2. Геометрические признаки аффинных инвариантов

Особый интерес представляет применение оптимальных методов для геометрических признаков, построенных на основе координат ХП и отражающих пространственное расположение ХП объекта. Любое подмножество трех неколлинеарных точек объекта определяет аффинный базис, который задает на объекте систему координат. После выбора системы координат ХП можно представить с помощью аффинных инвариантов (АИ). После воздействия аффинного преобразования на объект значения АИ не изменяются [1]. Существуют различные способы пространственной группировки АИ для сопоставления описаний. В работе [3] для отдельного ХП формируется подмножество АИ путем рассмотрения всевозможных базисов (разные системы аффинных координат). Другим способом является формирование подмножества АИ для всех ХП в одной и той же системе координат, т.е. для отдельного базиса. Такой способ представления имеет вид значительного числа подмножеств (количество базисов) с небольшим количеством элементов. Данное представление удобно для применения ВМ из-за небольших по размеру матриц P. Кроме того, оно дает возможность управлять количеством подмножеств (базисов) и принимать решение по неполному описанию.

Рассмотрим множество АИ, отображающих геометрические свойства описания объекта и представленных в виде конечного множества числовых троек $\{\alpha_q\}_{q=1}^{r}$ [3], где r – число АИ. Расстояние между признаками α_q и α_p эталона и изображения для матрицы P можно вычислить как

$$\rho_{qp} = \sqrt{\sum_{i=1}^{3} (\alpha_{qi} - \alpha_{pi})^2} .$$

Взаимно-однозначное соответствие элементов описаний может быть установлено путем решения задачи поиска взвешенного паросочетания. Решение о принадлежности объекта конкретному эталону принимается на основе максимального числа установленных соответствий между подмножествами АИ или на основе минимума величины S_2.

3. Результаты экспериментов

В качестве критерия, характеризующего достоверность распознавания в конечной базе данных, рассмотрим величину $\theta = h_1 / h_2$, где h_2 – максимум гистограммы голосов, h_1 – ближайший к нему максимум. Значение критерия θ показывает, насколько уверенно осуществляется принятие решения на основе полученного максимума числа голосов, отданных за конкретный эталон. Чем меньше значение θ, тем в большей степени глобальное решение значимо по отношению к локальному, соответствующему наиболее близкому из остальных (возможно, ложных) классов.

Для базы объектов видео-данных Coil-20 [1] вычислено множество АИ $\{\alpha_q\}$. Эксперимент для традиционного подхода на основе голосования ХП состоял в том, что на вход системы распознавания поступало описание одного из эталонов, которое последовательно сравнивалось со всеми описаниями базы, в результате принималось решение в соответствии с максимумом нормированного количества голосов [2]. Для предложенного оптимального метода сопоставление осуществлялось по принципу «множество-множество» с достижением минимума в выражении (1) при условии однозначного соответствия элементов множеств. Для базы Coil-20 с применением структурных описаний в виде множеств АИ достигнуто безошибочное распознавание.

Сравнительные эксперименты показали, что в традиционном подходе величина θ для правильно найденного класса относительно других эталонов достигает значения $\theta = 0{,}8$, в то время как для оптимального метода на основе ВМ максимальное из значений $\theta = 0{,}16$. Это непосредственно доказывает, что достоверность распознавания для оптимального метода существенно лучше.

В то же время отметим, что для оптимального метода затраты времени на решение задачи распознавания несколько больше, чем для традиционного метода. При этом временные затраты значительно увеличиваются с увеличением размерности сравниваемых описаний, а применение аппарата АИ, как известно, по сравнению с обычным пред-

ставлением в виде множества ХП еще более усиливает временные различия. Например, для $m=6$ с использованием пространства АИ временные затраты оптимального метода при распознавании оказались в 1,5 раза выше, а при $m=13$ – в десятки раз больше. Отметим, что эти затраты могут регулироваться за счет процедуры предварительной обработки структурных описаний для конкретно решаемой прикладной задачи.

Выводы

Оптимальные методы для сопоставления структурных описаний объектов за счет более тщательного отбора соответствий обеспечивают лучшую достоверность распознавания по сравнению с традиционными подходами на основе голосования. Проведен анализ и исследование оптимального метода сопоставления на основе ВМ для описаний, представленных множеством аффинных инвариантов.

В целом полученные характеристики для оптимального метода могут быть оценены как предельно достижимые, в то время как традиционный подход с голосованием отдельных элементов описания можно считать некоторой практической аппроксимацией. Оптимальные методы целесообразно применять для задач, требующих наиболее тщательного анализа соответствия множеств описаний.

Впервые показано, что применение оптимальных методов определения максимального паросочетания в двудольном графе для задачи сопоставления структурных описаний видео-объектов повышает достоверность распознавания за счет однозначного и более точного учета подобия подмножеств элементов описаний. На примере геометрических признаков изображений получено экспериментальное подтверждение эффективности предложенных методов.

Практически важным является получение предпочтительных характеристик распознавания по сравнению с известными методами, что говорит о целесообразности развития и применения оптимальных методов сопоставления в задачах компьютерного зрения.

Перспективы исследования состоят в разработке подходов к построению мер подобия на основе структурного анализа подмножеств значений аффинных инвариантов, сгруппированных для отдельных базисов.

Список литературы: 1. *Шапиро, Л.* Компьютерное зрение [Текст] / Л. Шапиро, Дж. Стокман; пер. с англ.– М.: БИНОМ. Лаборатория знаний, 2006. – 752 с. **2.** Gorokhovatskiy V.A. Compression of Descriptions in the Structural Image Recognition / V.A. Gorokhovatskiy // Telecommunications and Radio Engineering. – 2011, Vol. 70, No 15. – P. 1363-1371. **3.** *Гороховатский, В.А.* Модели комплексированных мер подобия структурных описаний изображений [Текст] / В.А. Гороховатский, Т.В. Полякова, Е.П. Путятин // Реєстрація, зберігання і обробка даних. — 2011. — Т. 13, № 1. — С. 21–28. **4.** *Рассел, С.* Искусственный интеллект: современный подход; 2-е изд. [Текст]/ С. Рассел, П. Норвиг; пер. с англ.– М.: Изд. дом «Вильямс», 2006. – 1408 с. **5.** *Гольштейн, Е.Г.* Задачи линейного программирования транспортного типа [Текст] / Е.Г. Гольштейн, Д.Б. Юдин, - М.: Наука, 1969. – 382 с. **6.** *Сонькин, Д.М.* Адаптивный алгоритм распределения заказов на обслуживание автомобилями такси [Текст] / Д.М. Сонькин // Известия Томского политехнического университета. – 2009. – Т. 315, № 5. –С.65-69. **7.** *Пападимитриу, Х.* Комбинаторная оптимизация. Алгоритмы и сложность [Текст] / Х. Пападимитриу, К. Стайглиц // М.: Мир, 1985. – 512 с.

Поступила в редколлегию 19.05.2011

УДК 004.932.2:004.93’1

Оптимальні методи зіставлення структурних описів відеооб’єктів / В.О. Гороховатський, Т.В. Полякова // Біоніка інтелекту: наук.-техн. журнал. – 2011. – № 3 (77). – С. 85-88.

Обговорюються питання застосування оптимальних підходів при зіставленні структурних описів відеооб’єктів на прикладі угорського методу для пошуку максимального паросполучення у дводольному зваженому графі. Результати експериментів з використанням ознак аффінних інваріантів, побудованих на множині координат характерних ознак зображення, підтверджують ефективність запропонованих методів в плані більш високої достовірності розпізнавання в порівнянні з традиційними методами на основі голосування.

Бібліогр.: 7 найм.

UDC 004.932.2:004.93’1

Optimal methods of structural descriptions of video objects comparison / V.O. Gorokhovatsky, T.V. Polyakova // Bionics of Intelligense: Sci. Mag. – 2011. – № 3 (77). – P. 85-88.

The application of optimal approaches for structural descriptions of video-objects comparison are discussed, e.g. Hungarian method for finding maximum matching in the bipartite weighted graph. Experimental results based on affine invariant features which are built on coordinate set of image characteristic features confirm efficiency of proposed methods at higher recognition rate in comparison with traditional voting methods.

Ref.: 7 items.

УДК 004.89

М.М. Зацеркляний, А.Л. Єрохін, А.С. Бабій, О.П. Турута
Харківський національний університет внутрішніх справ,
м. Харків, Україна, ayerokhin@ukr.net

РОЗРОБКА МЕТОДУ ВИЯВЛЕННЯ СЕЗОННИХ КОЛИВАНЬ З ЗАСТОСУВАННЯМ НЕЧІТКОГО ЗГЛАДЖУВАННЯ НА БАЗІ F-ПЕРЕТВОРЕННЯ

У статті на основі методики виявлення сезонної складової запропонована модифікація підходу до виявлення тренда, яка полягає у використанні методу згладжування на базі нечіткого перетворення (F-перетворення). За основу покладена адитивна модель подання динамічного ряду. У методі попередньо виділяється тренд, а потім сезонний компонент. Тренд виділяється за допомогою нечіткого згладжування.

МОДЕЛЮВАННЯ НЕСТАЦІОНАРНИХ ПРОЦЕССІВ, СЕЗОННІ КОЛИВАННЯ, ЗГЛАДЖУВАННЯ ДИНАМІЧНОГО РЯДУ

Вступ

Використання сучасних методів та засобів штучного інтелекту для ефективного вирішення практичних задач кримінології є надзвичайно важливим. У 1988 році в рамках ООН створена інформаційна система в області боротьби зі злочинністю та розміщена в мережі Internet. Міжнародні конгреси ООН із попередження злочинності і поведінки з правопорушниками рекомендують державам вжити заходи щодо комп'ютеризації кримінального правосуддя. Ці рекомендації затверджені й Генеральною Асамблеєю ООН. Подальший розвиток інформатизації правозастосовної діяльності передбачений у резолюції 1993/34 Економічної і Соціальної Ради ООН.

Для виконання названої резолюції Секретаріат ООН приступив до вивчення питання про розвиток і поширення інформаційної мережі та створення міжнародного інформаційно-аналітичного центру з досить широкими функціями. Кінцевою метою роботи такого центру стане зміцнення потенціалу національних держав у питаннях одержання, збереження, опрацювання, аналізу та поширення електронної інформації із області попередження злочинності і кримінального правосуддя.

Добре розроблене законодавство та система заходів попередження при їх невиконанні не забезпечують необхідного рівня законності. Для цього потрібний комплекс умов організаційного характеру – контроль, нагляд, перевірка виконання тощо, де є можливості для широкого застосування інформаційних технологій.

Забезпечення структур управління, до яких належать і органи внутрішніх справ, належною інформацією – умова ефективної діяльності. Вимоги відбору необхідної інформації відносяться до всіх її форм, але в першу чергу – до статистичної, що є джерелом відомостей, які суттєво впливають на управлінський процес.

Досить часто фактор сезонності є достатньо важливим під час прийняття рішення про проведення профілактичних заходів або протидії злочинності. У випадку, якщо суспільне явище має багато чинників, або носить комплексний характер, необхідні статистичні методи виявлення сезонності.

1. Аналіз літератури

Питанням аналізу та прогнозування злочинності, застосування методів штучного інтелекту для задач кримінології присвячені роботи [1-6]. Методи прогнозування можуть ґрунтуватися на штучних нейронних мережах, причому нейронні мережі в таких задачах виступають в якості універсального апроксиматора даних [2-6], тобто у процесі навчання підбираються функціональні коефіцієнти, а розрахунок значення апроксимаційної функції для заданих вхідних даних відбувається на етапі функціонування.

Методи прогнозування можуть ґрунтуватися й на нечітких множинах. Основні принципи функціонування та підходи до застосування нечітких множин для прогнозування розглянуті в [7]. Підходи, що спираються на моделі виведення Мамдані-Заде [8] і TSK [9], дозволяють описати досліджуване явище у вигляді нелінійної функції вхідних змінних x_i (i = 1,2,...,N) і параметрів нечіткої системи. В [10] відзначено, що ці підходи дають можливість апроксимувати з довільною точністю довільну нелінійну функцію багатьох змінних сумою нечітких функцій однієї змінної.

В роботах [11,12] було розглянуто застосування методу нечіткого згладжування на основі F-перетворень для аналізу даних, в тому числі для виявлення трендової складової. В роботі [13] було розглянуто застосування методики Четверякова для аналізу сезонних коливань динамічного ряду.

В даній статті пропонується модифікація методики визначення сезонних коливань злочинності шляхом застосування нечіткого згладжування на етапі виявлення тренду.

2. Огляд методів аналізу та прогнозування числових рядів

Використання сучасних інформаційних технологій дозволяє суттєво впливати на недопущення

посадових правопорушень, адже фіксація факту перевищення влади чи службових повноважень у процесі припинення злочинів чи інших неправових дій працівників дозволяє вести серйозну профілактичну роботу з кадрами. Створення умов для розгляду кожного факту правопорушення з боку працівника правоохоронного блоку є основою для притягнення його до юридичної відповідальності за свої дії.

Вимоги відбору необхідної інформації відносяться до всіх її форм, але в першу чергу – до статистичної. Неповне відображення стану злочинності в статистичних обліках, незважаючи на наявність інформації про невраховані злочини (латентну злочинність), пов'язано з випадками навмисного приховування злочинів від обліку, які зустрічаються на практиці.

Негативний ефект від приховування злочинів проявляється двояко: воно робить інформацію необ'єктивною і тим самим перешкоджає забезпеченню її повноти про стан злочинності, а також є тяжким порушенням законності.

Отже, значну роль в реалізації "Комплексної цільової програми боротьби зі злочинністю" відіграє система інформаційного забезпечення, яка здійснює інформаційну підтримку органів внутрішніх справ у розкритті, розслідуванні та попередженні злочинів, у процесі встановлення та розшуку злочинців, надає багатоцільову статистичну, аналітичну та довідкову інформацію. Тому в останні роки в Україні спостерігається інтенсивне впровадження сучасних інформаційних технологій, і особливо методів і засобі штучного інтелекту, у діяльність правоохоронних органів. Інформація, зафіксована на основі офіційного обліку або шляхом спеціально організованого дослідження у статистичних картках, журналах обліку, базах даних, комп'ютерних інформаційних системах, в анкетах опитування громадян, при вивченні кримінальних, адміністративних, цивільних справ та інших матеріалів, є розрізненими "горами даних" про одиниці досліджуваної сукупності. Ця інформація стає дієвим засобом боротьби зі злочинністю тільки у випадку її аналітичного опрацювання, розвиток якого на сьогодні не є задовільним.

Основний вид ресурсу в організаційних системах, до яких належать органи внутрішніх справ, – люди, які забезпечують виконання задач, що стоять перед цією системою.

Прогнозування кадрових ресурсів, зокрема потреби різних служб і підрозділів органів внутрішніх справ у фахівцях відповідної кваліфікації, завжди було однією з найважливіших управлінських задач у системі МВС. Особливу актуальність ця проблема здобуває зараз в умовах децентралізації управління, підвищення самостійності низових органів у розв'язуванні організаційно-штатних задач.

Крім кримінологічних прогнозів при обґрунтуванні потреби органів внутрішніх справ у кадрах необхідно враховувати сучасні соціально-економічні, політичні і демографічні процеси, оскільки вони не тільки визначають кримінальну обстановку в країні, а й значно впливають на умови комплектування ОВС.

Таким чином, в основі розрахунку потреби в кадрах знаходиться комплекс прогнозних оцінок різних факторів, які впливають як на стан злочинності, так і на умови комплектування ОВС кваліфікованими кадрами.

Питаннями оцінювання злочинності займалися вчені різних країн світу протягом останніх декількох століть.

Вивчення структурних показників у часі дає можливість виявити реальні тенденції складових частин юридично значущих явищ, спираючись на які можна розв'язувати і прогностичні проблеми. Важливе значення при аналізі злочинності має її інтенсивність. Інтенсивність злочинності є складним якіснокількісним параметром кримінологічної обстановки в країні, регіоні, районі чи населеному пункті, який вказує на рівень злочинних проявів, темпи їх зростання чи міру суспільної небезпеки (ваги).

Проблеми математичного моделювання нестаціонарних процесів, до яких відносяться соціальні процеси, в тому числі злочинність, стають все більш актуальними завдяки можливості розвитку комп'ютерних інформаційних технологій, зокрема систем підтримки прийняття рішень (СППР), які інтегрують у собі сучасні методи взаємодії комп'ютер–особа, що приймає рішення, моделі та методи прийняття рішень, методи прогнозування та менеджменту процесів різної природи.

Оцінювання злочинності неможливо проводити без врахування економічного, соціального, демографічного стану громади.

Прийняття рішень при оцінюванні злочинності, необхідно відрізняти від прийняття рішень при протидії злочинності. Оцінювання є одним з етапів протидії злочинності.

Під оцінюванням злочинності розуміється порівняння рівнів зареєстрованої злочинності серед громад, подібних за економічними, соціальними та демографічними показниками, та вибір адекватної моделі розвитку злочинності для подальшого прийняття рішень, пов'язаних з протидією злочинності для даної групи підрозділів або зменшенню рівнів латентності.

Створюючи СППР як сукупність процедур з обробки даних та суджень, що допомагають керівнику в прийнятті рішень, що ґрунтуються на використанні моделей, будемо враховувати наступні специфічні риси:

– можливість вироблення варіантів рішення в спеціальних, неочікуваних для ОПР ситуаціях;

– можливість моделей, що застосовуються в системах, адаптуватися до конкретної, специфічної реальності в результаті діалога з користувачем;

– можливість системи інтерактивного генерування моделей.

Все це поєднується із зручним поданням результатів обчислень альтернативних варіантів прийняття рішень, адаптивним дружнім інтерфейсом, базою знань і даних.

Головною особливістю інформаційної технології підтримки ухвалення рішень є якісно новий метод організації взаємодії людини і комп'ютера. Вироблення рішень, що є основною метою цієї технології, відбувається в результаті ітераційного процесу, в якому беруть участь:

– система підтримки ухвалення рішень у ролі обчислювальної ланки і об'єкту управління;

– людина як управляюча ланка, що задає вхідні дані і оцінює одержаний результат обчислень на комп'ютері.

Завершення ітераційного процесу відбувається за вказівкою людини. В цьому випадку можна говорити про здатність інформаційної системи спільно з користувачем створювати нову інформацію для ухвалення рішень.

На сьогоднішній день в теорії ухвалення рішень широко відомі такі методи:

1. *Методи теорії корисності.* Теорія корисності носить аксиометричний характер. Згідно з цими методами, якщо переваги людей відносно певних ігор задовольняють ряду аксіом, то їх поведінка може розглядатися як максимізація очікуваної корисності. Д. Седвіж розробив аксіоматичну теорію, яка дозволяє одночасно вимірювати корисність і суб'єктивну ймовірність. Це знайшло відображення в моделі суб'єктивної очікуваної корисності, де ймовірність визначається як міра впевненості в здійсненні тієї чи іншої події. Модель знайшла широке використання серед економістів і розглядається ними як обгрунтований засіб вибору якнайкращих рішень.

Суть методу дерев рішень полягає в розбитті задачі на ряд підзадач, а ті, в свою чергу, на інші підзадачі, і так далі. В результаті основна задача подається у вигляді дерева рішень. В частині вершин дерева рішень вибір здійснюється безпосередньо особою, що приймає рішення, в іншій частині – на основі суб'єктивної ймовірності настання подій. Метод дерев рішень дозволяє особі, що приймає рішення, визначити оптимальну послідовність дій (стратегію) з урахуванням особистих оцінок і переваг.

В основу багатокритерійної теорії корисності (МТП) покладається припущення, що варіанти рішень мають оцінки за багатьма критеріями. Причому при виконанні умови строгої умовної незалежності по корисності, функція корисності має або адитивний, або мультиплікативний вигляд.

2. *Методи теорії проспектів* (ТП). Проспект – це гра з результатами ймовірності. В методах теорії проспектів враховується 3 поведінкові ефекти:

– ефект визначеності – тенденція надавати більшу вагу детермінованим результатам;

– ефект віддзеркалення – до вимірювання переваг при переході від виграшів до втрат;

– ефект ізоляції – тенденція до спрощення вибору шляхом виключення загальних компонентів варіантів рішення.

Методи теорії проспектів мають аксіоматичні основи. Недоліком є те, що даний метод не знімає усіх проблем, які виникають при вивченні поведінки людей в задачах вибору рішень.

3. *Методи ELECTRE.* В методах ELECTRE пропонується конструктивний підхід до вироблення рішень, в рамках якого методи, моделі і концепції розглядаються як допоміжні засоби практичного аналізу ситуації. Ці засоби дозволяють з'ясувати цілі ухвалення рішень і краще зрозуміти переваги особи, що приймає рішення. Недоліком методів ELECTRE є те, що вони є допоміжними засобами, а не способом вироблення кращого рішення як при аксіоматичному підході.

4. *Метод аналізу ієрархій.* В цьому методі використовується дерево критеріїв, в якому загальні критерії діляться на критерії окремого характеру. Для кожної групи критеріїв визначаються коефіцієнти важливості. Альтернативи також порівнюються між собою за окремими критеріями з метою визначення кожної з них. Засобом визначення коефіцієнтів важливості критеріїв або критерійної цінності альтернатив є попарне порівняння. Результат порівняння оцінюється за бальною шкалою. На основі таких порівнянь обчислюються коефіцієнти важливості критеріїв, оцінки альтернатив і знаходиться загальна оцінка як зважена сума оцінок критеріїв.

5. *Евристичні методи.* До евристичних методів відносяться такі методи:

– метод зваженої суми оцінок критеріїв. Кожній альтернативі дається числова (бальна) оцінка за кожним із критеріїв. Критеріям приписується кількісна вага, що характеризує їх порівняльну важливість. Вага множиться на критерійні оцінки, одержані числа сумуються – так визначається цінність альтернативи. Далі вибирається альтернатива з найбільшим показником цінності;

– метод компенсації. Даний метод використовується при попарному порівнянні альтернатив.

3. Розробка методу виявлення сезонних коливань

Сезонні коливання властиві абсолютній більшості юридично значущих явищ. Сезонність зв'язується, як правило, зі зміною природно-кліматичних умов у рамках обмеженого проміжку часу – річного періоду. Найбільш яскраво цей зв'язок видно там, де досліджувані процеси прямо пов'язані з природними особливостями тієї чи іншої пори року. Проте сезонні коливання формуються не тільки під впливом природно-кліматичних чинників, а й, нехай меншою мірою, під впливом інших особливостей досліджуваної системи.

Не в усіх випадках сезонність є наслідком дії некерованих або майже некерованих факторів. Частіше вони піддаються регулюванню. Та навіть і в тих випадках, коли прямий вплив на процеси, які викликають сезонні коливання, неможливий, необхідно враховувати їх дію при вдосконалюванні відповідних процесів і процесів управління. Для цілеспрямованого впливу на сезонність необхідно вміти вимірювати та аналізувати сезонність, уміти передбачати розвиток процесів, залежних від сезонних коливанн.

Розглядаємо часовий ряд:

$$x_i = U_i + V_i + \varepsilon_i, i = 1,...., N, \quad (1)$$

де U_i – тренд; V_i – сезонний компонент, ε_i – випадковий компонент, N – число рівнів спостереження.

Щодо U_i передбачається, що це деяка гладка функція, міра гладкості якої заздалегідь невідома. Під мірою гладкості тренда розуміється мінімальний ступінь полінома, який адекватно згладжує компонент U_i. Сезонний компонент V_i має період T_0: $V_{i+T_0} = V_i$ (T_0=12 для ряду місячних даних; T_0=4 – для ряду квартальних даних).

Крім того, відомо, що $N=mT_0$, m – ціле число. Очевидно, коли T_0 – число місяців або кварталів у році, то m – число років, поданих у часовому ряді $\{x_i\}$. Початкові дані часового ряду можна подати у вигляді матриці $\{x_{ij}\}$ розміру $[m\times T_0]$. У цьому випадку вираз (1) записується так:

$$x_{i,j} = U_{i,j} + V_{i,j} + \varepsilon_{i,j}, i = \overline{1,m}, j = \overline{1,T_0}. \quad (2)$$

Аналіз сезонності полягає у дослідженні сезонних коливань та у дослідженні того зовнішнього циклічного механізму, який їх зумовлює. Для дослідження сезонних коливань поза зв'язком із причинами, які їх породжують, необхідно відфільтрувати із часового ряду $\{x_i\}$ сезонний компонент V_i і проаналізувати його динаміку.

В методах фільтрації попередньо виділяється тренд, а потім сезонний компонент. Тренд у чистому вигляді необхідний і для аналізу динаміки сезонної хвилі.

Розглянемо насамперед деякі теоретичні питання виявлення і фільтрації сезонного компонента динамічного ряду. Як і раніше, будемо розглядати динамічні ряди, породжувані адитивним випадковим процесом (1).

Основна ідея ітераційних процедур виявлення тренду динамічного ряду полягає в багатократному застосуванні ковзної середньої:

$$\bar{x}_i = \frac{\frac{x_{i-T_0/2}}{2} + x_{i-T_0/2+1} + \cdots + x_i + \cdots + x_{i+T_0/2+1} + \frac{x_{i+T_0/2}}{2}}{T_0} = \frac{\sum_{\tau=-p}^{p} \alpha_\tau x_{i+\tau}}{T_0} \quad (3)$$

і оцінці сезонного компонента на кожній ітерації.

При переході від однієї ітерації до іншої може відбуватися зміна довжини ділянки ковзання T_0 і закону зміни вагових коефіцієнтів α_τ.

Для виявлення тренду в динамічних рядах злочинності пропонується модифікація, а саме: використовувати нечітке згладжування динамічного ряду на основі нечіткого перетворення (F-перетворення).

Цей метод реалізується наступним алгоритмом [10, 11]:

1. Для виявлення тренду скористаємося підходом, описаним в [14], що базується на використанні нечіткого згладжування динамічного ряду на основі нечіткого перетворення (F-перетворення).

Нехай існує функція f, значення якої відомі в точках $x_1 ... x_N \in w$.

Розділимо динамічний ряд (w – інтервал динамічного ряду) на множину рівновіддалених вузлів

$$t_k = \upsilon_L + h(k-1), k = 1,...n,$$

де $N > n, h = \frac{\upsilon_R - \upsilon_L}{n-1}$. Визначимо n базисних функцій $A_1 ... A_n$, що покривають весь інтервал динамічного ряду та відповідають таким умовам:

1. A_k – неперервна,
2. A_k – монотонно зростає на $[t_{k-1}, t_k]$ і монотонно спадає на $[t_k, t_{k+1}]$,
3. $A_k : w \to [0,1], A_k(t_k) = 1$,
4. $A_k(t) = 0$, якщо $t \notin (t_{k-1}, t_{k+1})$, де вважаємо, що $t_0 = t_1 = \upsilon_L, t_{n+1} = t_n = \upsilon_R$,
5. $\sum A_k(t) = 1$ для всіх $t \in w$.

Для даного випадку скористаємося базисною функцією

$$A_k(t) = \begin{cases} \frac{t - t_{k-1}}{t_k - t_{k-1}}, \text{якщо}: t_{k-1} \le t \le t_k \\ \frac{t_{k+1} - t}{t_{k+1} - t_k}, \text{якщо}: t_k \le t \le t_{k+1} \\ 0, \text{в іншому випадку} \end{cases} \quad (4)$$

Розділимо інтервал w на n нечітких областей. Використовуючи базисні фунції, ми перетворимо дану функцію f в кортеж з n дійсних чисел $[U_1,...,U_n]$, що визначені

$$U_k = \frac{\sum f(t_i) A_k(t_i)}{\sum A_k(t_i)}, k = 1,...,n \quad (5)$$

F – компоненти (U_k) представляють собою тренд динамічного ряду. Для отримання значень тренду в точках, де відсутні значення F-компонентів після згладження, скористаємося в даному випадку лінійною інтерполяцією між двома найближчими точками з відомими значеннями:

Нехай відомі значення U_{d0} та U_{d1}. Тоді відшукуване значення $U_i^{(1)}$, де $d0 < i < d1$ можна знайти за такою формулою:

$$U_i^{(1)} = U_{d0} + \frac{U_{d1} - U_{d0}}{d1 - d0}(i - d0).$$

У результаті дістаємо попередню оцінку тренду $U_i^{(1)}$ і відхилення емпіричного ряду від вирівняного $l_i' = l_i = x_i - U_i^{(1)}$, або $l'_{ij} = x_{ij} - U_{ij}^{(1)}$, $(i=\overline{1,m}\,;\, j=\overline{1,T_0})$.

2. Для кожного року i обчислюється σ_i – середнє квадратичне відхилення:

$$\sigma_i = \sqrt{\frac{\sum_{j=1}^{T_0} l_{ij}'^2 - \frac{\left(\sum_{j=1}^{T_0} l_{ij}'\right)^2}{T_0}}{T_0 - 1}}, \qquad (6)$$

на яке діляться окремі місячні (квартальні) відхилення відповідного року:

$$\bar{l}_{ij} = \frac{l_{ij}'}{\sigma_i}. \qquad (7)$$

3. З нормованих таким чином відхилень обчислюється в першому наближенні середня сезонна хвиля:

$$V_j^{(1)} = \frac{\sum_{i=1}^{m} \bar{l}_{ij}}{m}. \qquad (8)$$

4. Середня сезонна хвиля множиться на середнє квадратичне відхилення кожного року і віднімається від рівнів початкового емпіричного ряду:

$$\overline{=}\, x_{ij} = x_{ij} - V_j^{(1)} \cdot \sigma_i. \qquad (9)$$

Утворений таким чином ряд позбавлений сезонної хвилі, одержаної у першому наближенні.

5. Цей ряд знову піддається нечіткому згладжуванню на основі F-перетворення (для місячних даних по п'яти або семи точках у залежності від інтенсивності дрібних коливань і тривалості більш значних). В результаті одержується нова оцінка тренду $U_i^{(2)}$.

6. Відхилення початкового емпіричного ряду $\{x_i\}$ від ряду $\{U_i^{(2)}\}$, одержаного в п. 5,

$$l_i^{(2)} = yi^{U_i^{(2)}} \qquad (10)$$

знову піддаються аналогічному опрацюванню згідно з пунктами 2 і 3 для виявлення наступного наближення середньої сезонної хвилі.

Висновки

В роботі проведено аналіз підходів, які присвячені аналізу та прогнозуванню злочинності та застосуванню систем і засобів штучного інтелекту до задач кримінології.

В роботі отримав подальшого розвитку метод виявлення сезонної складової. На основі розглянутої методики запропоновано модифікацію підходу до виявлення тренду, а саме використання методу згладжування на базі нечіткого перетворення (F-перетворення), що дозволяє зменшити вплив аномальних рівнів.

Розроблений метод в подальшому може використовуватися для оцінювання сезонності соціальних явищ.

Застосування саме нечіткої моделі для згладжування ряду дозволяє зменшити вплив аномальних рівнів, пов'язаних з короткостроковим впливом невідомого фактору.

Список литературы: 1. *Яковлев, С.В.* Анализ и прогнозирование показателей наркологической статистики в Украине и в Харьковской области [текст] / С.В. Яковлев, Ю.В. Гнусов // Молодёжь и наркотики (социология наркотизма). – Харьков: Торсинг, 2000. – С.194–221. **2.** *Ерохин, А.Л.* Использование нейросетевых методов для прогнозирования временных рядов [текст] / А.Л. Ерохин, Ю.В. Гнусов // Искусственный интеллект. – 2002. – №4. – С.686-691. **3.** *Specht, D.* A General Regression Neural Network. IEEE Trans. on Neural Networks, Nov. 1991, №2/6, P.568-576. **4.** *Burrascano, P.* Learning Vector Quantization for the Probabilistic Neural Network. IEEE Trans. on Neural Networks, July 1991, 2, P.458-461. **5.** *Caudill, M.* The Kohonen Model. Neural Network Primer. AI Expert, 1990. – P.25-31. **6.** *Hong D.H.* Fuzzy linear regression analysis for fuzzy input-output data using shape preventing operations, Fuzzy Sets and Systs., vol. 27, pp. 275-289, 1988. **7.** *Zade L.A.* Fuzzy sets // Information and Control, 1965. – P.338-353. **8.** *Takagi T., Sugeno M.* Fuzzy identificationof systems and it's application to modeling and control // IEEE Trans. SMC, 1985. – P.116-132. **9.** *Widrow, B., Hoff M. E.* "Adaptive switching circuits," 1960IRE WESCON Convention Record, New York IRE, pp. 96-104, 1960. **10.** *Widrow, B., Sterns S.D.* Adaptive Signal Processing, New York: Prentice-Hall, 1985. **11.** *Perfilieva, I., Novak,V., Pavliska,V., Dvorak,A. and Stepnicka, M.* "Analysis and Prediction of Time Series Using Fuzzy Transform", Proc. IEEE World Congresson Computational Intelligence, Hong Kong, 2008, pp.3875–3879. **12.** *Perfilieva, I., Valasek, R.* "Fuzzy Transforms in Removing Noise" in: Computational Intelligence, Theory and Applications (Advancesin Soft Computing), Edited by B. Reusch, Berlin: Springer, pp.221–230, 2005. **13.** *Зацеркляний, М.М.* Автоматизація аналізу сезоних коливань рівня злочинності [текст] / М.М. Зацеркляний, А.С. Бабій // Право і безпека. – 2005, №4'3. **14.** *Романов, А.А.* Преобразования для прогноза компоненты векторного тренда и числового представления временного ряда [текст] // Известия Самарского научного центра Российской академии наук. Т.12, №4(2), 2010. – С. 492-497.

Надійшла до редколегії 14.06.2011

УДК 004.89

Разработка метода выявления сезонных колебания с применение нечеткого сглаживания на базе F-преобразования / Н.М. Зацеркляный, А.Л. Ерохин, А.С. Бабий, А.П. Турута // Бионика интеллекта: научн.-техн. журнал. – 2011. – № 3 (77). – С. 90-93.

В статье на основе методики выявления сезонной составляющей предложена модификация подхода к выявлению тренда, а именно использование метода сглаживания на базе нечеткого преобразования (F-преобразование).

Библиогр.: 14 назв.

UDC 004.89

Developing a method to identify seasonal variations using fuzzy transform / M.M. Zacerklyaniy A.L.Yerokhin, A.S. Babiy, O.P.Turuta // Bionics of Intelligense: Sci. Mag. – 2011. – № 3 (77). – P. 90-93.

The paper-based methodology for the identification of the seasonal component proposed modification approach to identify a trend, namely to use a smoothing technique based on fuzzy transformation (F-transformation)

Ref.: 14 items.

УДК: 004.032.26:519.174.2

А.В. Шкловец, Н.Г. Аксак

Харьковский национальный университет радиоэлектроники, г. Харьков, Украина

ПОСТРОЕНИЕ ЧЕТЫРЁХУГОЛЬНЫХ КАРТ КОХОНЕНА НА ОСНОВЕ ТРИАНГУЛЯЦИИ ДЕЛОНЕ ДЛЯ ВИЗУАЛИЗАЦИИ МНОГОМЕРНЫХ ДАННЫХ

Для уменьшения вычислительных затрат и упрощения алгоритма аппроксимации треугольных карт Кохонена кубической параметрической сплайн-поверхностью в работе предложен метод построения четырехугольных карт Кохонена. Аппроксимация карт Кохонена сплайн-поверхностью производится для увеличения точности визуализации многомерных данных.

ТРИАНГУЛЯЦИЯ ДЕЛОНЕ, КАРТЫ КОХОНЕНА, ПАРАМЕТРИЧЕСКАЯ СПЛАЙН-ПОВЕРХНОСТЬ, ОСТОВНОЙ ЛЕС

Введение

При решении задачи многомерной визуализации данных с помощью двумерных самоорганизующихся карт Кохонена, представленных в виде триангуляции Делоне, возникает проблема отображения данных на грани или в вершины треугольников карты. В результате будет искажена структура данных, а часть данных может стать неразличимыми на карте [1], что приводит к снижению точности визуализации данных. Проблема вызвана кусочно-плоской структурой карты. В работах [2-4] предлагается аппроксимировать карту параметрической кубической сплайн-поверхностью и отображать данные на кусочно-гладкие карты Кохонена. Однако аппроксимация карт Кохонена, представленных в виде триангуляции, требует больших вычислительных затрат.

1. Постановка задачи

Пусть после обучения карты Кохонена в n-мерном евклидовом пространстве множество выходных нейронов $W=\{W^1,...,W^N\}$ имеет координаты $W^l=\left(w_1^l,...,w_n^l\right)^T$, $l=\overline{1,N}$, N – количество выходных нейронов. На множестве W задана триангуляция Делоне $T=\left\{T^1,...,T^M\right\}$, где $T^k=\left\{T_1^k,T_2^k,T_3^k\right\}$, $k=\overline{1,M}$, M – количество треугольников, T_p^k – номер нейрона в множестве W, $T_p^k\in\{1,N\}$, $p=\overline{1,3}$ (рис. 1) [5].

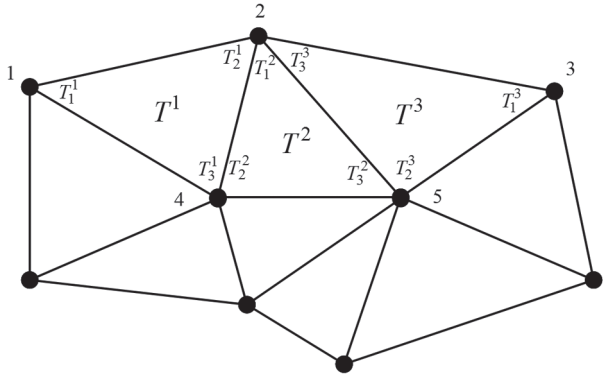


Рис. 1. Плоское изображение карты Кохонена в виде триангуляции Делоне

В работе [4] предлагается метод построения параметрической сплайн-поверхности на кусочно-плоской карте Кохонена. Идея построения заключается в сопоставлении каждому треугольнику T^k произвольной кубической поверхности $S^k(t_1,t_2)$, ограниченной треугольником с вершинами в точках $\{(0,0),(1,0),(0,1)\}$. Сплайн-поверхность будет построена, если все параметры поверхностей $S^k(t_1,t_2)$ $k=\overline{1,M}$ будут определены. Для этого формируется система линейных алгебраических уравнений относительно параметров параметрической сплайн-поверхности, представляющая собой:

1) условие прохождения параметрической сплайн-поверхности в точках $\{(0,0),(1,0),(0,1)\}$ через точки вершины треугольника T^k;

2) условия гладкого соприкосновения поверхностей $S^{k_1}(t_1,t_2)$ и $S^{k_2}(t_1,t_2)$, если соответствующие им треугольники имеют общую сторону.

Если неэквивалентных уравнений системы больше, чем переменных, то часть уравнений удаляется, что ведет к снижению гладкости сплайн-поверхности. Иначе выбирается некоторое количество переменных системы, через которые выражаются оставшиеся, а первоначально выбранные определяются из дополнительного условия оптимальности (например, условия минимального отклонения сплайн поверхности от кусочно-плоской карты).

Одной из проблем построения сплайн-поверхностей на триангуляции является формирование условия гладкого соприкосновения двух поверхностей по стороне $\{(1,0),(0,1)\}$. Чтобы описать данный случай, требуется составить большое количество уравнений, включающих в себя огромное количество переменных.

2. Метод построения четырехугольных карт Кохонена

Для устранения этой проблемы предлагается в триангуляции попарно объединить треугольники в четырёхугольники.

Необходимым условием для существования четырёхугольных карт является чётное количество треугольников в триангуляции.

Поставим в соответствие триангуляции T неориентированный граф $G=\left(\tilde{T},U\right)$, где $\tilde{T}=\left(\tilde{T}^1,...,\tilde{T}^M\right)$ – множество вершин и $U=\left(u^1,...,u^{N_u}\right)$ – множество ребер графа, N_u – общее количество ребер (рис. 2.) по следующим правилам:

1) каждому треугольнику $T^k\in T$ соответствует вершина $\tilde{T}^k\in G$, $k=\overline{1,M}$;

2) если два треугольника $T^p\in T$ и $T^q\in T$ имеют общую сторону, то вершины $\tilde{T}^p\in G$ и $\tilde{T}^q\in G$ объединены ребром $u^i=\left(\tilde{T}^p,\tilde{T}^q\right)$.

Граф G обладает следующими свойствами:

1) из каждой вершины графа G выходит не более трёх ребер, так как треугольник имеет не более трёх сторон, имеющих общую сторону с другими треугольниками;

2) всегда существуют вершины, имеющие менее трёх смежных ребер. Этим вершинам соответствуют треугольники, образующие границу триангуляции.

Задача построения четырехугольной карты эквивалента задаче построения остового леса $O_l\in O$ графа G, где O – множество всех остовных лесов, $i\in N$. В остовном лесе все вершины попарно соединены, при этом из каждой вершины выходит одно и только одно ребро. Построение всех остовных лесов O схоже с построением всех остовых деревьев графа G [6].

Построение всех остовых лесов O заключается в построении графа D, вершинами которого являются ребра $\left\{u^i\middle|u^i\in G,u^i\in O_l\right\}$ из графа G, принадлежащие остовому лесу O_l. Процедура построения позволяет найти все остовые леса O_l графа G, образованные объединением ребер в вершинах ветвей графа D. На рис. 3 приведен пример построения графа D для графа G (рис. 2*б*), а на рис. 4 представлен процесс построения остовного леса на каждой итерации.

Процедура построения остовных лесов O.

1) Установим вершину графа D с пустым множеством ребер графа G.

2) Найдем среди всех вершин графа G такие вершины $\tilde{T}^h\in G$, из которых выходит одно и только одно ребро $u^h\in U$. Эти ребра принадлежат остовному лесу ($u^h\in O_l$) и помещаются в вершину графа D.

3) Если таких вершин в графе G нет, выбираем любую вершину $\tilde{T}^i\in G$, имеющую два смежных ребра $u^i,u^j\in U$ (по свойству 2 графа G такая вершина существует) и рассмотрим два случая принадлежности остовому лесу ребер $u^i\in O_l$ и $u^j\in O_l$. При этом имеем раздвоение графа D на две ветви. В каждую новую вершину графа D помещаем рассматриваемое ребро.

4) Удаляем из графа G

а) ребра, принадлежащие остовому лесу,

б) вершины, в которые входят удаленные ребра,

в) все ребра, которые выходят из удаленных вершин.

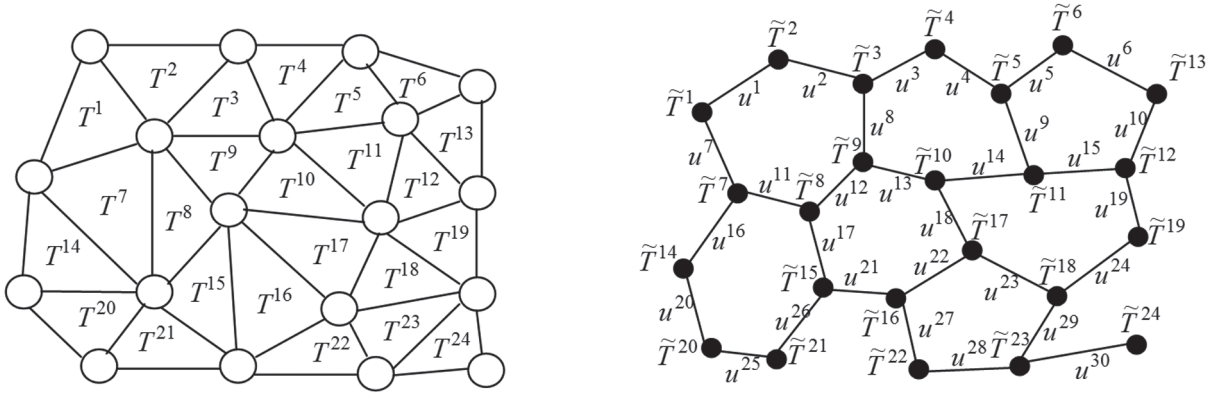
5) Если в графе G не осталось ни одной вершины, то остовые леса O построены.

6) Если в подмножестве вершин графа G, из которых исходят удаленные ребра на шаге 4в, образовались изолированные вершины (вершины, из которых не исходит ни одно ребро), то остовой лес при данном выборе вершин в шагах 3 или 7 не существует, и данную ветвь в графе D далее не следует рассматривать. Если эта ветвь была единственная, то остовой лес O_l для данного графа G не существует.

7) Среди вершин графа G, из которых исходят удаленные ребра на шаге 4в, найдем такие h вершин $\tilde{T}^h\in G$, из которых исходит одно и только одно ребро $u^h\in U$. Эти ребра принадлежат остовному лесу ($u^h\in O_l$). Если таких вершин в графе G нет, то в подмножестве вершин графа G, из которых исходят удаленные ребра на шаге 4в, выбираем любую вершину $\tilde{T}^i\in G$, из которой исходят два ребра $u^i,u^j\in U$. Рассмотрим два случая принадлежности ребер $u^i\in O_l$ и $u^j\in O_l$ остовному лесу. Достроим граф D аналогично шагам 2 и 3.

8) Переход к шагу 4.

Граф D состоит из нескольких ветвей, каждая из которых представляет собой остовной лес O графа G. Каждый остовной лес получается путем объеди-



а *б*

Рис. 2. *а* – триангуляция Делоне T, *б* – граф G

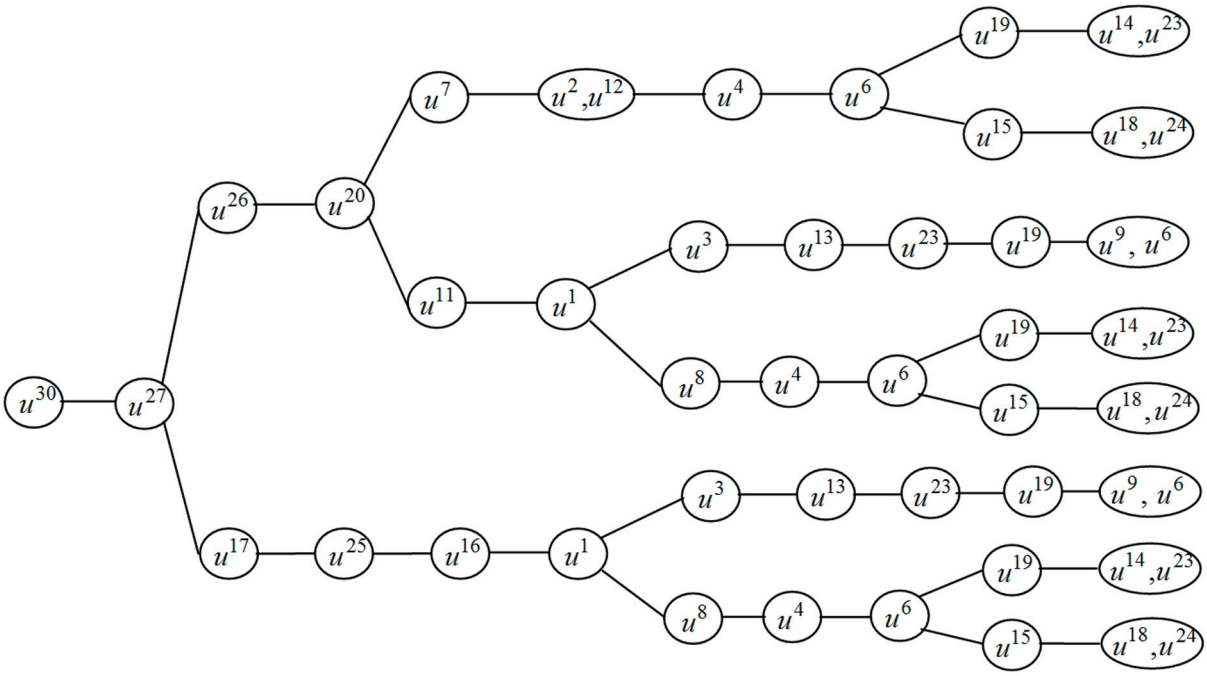


Рис. 3. Построенный граф D для графа G

Таблица 1

Остовые леса графа G

№	Остовые леса О графа G	№	Остовые леса О графа G
O_1	$\{u^{30},u^{27},u^{26},u^{20},u^{7},u^{2},u^{12},u^{4},u^{6},u^{19},u^{14},u^{23}\}$	O_5	$\{u^{30},u^{27},u^{26},u^{20},u^{11},u^{1},u^{8},u^{4},u^{6},u^{15},u^{18},u^{24}\}$
O_2	$\{u^{30},u^{27},u^{26},u^{20},u^{7},u^{2},u^{12},u^{4},u^{6},u^{15},u^{18},u^{24}\}$	O_6	$\{u^{30},u^{27},u^{17},u^{25},u^{16},u^{1},u^{3},u^{13},u^{23},u^{19},u^{9},u^{6}\}$
O_3	$\{u^{30},u^{27},u^{26},u^{20},u^{11},u^{1},u^{3},u^{13},u^{23},u^{19},u^{9},u^{6}\}$	O_7	$\{u^{30},u^{27},u^{17},u^{25},u^{16},u^{1},u^{8},u^{4},u^{6},u^{19},u^{14},u^{23}\}$
O_4	$\{u^{30},u^{27},u^{26},u^{20},u^{11},u^{1},u^{8},u^{4},u^{6},u^{19},u^{14},u^{23}\}$	O_8	$\{u^{30},u^{27},u^{17},u^{25},u^{16},u^{1},u^{8},u^{4},u^{6},u^{15},u^{18},u^{25}\}$

нения всех ребер одной ветви, записанных в вершинах графа D. Согласно рис. 5 для графа G, приведенного на рис. 4а, существует 8 остовных лесов.

На рис. 4 приведена последовательность итераций алгоритма построения остового леса O_1 графа G. На рисунке приняты следующие обозначения: жирной линией (——) показаны ребра $u^i \in U$, принадлежащие остовному лесу $u^i \in O_1$, выбранные на шагах 2, 3 и 7 и удаленные из графа G на шаге 4а; штрихованной линией (-----) показаны ребра, удаленные из графа G на шаге 4в; большой заштрихованной окружностью (⊘) показаны вершины, которые рассматриваются на шагах 6 и 7.

На основе полученного остового леса O_1 можно построить четырехугольную карту Кохонена $Q=\left(Q^1,...,Q^{\frac{M}{2}}\right)$ на основе триангуляции T. Это достигается путем удаления общих сторон треугольников из триангуляции T, которым соответствуют ребра графа G, образующие остовой лес O_1. Четырёхугольник Q^i, $i=\overline{1,\frac{M}{2}}$ определяется четырьмя точками с номерами $Q^i=\{Q_1^i,Q_2^i,Q_3^i,Q_4^i\}$, причем номера обозначены так, что попарно соединены точки $\{Q_1^i,Q_2^i\}$, $\{Q_1^i,Q_3^i\}$, $\{Q_2^i,Q_4^i\}$ и $\{Q_3^i,Q_4^i\}$.

На рис. 5 показан пример построения четырехугольной карты для триангуляции T, приведенной на рис. 2а.

Замечания:

1) Если найдена ветвь графа D, которой соответствует остовой лес O_1 графа G, то другие ветви можно не рассматривать. Вычислительная сложность предложенного метода линейно зависит от количества треугольников триангуляции T.

2) Если во всех ветвях графа D возникают изолированные вершины графа G, то алгоритм можно продолжить, и построить остовое дерево с изолированными вершинами. В результате, после построения четырехугольной карты в ней останутся треугольники, соответствующие изолированным вершинам графа G.

Выводы

В работе предложена процедура построения четырёхугольных карт Кохонена на основе их треугольного аналога, которая позволяет уменьшить вычислительную сложность аппроксима-

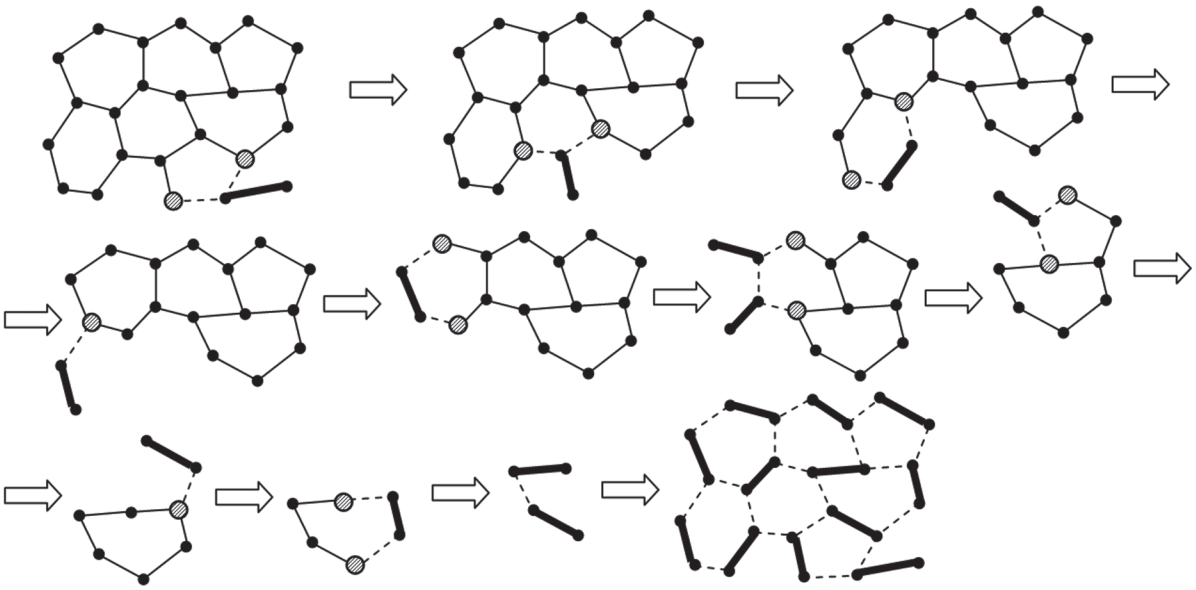


Рис. 4. Построение остового леса O_1 графа G, приведенного на рис. 2*б*

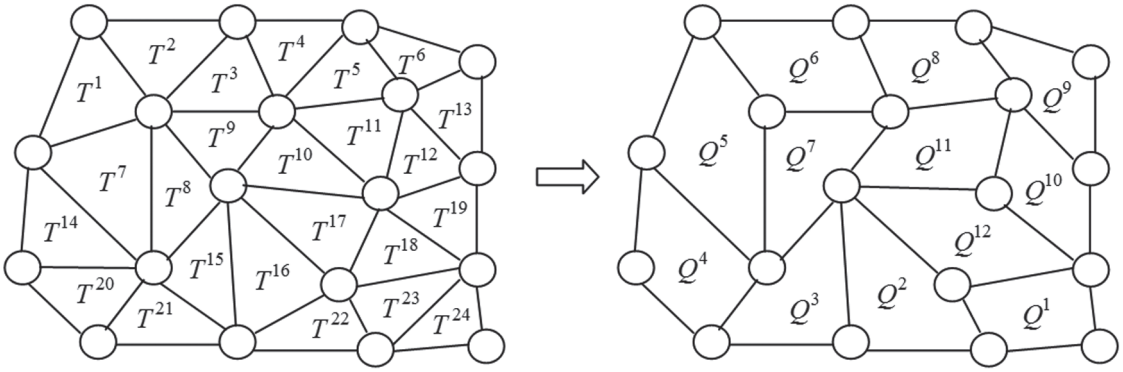


Рис. 5. Построение четырехугольной карты Кохонена по триангуляции

ции кусочно-плоской карты Кохонена кубической параметрической сплайн-поверхностью. Приведены необходимые условия существования четырёхугольных карт Кохонена. Построение таких карт ведет к существенному упрощению алгоритма аппроксимации параметрической сплайн-поверхностью и уменьшает количество скалярных операций. Показано, что предложенный метод имеет линейную вычислительную сложность. Планируется разработать метод аппроксимации сплайн поверхностями четырёхугольных карт Кохонена.

Список литературы: 1. *Зиновьев, А.Ю.* Визуализация многомерных данных [Текст] / А. Ю. Зиновьев. – Красноярск: Изд-во КГТУ, 2000. – 168 с. **2.** *Аксак, Н.Г.* Метод аппроксимации карт Кохонена кубическим параметрическим сплайном [Текст] / Н. Г. Аксак, А. В. Шкловец // Системи управління навігації та зв'язку. – 2010. – 2(14). – С.70-74. **3.** Метод аппроксимации сплайнами минимальной длины линейных карт Кохонена для визуализации многомерных данных [Текст] : сб. науч. тр. / НИЯУ МИФИ. М.: НИЯУ МИФИ, 2010. - 280 с. **4.** Аппроксимация двумерных карт Кохонена кубическими сплайн поверхностями [Текст] // сб. науч. тр. – Евпатория: МОІНУ, 2010. – 225 с. **5.** *Скворцов, А. В.* Триангуляция Делоне и её применение [Текст] / А. В. Скворцов. – Томск: изд-во Том. Ун-та, 2002. – 128 с. **6.** *Кристофидес, Н.* Теория графов. Алгоритмический подход [Текст] / Н. Кристофидис. – М: Мир, 1978. – 432 с.

Поступила в редколлегию 30.06.2011

УДК 004.032.26:519.174.2

Побудова чотирикутних карт Кохонена на основі тріангуляції Делоне для візуалізації багатовимірних даних / А. В. Шкловець, Н. Г. Аксак // Біоніка інтелекту: наук.-техн. журнал. – 2011. – № 3 (77). – С. 94-97.

Запропоновано метод побудови чотирикутних карт Кохонена для зменшення кількості скалярних операцій апроксимації кусочно-плоских карт Кохонена кубічними параметричними сплайн-поверхнями для збільшення точності візуалізації багатовимірних даних.

Табл 1. Іл. 5. Бібліогр.: 6. найм.

UDK 004.032.26:519.174.2

Building a rectangular Kohonen maps based on Delone triangulation to visualize multidimensional data / A.V. Shklovets, N. G. Axak // Bionics of Intelligense: Sci. Mag. – 2011. – № 3 (77). – P. 94-97.

A method for construction rectangular Kohonen maps to reduce the number of scalar operations of approximation planar Kohonen maps by cubic parametric spline surfaces to increase the accuracy of multidimensional data visualization was suggested.

Tab.: 1. Fig.: 5. Ref.: 6 items.

УДК 616.07:004.032.26

С.С. Мартиненко[1], Саад Джулгам[2]
[1]СумДУ, м. Суми, Україна, smart@unesco.sumdu.edu.ua
[2]СумДУ, м. Суми, Україна, saad710@mail.ru

СТИСНЕННЯ ТА РОЗПІЗНАВАННЯ МЕДИЧНОЇ ВІДЕОІНФОРМАЦІЇ

Розглядається ієрархічний алгоритм оптимізації параметрів навчання системи розпізнавання зображень медичних і біологічних об'єктів у рамках інформаційно-екстремальної інтелектуальної технології, що ґрунтується на максимізації кількості інформації, одержаної в процесі навчання системи.

МАГНІТОКАРДІОГРАМА, РОЗПІЗНАВАННЯ ЗОБРАЖЕНЬ, ІЄРАРХІЧНА СТРУКТУРА, ПОЛЯРНА СИСТЕМА КООРДИНАТ, ІНФОРМАЦІЙНИЙ КРИТЕРІЙ

Вступ

Задача розпізнавання медичної відеоінформації суттєво ускладнюється із збільшенням потужності класів розпізнавання. Одним із перспективних шляхів вирішення цієї проблеми є побудова ієрархічної структури алгоритмів навчання та екзамену системи розпізнавання [1,2]. Більшість відомих методів розпізнавання зображень носять в основному модельний характер, оскільки вони не орієнтовані на апріорно нечітке розбиття простору ознак на класи , яке має місце на практиці. Одним із перспективних підходів до підвищення функціональної ефективності систем розпізнавання зображень є інформаційно-екстремальна інтелектуальна технологія (IEI-технологія), що ґрунтується на максимізації інформаційної спроможності системи, що навчається [3, 4]. У працях [5, 6] розглядалося питання стиснення відеоінформації у рамках IEI-технології для однорівневої структури алгоритму навчання, але побудовані вирішальні правила не були безпомилковими за навчальною матрицею.

У статті розглядається задача підвищення функціональної ефективності навчання системи розпізнавання магнітокардіограм за ієрархічним алгоритмом шляхом оптимизации кроку зміни кута зчитування яскравості при обробленні зображень в полярній системі координат.

1. Постановка задачі

Нехай зображення магнітокардіограм різних станів серцево-судинної системи утворють ієрархічну структуру алфавіту класів розпізнавання $\{X_{r,s,m} \mid r=\overline{1,R};\ s=\overline{1,S_r};\ m=\overline{1,M_{r,s}}\}$. Відомий структурований вектор параметрів функціонування системи розпізнавання $g=< x_{r,s,m}, d_{r,s,m}, \delta_{r,s}, \phi_{r,s} >$, де $x_{r,s,m}$ – еталонний вектор-реалізація класу $X^o_{r,s,m}$; d_m – радіус контейнера класу $X^o_{r,s,m}$, що відновлюється в радіальному базисі простору ознак розпізнавання; $\delta_{r,s}$ – параметр поля контрольних допусків на ознаки розпізнавання для алфавіту класів s_r -ї страти, $\phi_{r,s}$ – крок зміни кута зчитування яскравості зображення при його обробленні в полярних координатах для алфавіту класів s_r -ї страти. При цьому задано обмеження на відповідні параметри функціонування.

Необхідно на етапі навчання побудувати оптимальні вирішальні правила, які забезпечать максимум усередненого критерію функціональної ефективності (КФЕ) $\overline{E}_r$ навчання системи розпізнавати класи r -го ярусу ієрархічної структури

$$\overline{E}_r = \frac{1}{M_{r,s}S_r}\sum_{s=1}^{S_r}\sum_{m=1}^{M_{r,s}} E^*_{r,s,m}, \qquad (1)$$

де $S_r, M_{r,s}$ – кількість страт і кількість класів в s_r -й страті r -го ярусу відповідно; $E^*_{r,s,m}$ – глобальний максимум інформаційного КФЕ навчання системи розпізнавати класи s_r -ї страти.

На етапі екзамену необхідно віднести реалізацію, що розпізнається, до відповідного класу розпізнавання із заданого алфавіту.

2. Категорійна модель системи розпізнавання

Математичну модель процесу навчання у вигляді діаграми відображень множин показано на рис. 1, де наведено такі позначення: G – множина факторів, що впливають на систему; T – множина моментів часу зняття інформації; Ω – простір ознак розпізнавання; Z – простір можливих функціональних станів системи; Θ – вхідні реалізації образу; Y – вхідна вибіркова множина (вхідна навчальна матриця яскравості зображень); $\Phi_1 : G\times T\times\Omega\times Z\to\Theta$ – оператор формування вхідних реалізацій образів; $\Phi_2 : \Theta\to Y$ – оператор формування вибіркової множини Y. Оператор $\theta : Y\to\tilde{\Re}^{|M|}$ будує розбиття $\tilde{\Re}^{|M|}$ простору ознак на класи розпізнавання, яке у загальному випадку є нечітким, а оператор класифікації $\Psi : \tilde{\Re}^{|M|}\to I^{|l|}$ перевіряє основну статистичну гіпотезу про належність реалізацій $\{x^{(j)}_m \mid j=\overline{1,n}\}$ класу X^o_m і формує множину гіпотез $I^{|l|}$, де l – кількість статистичних гіпотез.

Оператор γ: I| 1 | $\to\Im$ | q | шляхом оцінки статистичних гіпотез формує множину точнісних характеристик $\Im^{|q|}$, де $q=l^2$ – кількість точнісних характеристик. Оператор $\phi : \Im^{|q|}\to$E обчислює множину значень інформаційного критерію E, який

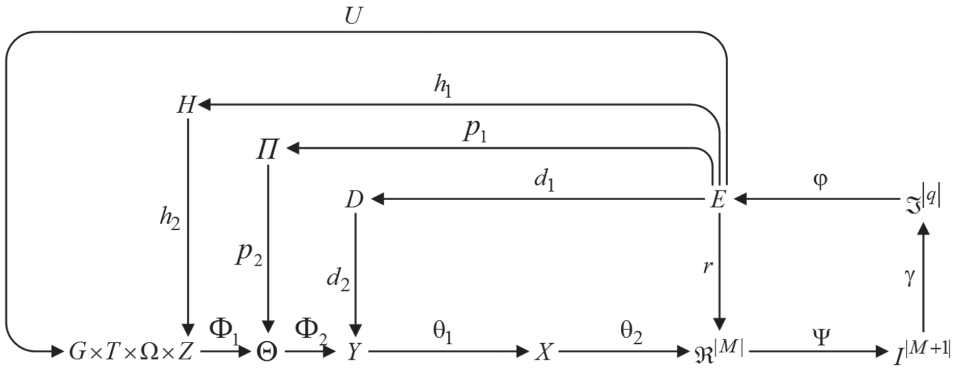


Рис. 1. Математична модель навчання системи розпізнавання

є функціоналом точнісних характеристик. Контур діаграми, який послідовно складається з операторів Ψ,γ,ϕ і r оптимізує геометричні параметри розбиття $\tilde{\Re}^{|M|}$ шляхом пошуку глобального максимуму критерію E в робочій (допустимій) області визначення його функції. Оптимізація системи контрольних допусків здійснюється за ітераційною процедурою, в якій послідовно задіяно оператори $\theta, \psi, \gamma, \phi, \delta_1$ і δ_2. Контур оптимізації кроку зміни кута зчитування яскравості зображень замикається операторами $p_1: \mathrm{E} \to \Pi$, де Π – терм-множина кроків зміни кута зчитування, і $p_2: \Pi \to \Theta$, який змінює вхідні реалізації образу. Оптимізація ієрархічної структури відбувається за допомогою операторів $h_1 : E \to H$, де H – множина, яка складається з можливих варіантів ієрархічних структур, та $h_2 : H \to Z$. Оператор U: $E \to G \times T \times \Omega \times Z$ регламентує процес навчання системи.

3. Алгоритм навчання системи розпізнавання

Як відомо, оброблення діагностичних медичних зображень в полярних координатах дозволяє забезпечити алгоритму навчання інваріантність до їх зсуву, повороту та зміні масштабу. Інформаційно-екстремальний ієрархічний алгоритм навчання системи розпізнавання з оптимізацією кроку зміни кута зчитування при обробленні зображень в полярних координатах подамо у вигляді багатоциклічної ітераційної процедури пошуку глобального максимуму інформаційного КФЕ (1), що обчислюється в робочій (допустимій) області визначення його функції

$$\Delta\phi^{*}_{r,s} = \arg < \max_{G_\phi}\{\max_{G_\delta}\{\max_{G_E} \overline{E}_r\}\} >, \qquad (2)$$

де G_ϕ, G_δ, G_E – допустимі області значень кроку зміни кута зчитування $\Delta\phi_{r,s}$, параметра поля контрольних допусків $\delta_{r,s}$ і КФЕ відповідно.

Для наочності обмежимося оптимізацією параметрів навчання системи розпізнавання для одного ярусу ієрархічної структури алфавіту класів. Вхідними даними є діагностичні медичні зображення, що характеризують класи r-го ярусу; система нормованих полів допусків $\{\delta_{H,i} \mid i = \overline{1,N}\}$ на ознаки розпізнавання, що визначає області значень відповідних контрольних допусків, та розподіл класів розпізнавання по стратам ярусу згідно із заданою ієрархічною структурою алфавіту класів розпізнавання. При цьому в кожній страті визначається базовий клас, відносно якого формується система контрольних допусків на ознаки розпізнавання.

Оброблення зображень в полярній системі координат та формування навчальної матриці $\| y^{(j)}_{r,s,m,i} \mid m = \overline{1, M_{r,s}}; i = \overline{1,N}; j = \overline{1,n} \|$, в якій i-а ознака в j-му векторі-реалізації образу обчислюється шляхом усереднення значень яскравості відповідних RGB-складових зображення за формулою [5]

$$\Theta^{(j)}_i = \frac{1}{2\pi R_i} \sum_{l=1}^{L} \theta^{(j)}_{l,i}, \; i = \overline{1,N}, \qquad (3)$$

де $\Theta^{(j)}_i$ –значення ознаки розпізнавання в j-й реалізації образу; $\theta^{(l)}_{l,i}$ – значення RGB-складової в l-му пікселі i-го кола зчитування яскравості зображення; L – кількість пікселів в i-му колізчитування; R_i – радіус i-го кола зчитування.

Розглянемо основні етапи реалізації інформаційно-екстремального ієрархічного алгоритму навчання системи розпізнавання з оптимізацією кроку зміни кута зчитування при обробленні в полярних координатах діагностичних медичних зображень.

1. Обнуління лічильника кількості страт в r-му ярусі: $s := 0$;

2. $s := s + 1$;

3. Обнуління лічильника зміни кроку зчитування: $\Delta_{r,s} := 0$;

4. $\Delta_{r,s} := \Delta_{r,s} + 1$;

5. Обнуління лічильника кроків зміни параметра поля контрольних допусків $\delta_{r,s}$: $\delta := 0$;

6. $\delta := \delta + 1$;

7. На кожному кроці зміни значення параметра поля допусків обчислюються нижній $A_{KH\,r,s,,i}[\delta]$ і верхній $A_{KBr,s,i}[\delta]$ контрольні допуски для всіх ознак розпізнавання за формулами

$$A_{KH\,r,s,i}[\delta] = y_{r,s,1,i} - \delta_{r,s}[k];$$
$$A_{KBr,s,i}[\delta] = y_{r,s,1,i} + \delta_{r,s}[l], \qquad (4)$$

де $y_{r,s,1,i}$ – вибіркове середнє значення i-ї ознаки в навчальній матриці базового (першого) класу $X^{o}_{r,s,1}$.

8. Формується бінарна навчальна матриця $\| x^{(j)}_{r,s,m,i} \|$ за правилом

$$x^{(j)}_{r,s,m,i} = \begin{cases} 1, & if \quad A_{KH\,r,s,\,i} \le y^{(j)}_{r,s,m,i} \le A_{KB\,r,s,\,i}; \\ 0, & if \quad else. \end{cases}$$

9. Для класу $X^o_{r,s,m}$ формується двійковий еталонний вектор-реалізація, i-й елемент якого визначається за правилом

$$x_{r,s,m,i} = \begin{cases} 1, & if \quad \frac{1}{n}\sum_{j=1}^{n} x^{(j)}_{r,s,m,i} > \rho_{r,s}; \\ 0, & if \quad else. \end{cases}$$

де $\rho_{r,s}$ – рівень селекції (квантування) координат еталонного вектора $x_{r,s,m} \in X^o_{r,s,m}$, який за замовчуванням дорівнює $\rho_{r,s} = 0{,}5$.

10. Формується за критерієм мінімальної кодової відстані структуроване попарне розбиття $\{\Re^{|2|}_{r,s,m} =< x_{r,s,m}; x_{r,s,c} >\}$ множини $\{x_{r,s,m}\}$, яке задає план навчання. Тут $x_{r,s,c}$ – еталонний вектор-реалізація класу $X^o_{r,s,c}$, найближчого до класу $X^o_{r,s,m}$.

11. Обчислюється значення інформаційного КФЕ навчання системи розпізнавати реалізації класу $X^o_{r,s,m}$. Як КФЕ може розглядатися будь-яка статистична інформаційна міра. Наприклад, для двоальтернативних рішень та рівноймовірних гіпотез можна застосувати модифікацію інформаційної міри Кульбака [4]

$$E^{(k)}_{r,s,m} = 0{,}5\log_2\left(\frac{D^{(k)}_{1\,r,s,m} + D^{(k)}_{2\,r,s,m}}{\alpha^{(k)}_{r,s,m} + \beta^{(k)}_{r,s,m}}\right) *$$

$$*\{[D^{(k)}_1 + D^{(k)}_2] - [\alpha^{(k)} + \beta^{(k)}]\} = \left|\begin{matrix} \alpha^{(k)}_{r,s,m} = 1 - D^{(k)}_{1\,r,s,m}; \\ \beta^{(k)}_{r,s,m} = 1 - D^{(k)}_{2\,r,s,m}. \end{matrix}\right| = \quad (5)$$

$$= 0{,}5\log_2\left(\frac{2 - [\alpha^{(k)}_{r,s,m} + \beta^{(k)}_{r,s,m}]}{\alpha^{(k)}_{r,s,m} + \beta^{(k)}_{r,s,m}}\right) * \{1 - [\alpha^{(k)}_{r,s,m} + \beta^{(k)}_{r,s,m}]\},$$

де $D^{(k)}_{1\,r,s,m}$ – перша достовірність прийняття рішень на k-му кроці навчання; $D^{(k)}_{2\,r,s,m}$ – друга достовірність; $\alpha^{(k)}_{r,s,m}$ – помилка першого роду; $\beta^{(k)}_{r,s,m}$ – помилка другого роду.

12. Якщо $\delta^{(k)}_{r,s} \le \delta_H / 2$, то виконується пункт 6, інакше – пункт 13.

13. У робочих областях визначення функції (5) здійснюється пошук її глобального максимального значення $E^*_{r,s,1}$.

14. Здійснюється за планом навчання оптимізація контейнерів інших класів, що відновлюються в радіальному просторі ознак розпізнавання.

15. Обчислюється усереднене максимальне значення $\overline{E}^{(k)}_{\max r,s}$ КФЕ навчання системи розпізнавати реалізації всіх класів при поточному значенні параметра навчання $\Delta^{(k)}_{\phi r,s}$.

16. Якщо $\Delta^{(k)}_{\phi r,s} \le \Delta_{\phi\,\text{гран}}$, то виконується пункт 4, інакше – пункт 17.

17. Визначаються усереднене максимальне значення $\overline{E}^*_{r,s}$ КФЕ навчання системи розпізнавати реалізації всіх класів поточної страти і оптимальні параметри навчання: еталонні вектори реалізації $\{x^*_{r,s,m}\}$, вершини яких визначають геометричні центри контейнерів відповідних класів; радіуси контейнерів класів розпізнавання $\{d^*_{r,s,m}\}$; параметр поля контрольних допусків $\delta^*_{r,s}$; система контрольних допусків на ознаки розпізнавання, які обчислюються за формулою (4) і параметр $\Delta^*_{r,s}$ – крок зміни кута зчитування яскравості зображення при його обробленні в полярних координатах для алфавіту класів поточної страти.

18. Якщо $s \le S_r$, то виконується пункт 2, інакше – ЗУПИН.

Таким чином, у рамках ІЕІ-технології процес оптимізації параметрів функціонування системи розпізнавання зображень за ієрархічним алгоритмом зводиться до ітераційного пошуку глобального максимуму інформаційного критерію (1) в робочій області визначення його функції.

4. Приклад реалізації алгоритму навчання

Реалізацію алгоритму навчання розглянемо на прикладі розпізнавання магнітокардіограм. Алфавіт класів розпізнавання складався з чотирьох класів, яка характеризували такі стани серецево-судинної системи людини: X^o_1 – нормальний стан, X^o_2 – ішемічна хвороба серця, X^o_3 – шуми в серці і X^o_4 – некоронарогенна хвороба серця. Оброблення зображень цих класів відбувалось у полярній системі координат за формулою (3)

Навчання системи розпізнавання магнітокардіограм відбувалося за ієрархічним алгоритмом, структуру якого показано на рис. 2.

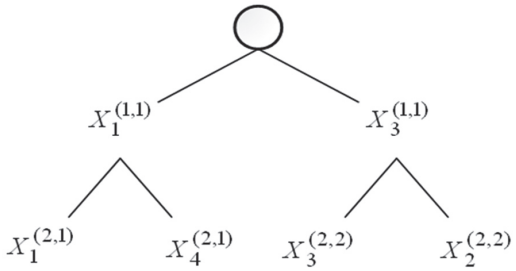
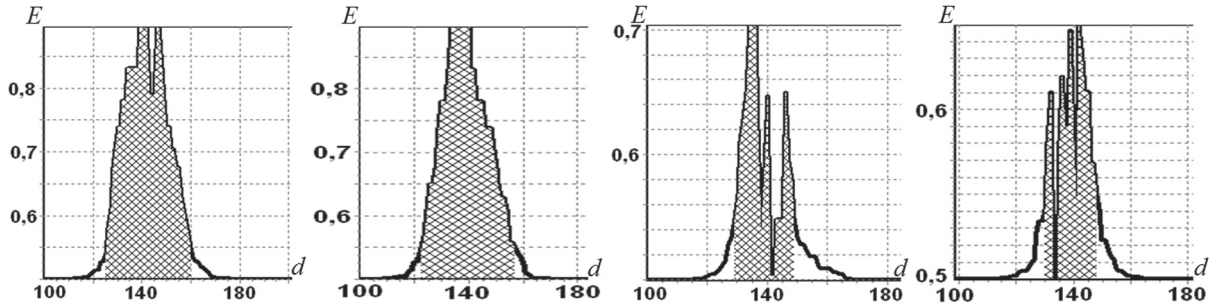


Рис. 2. Ієрархічна структура алфавіту класів розпізнавання

На рис. 2 верхні індекси в позначеннях класів визначають номер ярусу r та номер страти s відповідно.

У роботі [6] для верхнього рівня ієрархічної структури (рис. 2) для аналогічного алфавіту класів розпізнавання вдалося побудувати безпомилкові за навчальною матрицею вирішальні правила, але для другого рівня ієрархічної структури усереднене значення КФЕ дорівнювало $\overline{E}^*_2 = 0{,}72$. Тому для кожної страти цього рівня було реалізовано алгоритм навчання з оптимізацією кроку зміни кута зчитування яскравості зображень магнітокардіограм при їх обробленні в полярних координатах. Оптимізація кроку зміни кута зчитування яскравості при обробленні зображень магнітокардіо-



а *б* *в* *г*

Рис. 3. Графіки залежності КФЕ від радіусів контейнерів: *а* – клас $X^o_{2,1,1}$; *б* – клас $X^o_{2,1,2}$; *в* – клас $X^o_{2,2,1}$; *г* – клас $X^o_{2,2,2}$

грам здійснювалася за трициклічною ітераційною процедурою пошуку глобального максимуму КФЕ навчання системи (2) для кожної страти нижнього ярусу структури. Реалізація вищенаведеного алгоритму дозволила підвищити функціональну ефективність навчання, а усереднене значення КФЕ (1) збільшилося до величини $\overline{E}^*_2 = 0{,}78$. При цьому оптимальне значення кроку зміни кута зчитування дорівнювало для кожної із страт значенням $\Delta^*_{\phi 2,1} = \Delta^*_{\phi 2,2} = 5^\circ$. Таким чином, встановлено, що величина кроку зміни кута зчитування при формуванні навчальної матриці яскравості зображень впливає на функціональну ефективність навчання системи розпізнавання.

На рис. 3 показано результати оптимізації радіусів контейнерів класів розпізнавання із заданого алфавіту для страти $s_{2,1}$ (рис. 3*а* і рис. 3*б*) і для страти $s_{2,2}$ (рис. 3*в* і рис. 3*г*). На рис. 3 подвійною штриховкою позначено робочі (допустимі) області визначення функції інформаційного критерію (4), в яких здійснювався пошук глобальних максимумів КФЕ навчання системи розпізнавання.

Аналіз рис. 3 показує, що оптимальні радіуси контейнерів відповідних класів дорівнюють $d^*_{2,1,1} = 140$ (тут і далі в кодових одиницях), $d^*_{2,1,2} = 142$, $d^*_{2,2,1} = 134$ і $d^*_{2,2,2} = 145$.

За результатами алгоритму екзамену, при якому послідовно розпізнавалися 100 реалізацій класів із заданого алфавіту, усереднене значення повної ймовірності правильного розпізнавання дорівнювало $P_t = 0{,}87$, що перевищує відомі результати [7].

Висновки

1. Використання ієрархічної структури навчання системи розпізнавання із оптимізацією параметра зміни кроку кута зчитування яскравості при обробленні зображень магнітокардіограм в полярній систем і координат дозволило підвищити достовірність розпізнавання.

2. Для побудови безпомилкових вирішальних правил необхідно провести оптимізацію інших параметрів навчання.

Список літератури: 1. *Саати, Т.* Принятие решений: Метод анализа иерархий [Текст] / Т. Саати – М.: Радио и связь, 1993. – 273 с. **2.** Прикладная статистика: Классификация и снижение размерности: Справ. изд./ под ред. С.А. Айвазян; [составители: С.А. Айвазян, В.М. Бухштабер, И.С. Енюков, Л.Д. Мешалкин]. – М.: Финансы и статистика, 1989. – 607 с. **3.** *Краснопоясовський, А.С.* Інформаційний синтез інтелектуальних систем керування: Підхід, що ґрунтується на методі функціонально-статистичних випробувань [Текст] / Краснопоясовський А.С. – Суми: Видавництво СумДУ, 2004. – 261 с. **4.** *Довбиш, А.С.* Основи проектування інтелектуальних систем: Навч. посібник [Текст] / А.С. Довбиш– Суми: Видавництво Сум ДУ, 2009.– 171 с. **5.** *Мартиненко, С.С.* Оброблення та розпізнавання магнітокардіограм [Текст] / С.С. Мартиненко // Вісник Сумського державного університету. Серія Технічні науки.– Суми:Видавництво СумДУ, 2010. – № 1. – С.16-23. **6.** Информационно-экстремальный алгоритм распознавания магнитокардиограмм [Текст] / А.С. Довбиш, С.С. Мартиненко, А.С. Коваленко, Н.Н. Будник // Международный научно-технический журнал «Проблемы управления и информатики».– 2011.– №1.– С. 140-146. **7.** Use of machine learning for classification of magnetocardiograms / Embrechts M, Szymanski B, Sternickel K. [and others] //. Proc. IEEE Conference on System, Man and Cybernetics. – Washington DC, October 2003. – p. 1400-5.

Надійшла до редколегії 05.07.2011

УДК 616.07:004.032.26

Сжатие и распознавание медицинской видеоинформации / С.С. Мартыненко, Саад Джулгам // Бионика интеллекта: научн.-техн. журнал. – 2011. – № 3 (77). – С. 98-101.

Рассматривается иерархический алгоритм оптимизации параметров обучения системы распознавания изображений медицинских и биологических объектов в рамках информационно-экстремальной интеллектуальной технологии, которая базируется на максимизации количества информации, полученной в процессе обучения системы.

Ил.: 3. Библиогр.: 7 назв.

UDC 616.07:004.032.26

Medical video information compressing and recognition / S.S. Martynenko, Julgam Saad // Bionics of Intelligense: Sci. Mag. – 2011. – № 3 (77). –P. 98-101.

Article presents hierarchical algorithm of learning parameters optimization of recognition system of images of medical and biological objects. Algorithm is presented in bounds of information-extreme intelligence technology, based on maximization of amount of information, gained in system learning process.

Fig.: 3. Ref.: 7 items.

УДК 004.89

Н.М. Кораблев[1], А.А. Фомичев[2]
[1]ХНУРЭ, г.Харьков, Украина, korablev@kture.kharkov.ua
[2]ХНУРЭ, г.Харьков, Украина, alexandros_1985@mail.ru

КЛАСТЕРИЗАЦИЯ ДАННЫХ МЕТОДОМ K-MEANS С ИСПОЛЬЗОВАНИЕМ ИСКУССТВЕННЫХ ИММУННЫХ СИСТЕМ

В данной работе рассматривается алгоритм кластеризации данных методом k-means, функционирующий на основе использования искусственных иммунных систем. При определении центров k кластеров используется значение средней пороговой аффинности объектов. Для повышения скорости работы алгоритма кластеризации используются принципы клонирования граничных объектов и клонирования центров (центроидов).

ИСКУССТВЕННЫЕ ИММУННЫЕ СИСТЕМЫ, ПОСЛЕДОВАТЕЛЬНОЕ КЛОНИРОВАНИЕ, ПОРОГОВАЯ АФФИННОСТЬ, УРОВЕНЬ СТИМУЛЯЦИИ

Введение

В настоящее время проблема кластеризации данных является одной из наиболее актуальных в области информационных технологий. Среди большого количества существующих методов кластеризации данных одним из часто используемых является метод k-means [1, 4]. Данный метод отличается тем, что количество кластеров изначально известно и определяется числом k. Характерными особенностями данного метода являются его простота реализации и модификации. Однако методу k-means присущи некоторые недостатки – проблема сходимости, и, как следствие, невозможность определения времени, необходимого на кластеризацию данных.

Для повышения скорости работы алгоритма без потери точности при классификации методом k-means предлагается использование биологических принципов организации вычислений, основанных на принципах работы искусственных иммунных систем (ИИС) [2, 3, 5]. При этом большое внимание уделяется определению исходных центров k кластеров, а также работе операторов клонирования и мутации клонов. Характерной особенностью иммунной модификации метода k-means является ограничение количества объектов, которые будут клонированы, на основе критерия пороговой аффинности, что значительно ускоряет работу алгоритма.

1. Постановка задачи

Дано множество объектов n, каждый из которых описывается набором признаков, и количество искомых кластеров k. Исходные объекты представляются популяцией антиген $AG(ag_1;...;ag_n)$. В качестве меры близости между объектами используется критерий аффинности Af_{ij} [2, 3, 5]:

$$Af_{ij} = (1 + d_{ij})^{-1}, \quad (1)$$

где Af_{ij} – аффинность между i и j объектами, а d_{ij} – евклидово расстояние между их признаками.

Необходимо разработать алгоритм кластеризации объектов, функционирующий на основе метода k-means и иммунных принципов группировки, использующий в качестве основной меры близости объектов критерий (1), позволяющий определять центры k кластеров и принадлежность к ним исходных объектов.

2. Алгоритм кластеризации

Работу алгоритма k-means можно условно разделить на четыре основных этапа [1, 4]:

1. Определение k центров кластеров.
2. Определение принадлежности объектов кластерам.
3. Определение центроидов k кластеров.
4. Сравнение центров и центроидов кластеров.

Определение центров кластеров может производиться как случайным образом, так и на основе некоторых критериев (например, по значению среднего евклидового расстояния с остальными объектами). Центроиды являются центрами кластеров, они могут быть кластеризуемыми объектами и определяются по средним значениям признаков объектов, входящих в данный кластер. На последнем этапе производится сравнение центров (центроидов), полученных на предыдущей итерации алгоритма, и центроидов, поученнях после переопределения принадлежности объектов кластерам. В случае их совпадения алгоритм кластеризации завершает работу, иначе происходит возврат ко второму этапу, где будет определяться принадлежность объектов к кластерам на основании выделенных ранее центроидов.

Использование ИИС–принципов в организации работы k-means приводит к некоторому изменению в организации вычислений. Иммунная система оперирует с двумя типами клеток (лимфоцитов): Т-клетки, распознающие антигены и стимулирующие иммунный ответ, и В-клетки, нейтрализующие антигены. Антигенами $AG(ag_1;...;ag_n)$ являются все исходные объекты, В-клетки [3]

формируют популяцию $Bc(bc_1;...;bc_k)$ и являются кластерами, которые переопределяются в ходе работы алгоритма. В работе данного метода на начальном этапе используется популяция Т-клеток. Их основная задача заключается в определении критерия пороговой аффинности Aff_{thr} и стимуляции В-клеток.

Основная идея модифицированного k-means алгоритма при использовании ИИС состоит в том, что предлагаемая модель иммунной сети предполагает, что процесс поиска антигенов уже произведен и Т-клетки, вступившие во взаимодействие с антигенами, определили их признаки (априорная завершенность процесса распознавания). В результате распознавания популяция Т-клеток содержит признаки всех исходных антигенов, после чего для каждой Т-клетки определяется уровень стимуляции. После отбора k Т-клеток с наибольшим уровнем стимуляции формируется популяция В-клеток путем клонирования отобранных объектов. После этого популяция В-клеток клонируется и мутирует до тех пор, пока не будут выделены все В-клетки, каждая из которых в ходе клонирования и мутации станет соответствующей некоторому количеству входных антигенов.

Формально предлагаемый иммунный алгоритм k-means может быть представлен в виде следующей последовательности операторов:

$$\begin{gathered} kmAIS(ag_1;...;ag_n) = stimTcell(t_1;...t_n), \\ selBmem(b_1;...;b_k), clonBcell(b_1';...;b_k'), \\ stimBcell(b_1';...;b_k'), \\ matchBcell((b_1;...;b_k),(b_1';...;b_k')), \end{gathered} \quad (2)$$

где $stimTcell(t_1;...t_n)$ – оператор определения уровня стимуляции Т-клеток; $selBmem(b_1;...;b_k)$ – оператор выделения центров В-клеток (исходных центров кластеров); $clonBcell(b_1';...;b_k')$ – операторы клонирования, мутации и отбора В-клеток (формирование центроидов); $stimBcell(b_1';...;b_k')$ – оператор определения уровня стимуляции для полученных центроидов (определение принадлежности исходных объектов выделенным кластерам); $match((b_1;...;b_k),(b_1';...;b_k'))$ – оператор сравнения клеток памяти $Bmem(b_1;...;b_k)$ и сформированных клеток $Bcell(b_1';...;b_k')$.

Следует отметить, что работа оператора $clonBcell(b_1';...;b_k')$ может происходить на основе следующих подходов:

1. Клонирование граничных антител в В-клетке.
2. Клонирование центра (ядра) В-клетки.

Граничные антитела В-клетки определяются с помощью критерия пороговой аффинности Aff_{thr}. Это происходит из предположения, что все антитела выделенной В-клетки соответствуют антигенам в ее области, т.е. её антитела по своим признакам идентичны вступившим с ней во взаимодействие антигенам. После этого граничные антитела подвергаются клонированию и мутации. Принадлежность объекта (антигена) кластеру (В-клетке) определяется путем выбора максимального веса клонов объектов (антител), соответствующего одной из В-клеток (кластеров).

При клонировании всей В-клетки происходит мутация ее центра, представленного антителом с максимальным уровнем стимуляции ко всем антигенам, взаимодействующим с В-клеткой, либо её ядром (центроидом). После клонирования и мутации центра новое ядро (центроид) отбирается среди полученных клонов на основании конкурентного отбора, при этом ядром (центроидом) становится клон с наибольшим уровнем стимуляции ко всем антигенам В-клетки.

Для определения уровня стимуляции Т-клеток $stimTcell(t_1;...t_n)$ производится вычисление их средних аффинностей aff_{iAG} со всеми антигенами популяции $AG(ag_1;...;ag_n)$:

$$aff_{iAG} = \frac{\sum_{j=1}^{n} aff_{ij}}{n}, \quad (3)$$

где aff_{ij} – аффинность Т-клетки t_i и антигена ag_j.

После этого на основании полученных уровней стимуляции производится определение значения пороговой аффинности Aff_{thr}:

$$Aff_{thr} = \frac{\sum_{i=1}^{n} aff_{iAG}}{n}. \quad (4)$$

Пороговая аффинность используется при определении центров В-клеток (центров кластеров) и определении граничных антител (граничных объектов кластера) при кластеризации.

Перед формированием популяции В-клеток по значениям aff_{iAG} определяются k наиболее стимулированных Т-клеток, аффинность между которыми не превышает значения Aff_{thr}. Эти клетки формируют популяцию В-клеток, каждая из которых изначально характеризуется только одним антителом (центром). Данное антитело становится клеткой памяти системы. После определения центра происходит рост В-клеток – антитела относятся к В-клеткам по максимальным значениям аффинности с её ядром (центральным объектом, центроидом).

Клонирование и мутация В-клеток $clonBcell(b_1';...;b_k')$ может происходить на основе указанных подходов. В первом случае (клонирование граничных антител) из антител, входящих в В-клетку для клонирования, выбираются антитела, аффинность которых к центру В-клетки удовлетворяет выражению:

$$aff_{ij} \le \gamma \cdot Aff_{thr}, \quad (5)$$

где aff_{ij} – аффинность i-го антитела с j-ым центром В-клетки (центром кластера), а γ – коэффициент деформации (сжатия) клетки, устанавливаемый в процентах. Для каждого клонируемого антитела количество возможных клонов определяется следующим образом:

$$cl_i = int \| (k \cdot aff_{ij})^2 \|, \quad (6)$$

где $int\| \ \|$ – округляет значение до ближайшего большего целого, k – количество кластеров, а aff_{ij} – аффинность i-го антитела с центром j-ой В-клетки.

При мутации клонированных антител используется оператор гипермутации, в результате чего коэффициент мутации σ_i определяется как:

$$\sigma_i = \gamma \cdot (1 - aff_{ij}), \quad (7)$$

где aff_{ij} – аффинность i-го антитела с j-ым центром В-клетки, а γ – коэффициент деформации (сжатия) клетки, устанавливаемый в процентах.

После клонирования и мутации клоны антитела определяют аффинности со всеми k центрами В-клеток (центрами кластеров). Принадлежность антитела кластеру определяется по значению общей максимальной аффинности его клонов ядру (центру, центроиду) В-клетки. После этого происходит переопределение ядер (центроидов) В-клеток по средним аффинностям входящих в него антител.

Во втором случае (клонирование центра В-клетки) клонируется только ядро (центральный антиген), при этом количество клонов определяется как количество антител, содержащихся в данной В-клетке, а коэффициент мутации определяется как:

$$\sigma_i = \gamma \cdot aff_{iAG}, \quad (8)$$

где aff_{iAG} – аффинность центра i-й В-клетки с антигенами множества $AG(ag_1;...;ag_n)$.

После клонирования для каждого из клонов определяется средняя аффинность aff_{iAG} с антигенами $AG(ag_1;...;ag_n)$. В результате конкурентного отбора [5] остается только клон с наибольшим значением аффинности, который формирует новое ядро (центроид) мутировавшей В-клетки.

При определении уровня стимуляции полученных ядер (центроидов) В-клеток $stimBcell(b'_1;...;b'_k)$ для каждого нового ядра вычисляется аффинность aff_{iAG}, которая определяет уровень стимуляции В-клетки.

В работе оператора сравнения $match((b_1;...;b_k),(b'_1;...;b'_k))$ происходит определение аффинности между соответствующими парами центров В-клеток. Клонирование и мутация В-клетки прекращается, если выполняются следующие условия:

$$aff_{ij} \geq \frac{\gamma \cdot Aff_{thr}}{2}; \quad \sigma_i - \sigma_j \leq \frac{\gamma \cdot Aff_{thr}}{2}, \quad (9)$$

где γ – коэффициент деформации клетки; Aff_{thr} – пороговая аффинность; aff_{ij} – аффинность между соответствующей парой центров; σ_i и σ_j – уровни стимуляции антител.

В случае, если данное условие не выполняется, в памяти происходит замещение предыдущего центра В-клетки новым ядром (центроидом) и операции клонирования, стимуляции и сравнения повторяются.

Процесс кластеризации с помощью иммунного k-means алгоритма можно представить как следующую последовательность шагов:

1. Определение уровней стимуляции aff_{iAG} (3) для популяции Т-клеток и значения пороговой аффинности Aff_{thr} (4) для формируемого популяции В-клеток.

2. Определение k наиболее стимулированных Т-клеток, аффинность между которыми не превышает значения пороговой аффинности Aff_{thr}. Формирование исходной популяции В-клеток по результатам отбора Т-клеток.

3. Определение принадлежности свободных антител В-клеткам по их аффинностям с центрами В-клеток.

4. Клонирование и мутация В-клеток по одному из предложенных вариантов.

5. Определение принадлежности антител к мутировавшим В-клеткам по аффинностям с их центрами.

6. Сравнение центров В-клеток, расположенных в памяти, с центрами, выделенными в результате клонирования и мутации (8). В случае невыполнения условий (9) замещение клеток памяти выделенными центрами и переход к шагу 4, иначе переход к шагу 7.

7. Конец.

В результате работы будет получено множество k В-клеток (кластеров), содержащих антитела. Для антигенов принадлежность к кластерам (В-клеткам) определяется по принадлежности антител, вступивших с ними во взаимодействие.

3. Результаты экспериментальных исследований

Тестирование алгоритмов кластеризации производилось на нескольких наборах данных (табл. 1).

Таблица 1

Характеристики наборов данных

Характеристики	Набор 1	Набор 2	Набор 3
Количество объектов	100	1000	10000
Количество групп признаков	2	5	10
Размерность групп признаков	3	6	10
Кол. кластеров k	3	8	14

На первом этапе работы алгоритмов для определения исходных центров кластеров по стан-

дартному методу k-means потребовалось меньше времени, чем его иммунным модификациям. Этап роста выделенных кластеров потребовал от всех типов алгоритмов приблизительно одинаковых затрат времени, но стандартный метод k-means снова показал лучший результат по сравнению с двумя его иммунными модификациями. Аналогичный результат наблюдался также и на этапе выделения новых центров (центроидов) k кластеров, где стандартному алгоритму k-means снова потребовалось меньше времени, в то время как для иммунных методов на данном этапе характерны наибольшие затраты времени, поскольку клонирование и мутация объектов сопряжены с выполнением большого количества вычислительных операций. При этом важной особенностью является наблюдаемое различие между временными затратами различных иммунных модификаций k-means: подход клонирования граничных антител при небольших наборах данных, характеризующихся малым количеством объектов, небольшим количеством групп признаков и их размерностью, требует больших временных затрат, чем подход клонирования центра кластера. Однако при повышении количества объектов в наборе данных, повышении количества групп признаков или повышении размерности при клонировании граничных объектов затраты времени меньше, чем при клонировании и мутации центра В-клеток. На сравнение исходных центров и выделенных центроидов алгоритмам потребовалось приблизительно одинаковое количество времени.

Несмотря на то, что на выполнение одного прохода цикла кластеризации стандартному алгоритму k-means потребовалось значительно меньше времени, чем иммунным аналогам, его общие затраты времени больше, т.к. для кластеризации ему потребовалось значительно большее количество проходов цикла, чем его иммунным модификациям (табл. 2).

Таблица 2

Результаты работы алгоритмов

Алгоритмы	Набор 1		Набор 2		Набор 3	
	C	T	C	T	C	T
Стандартный k-means	13	100	154	100	1816	100
Иммунный k-means 1-го типа	9	96	96	87	1032	50
Иммунный k-means 2-го типа	7	88	93	83	1029	57

В табл. 2 представлены результаты работы алгоритмов кластеризации на различных наборах данных, где C – количество проходов цикла кластеризации для выделенного набора данных, T – общее время, затраченное на кластеризацию выделенных наборов данных в процентах. Алгоритм, которому потребовалось максимальное количество времени T, имеет 100%, а значения затрат времени других алгоритмов определяются относительно выделенного максимума.

Большое значение в работе иммунных алгоритмов приобретает коэффициент деформации клетки γ, который используется при определении граничных объектов (антител) (5) и при определении уровня мутации объектов (7), (8). Данный коэффициент отображает деформацию клеток организма, которая наблюдается в результате воздействия на них вирусов и бактерий, попавших в данный организм. В предложенных иммунных алгоритмах коэффициент деформации равен 10% ($\gamma = 10\%$). При повышении коэффициента деформации В-клеток предложенные иммунные алгоритмы требуют большего количества проходов цикла кластеризации и уступают стандартному k-means в затратах времени на кластеризацию. Уменьшение коэффициента деформации уменьшает необходимое количество проходов цикла кластеризации, следствием чего является уменьшение затрат времени на кластеризацию, но понижает точность группировки объектов, поэтому его целесообразно устанавливать в диапазоне 5-15%.

Выводы

В работе предложены иммунные модификации алгоритма k-means, в которых используются новые подходы для решения основных задач кластеризации данных. Для сокращения количества вычислений, влияющих на затраты времени, необходимого на кластеризацию, клонированию и мутации подвергаются лишь некоторые объекты исходного множества (граничные антигены, центры В-клеток). Выделение клонируемых объектов производится при использовании значения пороговой аффинности и коэффициента деформации, что значительно уменьшает количество выделяемых объектов и уменьшает затраты времени при кластеризации.

Использование конкурентно-целевого отбора приводит к повышению эффективности отбора клонов при минимальных временных затратах, а определение принадлежности граничных объектов кластерам происходит на основании максимальной средней аффинности его клонов к центральному объекту кластера (центроиду), что значительно ускоряет процесс кластеризации и повышает её точность.

По результатам тестирования алгоритмов на различных наборах данных видно, что предложенные иммунные модификации k-means незначительно усложняют алгоритм кластеризации, однако увеличивают скорость его работы.

Список литературы: 1. *Mirkin, B.G.* Clustering for Data Mining. A Data recovery Approach / B.G. Mirkin. – Taylor & Francis Group, 2005. – 278 p. **2.** *Дасгупта, Д.* Искусственные иммунные системы и их применение [Текст]: Пер. с англ. – А.А. Романюха. М.: ФИЗМАТЛИТ, 2006. – 344 с. **3.** An Overview of Artificial Immune Systems / J. Timmis, T. Knight, L.N. de Castro, E. Hart. – Natural Computation, 2004. – 29 p. **4.** *Duda, R.O.* Pattern classification / R.O. Duda, P.R. Hart, D.G. Stork – Willey & Sons. – 2001. – 738 p. **5.** *Кораблёв, Н.М.* Классификация объектов на основе искусственных иммунных систем [Текст] / Н.М. Кораблёв, А.А. Фомичёв // Системи обробки інформації. – 2010 – Вип. 6 (87). – С. 13-17.

Поступила в редколлегию 12.07.2011

УДК 004.89

Кластеризація даних методом k-means за допомогою штучних імунних систем / М.М. Корабльов, О.О.Фомічов // Біоніка інтелекту: наук.-техн. журнал. – 2011. – № 3 (77). – С. 102-106.

У роботі запропоновані імунні підходи до вирішення задачі кластеризації даних методом k-means. Для підвищення швидкодії алгоритму кластеризації використовуються принципи клонування граничних об'єктів та клонування центрів. Проведені експериментальні дослідження запропонованих підходів вказали на високу швидкість кластеризації даних без втрат точності та хорошу масштабованість алгоритмів.

Табл. 2. Бібліогр.: 5 найм.

UDK 004.89

K-means data clustering using artificial immune systems / N.M. Korablev, A.A. Fomichev // Bionics of Intelligense: Sci. Mag. – 2011. – № 3 (77). –P. 102-106.

The paper presents the immune approaches to solving the problem of data clustering method k-means. To speed enhancing operation of clustering algorithm is used principles of boundary objects cloning and center objects cloning of clusters. Experimental studies suggested approaches indicated a high rate of clustering data without loss of accuracy and good scalability algorithms.

Tab. 2. Ref. 5 items.

УДК 004.93'1

К.В. Барило

Сумський державний університет, м. Суми, Україна, kate.barylo@gmail.com

ОПТИМІЗАЦІЯ РІВНЯ СЕЛЕКЦІЇ КООРДИНАТ ЕТАЛОННИХ ВЕКТОРІВ ПРИ РОЗПІЗНАВАННІ ЕЛЕКТРОНОГРАМ

Розглядається алгоритм оптимізації рівня селекції координат еталонних векторів-реалізацій класів розпізнавання для заданого алфавіту в рамках інформаційно-екстремальної інтелектуальної технології, що ґрунтується на максимізації інформаційної спроможності системи розпізнавання. При цьому досліджено вплив рівня селекції на функціональну ефективність системи розпізнавання зображень.

ЕЛЕКТРОНОГРАМА, РІВЕНЬ СЕЛЕКЦІЇ, РОЗПІЗНАВАННЯ, ІНФОРМАЦІЙНИЙ КРИТЕРІЙ, НАВЧАННЯ

Вступ

Складність машинного розпізнавання електронограм, одержаних в електронній мікроскопії в режимі малої дифракції, обумовлена як довільними початковими умовами формування їх зображень, так і впливом неконтрольованих випадкових факторів (нестабільність параметрів мікроскопа та електронного пучка) [1]. Одним із перспективних напрямів аналізу і синтезу систем розпізнавання електронограм є використання ідей і методів інформаційно-екстремальної інтелектуальної технології (ІЕІ-технологія), що ґрунтується на максимізації інформаційної спроможності в процесі навчання системи [2,3]. У рамках ІЕІ-технології важливу роль в процесі навчання системи розпізнавання відіграє рівень селекції координат двійкових еталонних векторів-реалізацій образу [4], оскільки вони визначають геометричні центри контейнерів класів розпізнавання при їх цілеспрямованому відновленні в радіальному базисі простору ознак розпізнавання. Оптимізація рівня селекції дозволяє підвищити середню міжкласову кодову відстань для заданого алфавіту класів розпізнавання у відповідності з максимально-дистанційним принципом теорії розпізнавання образів.

У статті розглянуто інформаційно-екстремальний алгоритм оптимізації рівня селекції координат еталонних векторів з метою підвищення функціональної ефективності навчання системи розпізнавання електронограм.

1. Постановка задачі дослідження

Нехай дано алфавіт класів розпізнавання $\{X_m^o \mid m=\overline{1,M}\}$, навчальна матриця типу "об'єкт-властивість" $\| y_{m,i}^{(j)} \|$, $i=\overline{1,N}$, $j=\overline{1,n}$, де $N,n-$ кількість ознак розпізнавання та реалізацій образу відповідно. При цьому задано структурований вектор параметрів функціонування системи, що навчається розпізнавати реалізації класу X_1^o: $g=<x_1,d_1,\rho>$, який складається відповідно з еталонної реалізації x_1 класу X_1^o, геометричного параметра d_1 – кодової відстані гіперповерхні контейнера K_1^o класу X_1^o від вершини еталонної реалізації $x_1 \in X_1^o$ і рівня селекції ρ координат еталонних векторів класів розпізнавання, який є рівнем квантування дискрет полігону емпіричних частот потрапляння значень ознак розпізнавання у свої поля контрольних допусків. При цьому полігон будується так: по осі абсцис відкладаються ранги ознак розпізнавання, які відповідають номерам ознак у векторі-кортежі $x_m^{(j)}$, а по осі ординат – відносні частоти $\omega_{m,i}=n_i/n$, де n_i – кількість випробувань, при яких значення i-ї ознаки знаходиться в своєму полі контрольних допусків.

Задано допустимі області значень відповідних параметрів: $x_1 \in \Omega_B^{|N|}$, де $\Omega_B^{|N|}$ – бінарний простір ознак потужності N; $d_1 \in [0; d(x_1 \oplus x_c)-1]$, де x_c – еталонна реалізація сусіднього (найближчого до X_1^o) класу X_c^o, і рівень селекції ρ координат еталонної реалізації x_1, $\rho \in [0;1]$.

Треба на етапі навчання за апріорно класифікованими реалізаціями нечітких образів побудувати оптимальне в інформаційному розумінні чітке розбиття $\Re^{|M|}$ дискретного простору ознак Ω_B на M класів розпізнавання шляхом ітераційної оптимізації координат вектора параметрів функціонування g_1 за умови, що значення усередненого за алфавітом $\{X_m^o\}$ інформаційного критерію функціональної ефективності (КФЕ) навчання системи розпізнавання набуває глобального максимуму в робочій (допустимій) області визначення його функції.

2. Алгоритм інформаційно-екстремальної оптимізації рівня селекції координат еталонних векторів класів розпізнавання

Оптимізацію рівня селекції координат еталонних векторів класів розпізнавання будемо здійснювати за паралельним алгоритмом, при якому рівень селекції змінюється одночасно для всіх ознак розпізнавання. При цьому за умови прийняття у загальному випадку гіпотези нечіткої компактності реалізацій образу оптимізація рівня селекції ρ буде здійснюватися у рамках інформаційно-екстремального алгоритму навчання для гіперсферичного класифікатора, в якому

контейнери класів розпізнавання відновлюються на кожному кроці навчання в радіальному базисі дискретного простору Хеммінга. Оптимальний рівень селекції ρ^* координат еталонної реалізації $x_m \in X_m^o$ визначається у результаті реалізації багатоцикличної ітераційної процедури:

$$\rho^* = \arg\max_{G_\rho}\{\max_{G_d}\overline{E}^*\}\}, \qquad (1)$$

де G_ρ, G_d – області допустимих значень параметрів ρ і d_1; $\overline{E}^*$ – максимальне усереднене значення КФЕ для алфавіту класів розпізнавання.

Таким чином, внутрішній цикл процедури (1) реалізує базовий алгоритм навчання і зовнішній цикл – пошук оптимального значення рівня селекції. Базовий алгоритм навчання виконується за попередньо знайденим значенням параметра поля контрольних допусків δ за паралельним алгоритмом оптимізації.

Як критерій оптимізації параметрів навчання у рамках ІЕІ-технології використовувалася модифікація інформаційної міри Кульбака [4], в якій розглядається відношення правдоподібності у вигляді логарифмічного відношення повної ймовірності правильного прийняття рішень P_t до повної ймовірності помилкового прийняття рішень P_f. Для рівноймовірних двохальтернативних гіпотез міра Кульбака має такий вигляд:

$$E_m = \log_2\frac{P_{t,m}}{P_{f,m}} * [P_{t,m} - P_{f,m}] = \left|\begin{matrix} P_{t,m} = 0{,}5D_{1,m} + 0{,}5D_{2,m}; \\ P_{f,m} = 0{,}5\alpha_{\boldsymbol{m}} + 0{,}5\beta_m. \end{matrix}\right| =$$

$$= \frac{1}{2}\log_2\left(\frac{D_1 + D_2}{\alpha+\beta}\right)\left[(D_1 + D_2) - (\alpha+\beta)\right] =$$

$$= \log_2\left(\frac{2-(\alpha+\beta)}{\alpha+\beta}\right)\left[2 - (\alpha+\beta)\right], \qquad (2)$$

де D_1, D_2, α, β – перша та друга достовірності, помилки першого та другого роду відповідно.

При цьому усереднене значення критерію (2) визначається як

$$\overline{E} = \frac{1}{M}\sum_{m=1}^{M} E_m^*,$$

де E_m^* – максимальне значення КФЕ навчання системи розпізнавати реалізації класу X_m^o, обчислене в робочій (допустимій) області визначення функції критерію.

Розглянемо основні етапи алгоритму паралельної оптимізації рівня селекції:

1) Встановлюємо значення для верхнього ρv та нижнього ρn значення рівня селекції відповідно.

2) Встановлюємо крок зміни рівня селекції ρc=0,1.

3) Встановлюємо значення поточного рівня селекції: $\rho z := \rho + \rho c$.

4) Реалізується базовий алгоритм навчання, який включає в себе такі етапи:

а) формування бінарної навчальної матриці $\| x_{m,i}^{(j)} \|$, елементи якої дорівнюють

$$x_{m,i}^{(j)} = \begin{cases} 1, if\ y_{m,i}^{(j)} \in \delta_{K,i}^*, \\ 0, if\ y_{m,i}^{(j)} \notin \delta_{K,i}^*, \end{cases}$$

де $\delta_{k,i}^*$ – попередньо знайдене за паралельним алгоритмом оптимальне значення параметра поля контрольних допусків;

б) формування масиву еталонних двійкових векторів $\{x_{m,i} \mid m = \overline{1,M}, i = \overline{1,N}\}$, елементи якого визначаються за правилом

$$x_{m,i} = \begin{cases} 1, if\ \frac{1}{n}\sum_{j=1}^{n} x_{m,i}^{(j)} > pz, \\ 0, if\ \ else, \end{cases}$$

де ρz – поточне значення рівня селекції координат вектора $x_m \in X_m^o$;

в) розбиття множини еталонних векторів на пари найближчих «сусідів»: $\Re_m^{|2|} = <xm, xl>$, де xl – еталонний вектор сусіднього класу X_l^o;

г) оптимізація кодової відстані d_m відбувається за рекурентною процедурою;

д) процедура закінчується при знаходженні максимуму КФЕ в робочій області його визначення: $E_m^* = \max_{\{d\}} E_m$, де $\{d\} = \{0,1,...,d < d(x_m \oplus x_l)\}$ – множина радіусів концентрованих гіперсфер, центр яких визначається вершиною $x_m \in X_m^o$.

5) Якщо $\rho z < \rho v$, то виконується крок 3, інакше – 6.

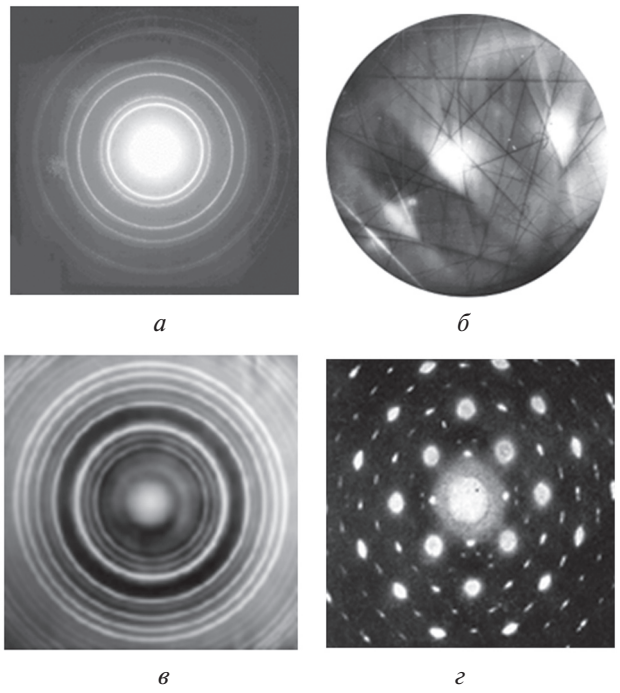
6) Визначається значення рівня селекції при якому усереднене значення КФЕ максимальне.

7) ЗУПИН.

Таким чином, алгоритм оптимізації рівня селекції координат двійкових еталонних векторів-реалізацій образу, як і інших параметрів навчання інтелектуальної СППР у рамках ІЕІ-технології полягає у наближенні глобального максимуму інформаційного критерію оптимізації до найбільшого його значення в області значень функції критерію.

3. Приклад реалізації алгоритму

Розглянемо алфавіт із чотирьох класів розпізнавання. Приклади електронограм-реалізацій кожного класу показано на рис. 1. При цьому клас X_1^o відповідає електронограмам алюмінію (рис. 1а), клас X_2^o – електронограми з Кікучі лініями (рис. 1б), X_3^o – матеріалу з монокристалічною структурою (рис. 1в), а X_4^o – матеріалу з полікристалічною структурою (рис. 1г). Розмірність кожного зображення електронограми становила 300×300 пікселів. Оскільки зображення електронограми є нестаціонарним за яскравістю, то за реалізацію приймається вся матриця яскравості. Матриця яскравості оброблялась у полярній системі координат з метою забезпечення інваріантності електронограм до операцій зсуву та повороту.



а *б* *в* *г*

Рис. 1. Електронограми: *а* – алюміній (клас X_1^o); *б* – з Кікучі лініями (клас X_2^o); *в* – монокристалічна структура (клас X_3^o); *г* – полікристалічна структура (клас X_4^o)

При обробленні зображень в полярних координатах вектор-реалізація образу формувався з ознак розпізнавання, які обчислювалися за формулою

$$\Theta_j = \frac{1}{N}\sum_{i=1}^{N}\theta_i ,$$

де Θ_j – числове значення усередненого спектру яскравості для кола зчитування в j-му радіусі, $j = \overline{1,R}$; θ_i – значення яскравості в і-му пікселі, $i = \overline{1,N}$; N – загальна кількість пікселів у колі зчитування.

Вхідна ціла навчальна матриця яскравості для кожного класу складалася із 30 реалізацій і мала такі параметри: $m = \overline{1.4}$, $i = \overline{1,150}$, $j = \overline{1,30}$.

Оптимізація параметра δ поля контрольних допусків на ознаки розпізнавання здійснювалася за паралельним алгоритмом при значенні рівня селекції $\rho = 0{,}5$. Графік залежності усередненого КФЕ (2) від параметра поля контрольних допусків δ показано на рис. 2.

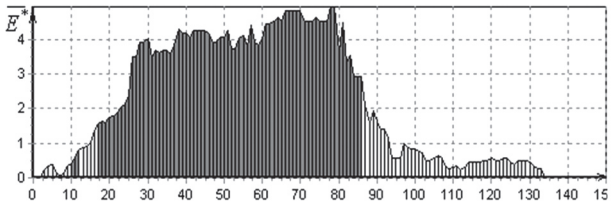
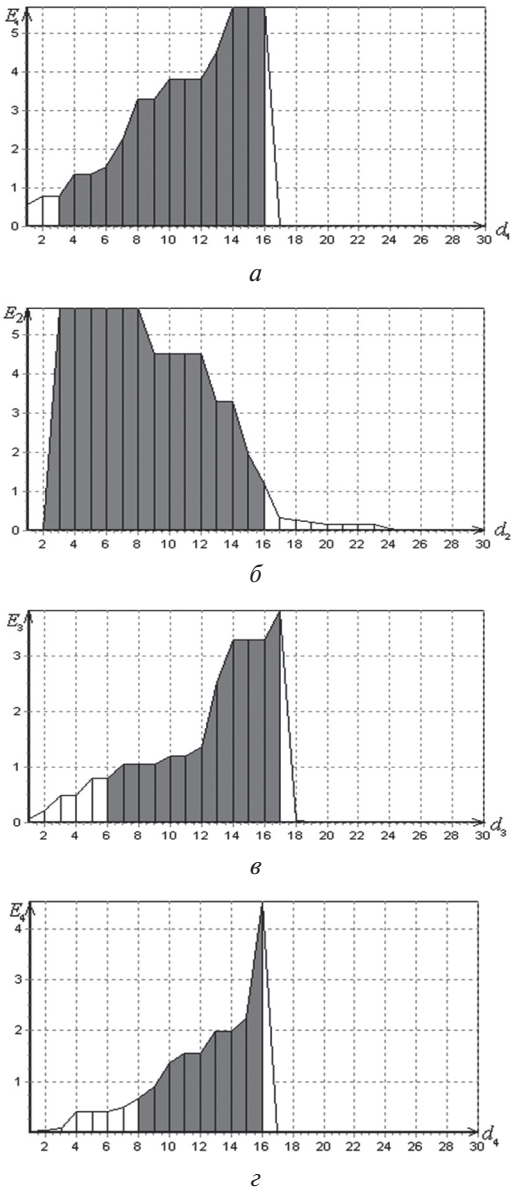


Рис. 2. Залежність усередненого КФЕ від параметра поля контрольних допусків δ при рівні селекції $\rho = 0{,}5$

Темними ділянками на графіку (рис. 2) позначено робочі області визначення КФЕ, в яких здійснюється пошук його глобального максимуму. Аналіз рис.2 показує, що оптимальне значення параметра δ, при якому значення усередненого КФЕ є максимальним, дорівнює $\delta^* = 79$. При цьому усереднене значення КФЕ становить 4,93.

На рис. 3 наведено залежність КФЕ від радіусів контейнерів класів розпізнавання після проведення оптимізації параметра поля контрольних допусків δ при значенні рівня селекції $\rho = 0{,}5$.



а *б* *в* *г*

Рис. 3. Графік залежності КФЕ від радіусів контейнерів: *а* – клас X_1^o; *б* – клас X_2^o; *в* – клас X_3^o; *г* – клас X_4^o

Аналіз рис. 3 показує, що відновлені в процесі навчання гіперсферичні контейнери класів розпізнавання мають такі оптимальні радіуси: $d_1^* = 14$ (тут і далі в кодових одиницях), $d_2^* = 5$, $d_3^* = 17$ і $d_4^* = 16$. При цьому оптимальні радіуси контейнерів класів X_1^o і X_2^o, для яких глобальні максимуми КФЕ на графіках мають ділянки типу "плато", визначалися за умови мінімального значення коефіцієнта нечіткої компактності [2]

$$l_m = \frac{d_m^*}{d(x_m \oplus x_c)} \to \min, \qquad (3)$$

де d_m^* – оптимальний радіус контейнера класу X_m^o;

$d(x_m \oplus x_c)$ міжцентрова відстань між еталонним вектором $x_m \in X_m^o$ і еталонним вектором x_c найближчого сусіднього класу X_c^o.

Після проведення навчання системи та знаходження оптимальних параметрів контейнерів класів розпізнавання міжцентрові кодові відстані для заданого алфавіту класів розпізнавання відповідно дорівнювали: $d(x_1 \oplus x_2)=17$, $d(x_1 \oplus x_3)=32$, $d(x_1 \oplus x4)=35$, $d(x_2 \oplus x_3)=17$, $d(x_2 \oplus x_4)=18$, $d(x_3 \oplus x_4)=23$. При цьому середнє значення міжцентрової кодової відстані дорівнює 24 кодових одиниць. А усереднене значення коефіцієнта нечіткої компактності (3) дорівнює

$$\bar{l} = \frac{1}{M}\sum_{m=1}^{M} l_m = 0{,}75.$$

Згідено з максимально-дистанційним принципом теорії розпізнавання образів одним із шляхів підвищення функціональної ефективності навчання системи є збільшення міжцентрових кодових відстаней. З цією метою в рамках IEI-технології було проведено на етапі навчання за процедурою (1) оптимізацію рівня селекції координат еталонних векторів-реалізацій класів розпізнавання. Графік залежності усередненого значення КФЕ для заданого алфавіту класів розпізнавання від значення рівня селекції ρ показано на рис. 4.

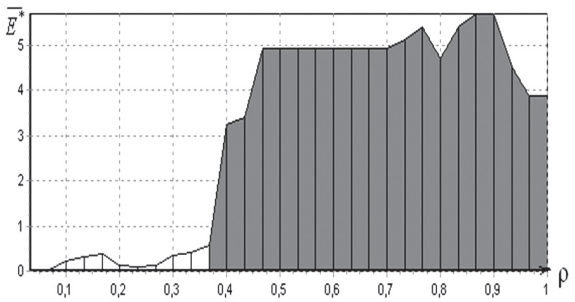


Рис. 4. Графік залежності усередненого КФЕ від значення рівня селекції

Аналіз рис.4 показує, що оптимальне значення рівня селекції визначається при глобальному максимумі КФЕ і дорівнює $\rho^* = 0{,}87$. Таким чином, порівняльний аналіз графіків, показаних на рис. 2 і рис. 4 дозволяє зробити висновок, що оптимізація рівня селекції дозволило збільшити значення усередненого КФЕ з 4,93 (при $\rho = 0{,}50$) до 5,68.

На рис. 5 показано графіки залежності КФЕ від радіусів контейнерів відповідних класів розпізнавання для оптимального рівня селекції координат еталонних векторів.

Аналіз рис. 5 показує, що відновлені в процесі навчання гіперсферичні контейнери класів розпізнавання мають такі оптимальні радіуси: $d_1^* = 22$, $d_2^* = 3$, $d_3^* = 25$ і $d_4^* = 22$. При цьому міжцентрові кодові відстані для заданого алфавіту класів розпізнавання відповідно дорівнюють: $d(x_1 \oplus x_2)=25$, $d(x_1 \oplus x_3)=48$, $d(x_1 \oplus x4)=44$, $d(x_2 \oplus x_3)=27$, $d(x_2 \oplus x_4)=28$, $d(x_3 \oplus x_4)=36$, а значення середньої міжцентрової кодової відстані дорівнює 35, що суттєво перевищує аналогічне значення, одержане при неоптимальному значенні рівня селекції ($\rho = 0{,}50$), яке застосовується за замовченням.

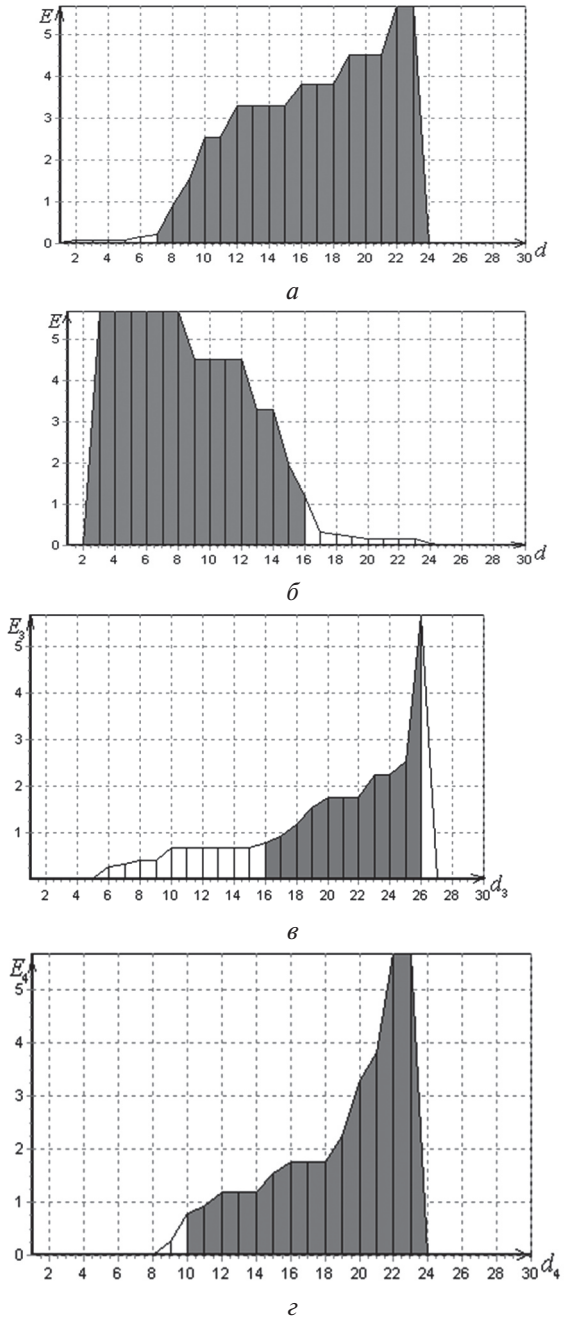


Рис. 5. Графік залежності КФЕ навчання від радіусів контейнерів класів розпізнавання: *а* – клас X_1^o; *б* – клас X_2^o; *в* – клас X_3^o; *г* – клас X_4^o

Таким чином, усереднений коефіцієнт нечіткої компактності дорівнює $\bar{l}=0{,}68$ (до оптимізації рівня селекції $\bar{l}=0{,}75$), що свідчить про зменшення при оптимальному рівні селекції ступеня перетину класів розпізнавання.

Висновки

1. Оптимізація рівня селекції координат еталонних векторів класів розпізнавання дозволяє підвищити функціональну ефективність навчання системи розпізнавання електронограм.

2. У процесі відновлення контейнерів класів розпізнавання при оптимальному рівні селекції середня міжцентрова відстань збільшилася, а середнє значення коефіцієнта нечіткої компактності зменшилося, що відповідає дистанційно-максимальному і дистанційно-мінімальному принципам теорії розпізнавання образів.

3. Оскільки побудовані вирішальні правила не є безпомилковими за навчальними матрицями, то для підвищення функціональної ефективності навчання доцільно застосовувати ієрархічні схеми алгоритмів.

Список літератури: **1.** *Томас, Г.* Просвечивающая электронная микроскопия материалов: [Текст] / Г. Томас, М.Дж. Горинджс: Под ред.. Б.К. Вайнштейна: Пер. с англ. – М.: Наука. Гл. ред. Физ.-мат. лит. – 1983. – 320 с. **2.** *Довбиш, А.С.* Основи проектування інтелектуальних систем: Навчальний посібник [Текст] / А.С. Довбиш. – Суми: Вид-во СумДУ, 2009. – 171 с. **3.** *Краснопоясовський, А.С.* Інформаційний синтез інтелектуальних систем керування, що навчаються: Підхід, що грунтується на методі функціонально-статистичних випробувань [Текст] / А.С. Краснопоясовський. – Суми: Видавництво СумДУ. – 2003. – 257 с. **4.** *Шелехов, І. В.*Вибір базового класу при розпізнаванні зображень [Текст] / І. В. Шелехов, К. В. Барило // Вісник Сумського державного університету. Серія Технічні науки. — 2010. — № 3, Т. 2. — С. 95-102. **5.** *Довбиш, А.С.* Інформаційно-екстремальний метод розпізнавання електронограм [Текст] / А.С. Довбиш, С.С. Мартиненко // Вісник Сумського державного університету. Серія: Технічні науки.– 2009. – № 2. – С. 85-91.

Надійшла до редколегії 26.08.2011

УДК 004.93’1

Оптимизация уровней селекции координат эталонных векторов при распознавании электронограмм / Е.В. Барило // Бионика интеллекта: науч.-техн. журнал. – 2011. – № 3 (77). – С. 107-111.

В статье рассматривается алгоритм оптимизации уровня селекции координат эталонных векторов-реализаций классов распознавания для заданного алфавита в рамках информационно-экстремальной интеллектуальной технологии, которая основана на максимизации информационной способности системы распознавания. При этом исследовано влияние уровня селекции на функциональную эффективность системы распознавания изображений.

Ил. 5. Библиогр.: 5 назв.

UDK 004.93’1

Optimization of selection level of the vectors-implementations coordinates in pattern recognition / K.V.Barylo // Bionics of Intelligense: Sci. Mag. – 2011. – № 3 (77). –P. 107-111.

In article algorithm optimization of the selection level of the vectors-implementations coordinate for the for a given class recognition alphabet by using information-extreme intellectual technology, which is based on maximizing the information capacity of the recognition system. The selection level of the pattern vectors-implementations coordinates influence on the system functional efficiency is estimated.

Fig.: 5/ Ref.: 5 items.

УДК 519.87

Бенаддия Абдельлатиф, О.Ф. Михаль
ХНУРЕ, г. Харьков, Украина, fuzzy16@pisem.net

ЛОКАЛЬНО-ПАРАЛЛЕЛЬНЫЙ СТЕКОВЫЙ АЛГОРИТМ БИНАРНОГО КЛЕТОЧНОГО АВТОМАТА

Приложение принципов локально-параллельной обработки информации к клеточным автоматам позволяет существенно увеличить размеры реализуемых моделей. Для полноты оперирования данными локально-параллельное представление данных должно быть дополнено алгоритмами, реализующими смену состояний ячеек для конкретных видов (вариантов) клеточных автоматов. Предложен и продемонстрирован на тестовых примерах один из таких алгоритмов — локально-параллельный стековый алгоритм бинарного клеточного автомата.

КЛЕТОЧНЫЕ АВТОМАТЫ, ЛОКАЛЬНАЯ ПАРАЛЛЕЛЬНОСТЬ, НЕЧЁТКАЯ ЛОГИКА, СТЕК

Введение

Актуальность проблематики, связанной с моделированием (описанием и предсказанием поведения) процессов, происходящих в *живой природе*, существенно возросла в связи с осознанием мировым сообществом в последнее десятилетие серьёзных взаимосвязанных проблем в столь разнородных направлениях, как экология, климатология, демография, социология. Перспективным аппаратом компьютерного моделирования крупноразмерных систем с распределённой обработкой информации, интерпретируемых применительно к указанным направлениям, являются *клеточные автоматы* (КА). Целесообразность применения КА для моделирования поведения систем указанного типа обусловлена простотой описания конфигурации и набора рабочих правил. Дополнительный выигрыш по объёму моделируемой системы и эффективности работы КА может быть обеспечен *локально-параллельными* (ЛП) алгоритмами [1]. Идеология ЛП алгоритмов совместима с принципами представления *нечёткой информации* [2, 3]. Для этого ключевой элемент нечёткого представления – *функция принадлежности* (ФП) – дискретизируется и масштабируется в соответствии с числом градаций, требуемым (достаточным) для описания конкретной системы. Ввиду дискретного характера и неотрицательности (по определению) значений ФП, ЛП аппарат нечёткого представления данных становится принципиально применимым также к КА. В частности, при чётком описании, как частном случае нечёткого, ФП принимает одно из двух значений (0, 1), что соответствует подвиду КА – *бинарным клеточным автоматам* (БКА) [4]. Для полноты оперирования данными необходимо дополнить аппарат описания [2, 3] ЛП алгоритмами, реализующими смену состояний ячеек КА для конкретных видов (вариантов) КА. Цель настоящей работы – разработка и исследование (оценка эффективности) ЛП варианта одной из процедур работы КА: стекового алгоритма определения уровня (степени) соседства ячеек БКА.

1. Клеточные автоматы

В основе КА [4] лежат следующие представления. Имеется регулярная решётка ячеек. Каждая ячейка может находиться в одном из конечного множества состояний. Для ячейки определено множество *соседних* ячеек – её окружение. Задаётся начальное состояние всех ячеек и набор правил перехода ячеек из одного состояния в другое. На каждой итерации, на основе правил перехода и в соответствии с текущим состоянием соседних ячеек, определяется новое состояние каждой ячейки. Таким образом, КА в целом последовательно сменяет состояния (этапы, фазы жизненного цикла, поколения).

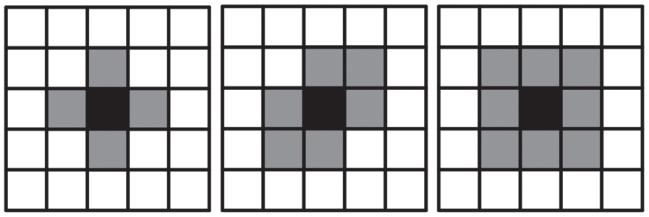
Характерно, что работа КА (переход в следующее поколение) необратима, как необратимы процессы в живой природе. Эволюционирование КА является однонаправленным, как и время в реальном физическом мире. Текущее j-е состояние КА однозначно определяет последующее $(j+1)$-е состояние, но $(j-1)$-е состояние не может быть однозначно восстановлено по текущему j-му состоянию. Таким образом, КА «автоматически» воспроизводят важную особенность реального мира – однонаправленность и необратимость времени.

Привлекательность КА в качестве аппарата моделирования крупноразмерных пространственно распределённых систем определяется также концептуальной наглядностью, малым набором рабочих правил и, как следствие, простотой программной реализации и визуализации процесса работы.

В БКА для записи состояния ячейки используется один бит. Подбором правил работы КА допустимы различные варианты соседства ячеек. Рис. 1 иллюстрирует некоторые из возможных топологий БКА: ситуации с 4, 6 и 8 ячейками-соседями. Центральная ячейка выделена чёрным цветом, её окружение (соседи) – серым.

Ориентированный граф, представленный на рис. 2, иллюстрирует совокупность правил для БКА с 8 соседями рис. 1 (в) для широко известного варианта бинарного КА «игра Жизнь» [5]. Ячейка

КА может находиться в одном из двух состояний: 0 или 1 (узлы графа). При уровне (степени) соседства 3 или 4 (при наличии в окружении 3 или 4 единичных ячеек) – единичное состояние сохраняется или 0 переходит в 1. Иначе – 0 сохраняется или 1 переходит в 0.



а *б* *в*

Рис. 1. Бинарные КА с вариантами соседства: 4 (*а*), 6 (*б*) и 8 клеток (*в*)

При компьютерном моделировании на КА содержательность (репрезентативность, информационная ёмкость) моделей растёт с их размерами. В частности, применительно к экологии и статистической биологии для моделирования поведения растительных ареалов или животных сообществ, с учётом информативности для статистической обработки [6], представляются целесообразными модели объёмом более 10^4 ячеек. В связи с этим при компьютерной реализации реальной модели на БКА возникает проблема экономного расходования ресурсов *вычислительной системы* (ВС) – оперативной памяти, дискового пространства, времени работы программы и др. Сокращение расходуемых системных ресурсов эквивалентно общему сокращению затрат на эксплуатацию модели. С другой стороны, экономное расходование ресурсов ВС при фиксированном общем объёме ресурсов – эквивалентно повышению размера или точности модели.

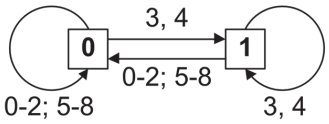


Рис. 2. Правила бинарного КА «игра Жизнь»

Каждая ячейка КА самостоятельно (независимо) изменяется в процессе работы программы. Изменения состояний i-й ячейки происходят, как отмечалось, в соответствии с правилами работы КА: ячейки влияют друг на друга, взаимно обусловливают свое поведение. При этом текущая информация о состоянии *i*-й ячейки должна храниться отдельно от информации о состоянии других ячеек. То есть, для каждой ячейки требуется отдельный элемент хранения. Стандартные форматы хранения данных – расточительны в отношении использования ресурсов ВС. Существенный выигрыш может быть обеспечен при ЛП представлении информации [2, 3].

2. Принцип локальной параллельности

Поясним принцип ЛП представления и обработки информации. Пусть имеется два n-компонентных вектора:

$$A: \{a_1, a_2, a_3,..., a_i,..., a_n\},$$
$$B: \{b_1, b_2, b_3,..., b_i,..., b_n\}, \quad (1)$$

где: $i \in (1, 2, 3,..., n)$; a_i, b_i –целые положительные числа, причём

$$a_i \le a_{max}; \quad b_i \le b_{max}; \quad a_{max} = b_{max}. \quad (2)$$

Условие (2), дополнительно налагаемое на (1), означает, что формат чисел – компонентов векторов – должен быть одинаковым и числа не могут быть сколь угодно большими.

Требуется найти покомпонентную сумму векторов:

$$C: \{ c_1, c_2, c_3,..., c_i,..., c_n \}; \quad c_i = a_i + b_i. \quad (3)$$

Согласно последовательной схеме, для этого должен быть выполнен следующий набор операций:

Шаг 1. Начальная установка: i=1.

Шаг 2. Извлечение значения операнда a_i из регистра памяти, в котором он хранится.

Шаг 3. Извлечение значения операнда b_i из регистра памяти, в котором он хранится.

Шаг 4. Суммирование: $c_i = a_i + b_i$.

Шаг 5. Помещение результата c_i в регистр памяти, в котором он должен храниться.

Шаг 6. $i = i + 1$. Если $i>n$, завершение вычисления; иначе переход к Шагу 2.

Согласно ЛП схеме, для нахождения покомпонентной суммы необходимо выполнить следующий набор операций:

Шаг 1. Конкатенация:

$$A:\{a_1, a_2, a_3,..., a_i,..., a_n\} \rightarrow$$
$$\rightarrow a_\# = (a_1 \oplus a_2 \oplus a_3 \oplus ... \oplus a_i \oplus ... \oplus a_n); \quad (4)$$
$$B:\{b_1, b_2, b_3,..., b_i,..., b_n\} \rightarrow$$
$$\rightarrow b_\# = (b_1 \oplus b_2 \oplus b_3 \oplus ... \oplus b_i \oplus ... \oplus b_n).$$

Шаг 2. Извлечение значения операнда (конкатенанда) $a_\#$ из регистра памяти, в котором он хранится.

Шаг 3. Извлечение значения операнда (конкатенанда) $b_\#$ из регистра памяти, в котором он хранится.

Шаг 4. Суммирование конкатенандов:
$c_\# = a_\# + b_\#$.

Шаг 5. Помещение результата $c_\#$ в регистр памяти, в котором он должен храниться.

Шаг 6. Деконкатенация:

$$c_\# = (c_1 \oplus c_2 \oplus c_3 \oplus ... \oplus c_i \oplus ... \oplus c_n) \rightarrow$$
$$\rightarrow C:\{c_1, c_2, c_3,..., c_i,..., c_n\}. \quad (5)$$

Операции конкатенации и деконкатенации требуют дополнительного пояснения. Обычно эти термины применяются при описании обработки

символьных последовательностей. В рассматриваемом случае конкатенируются целые положительные числа в двоичном представлении.

Числа (исходные данные, промежуточные значения, результаты) при их обработке располагаются в центральном процессоре в отдельных регистрах – линейках бинарных ячеек памяти. Отсюда применяемый далее термин – *регистровое представление* (РгП). В ЛП представлении результат конкатенации есть некоторое целое положительное число, помещаемое в едином регистре, – РгП, которое интерпретируется как состоящее из сегментов согласно длине конкатенантов. При деконкатенации сегменты извлекаются из РгП. Таким образом, конкатенация есть операция уплотнения представления компонентов векторов (1) в виде сегментов в РгП $a_{\#}$ и $b_{\#}$ вида (4). Содержимые РгП $a_{\#}$ и $b_{\#}$ обрабатываются целиком, как числа. Деконкатенация – обратная операция – раскрытие РгП. Содержимое i-го сегмента c_i извлекается и помещается в индивидуальный регистр. В результате, согласно ЛП схеме, операция (в рассматриваемом случае суммирование соответствующих компонентов векторов) производится над РгП, а результат C:$\{c_1, c_2, c_3,..., c_i,..., c_n\}$ выглядит так, как если бы операции производились над парами операндов раздельно.

Сравним представленные выше пошаговые последовательности операций. Вычислительные блоки операций Шаги 2 – 5 в последовательной и ЛП схемах совпадают. Различие состоит в том, что в последовательной схеме блок выполняется n-кратно, в ЛП схеме – однократно. Этим обеспечивается выигрыш в производительности. Разумеется, процедуры конкатенация и деконкатенация (Шаги 1 и 6 ЛП схемы) являются протяженными во времени, т.е. связаны с дополнительным расходованием ресурсов ВС. Однако, если в системе реализуются сложные вычисления, например с последовательным применением нескольких ЛП алгоритмов, РгП составляются из исходных данных только в самом начале вычислительного блока, а результаты извлекаются из РгП только в самом конце. Затраты ресурсов на конкатенацию-деконкатенацию при этом могут быть пренебрежимо малы по сравнению с общими затратами ресурсов ВС. При этом выигрыш в производительности приближается к n-кратному.

Очевидно, что ЛП схема тем более эффективна, чем больше число конкатенированных сегментов n. Приближенно (при больших объёмах вычислительных ЛП блоков), выигрыш в производительности пропорционален длине регистра вычислительного устройства и обратно пропорционален длине сегмента [2, 3].

Для единства структуры данных, разрядности сегментов, задействованных под операнды $a_{\#}$ и $b_{\#}$ и результат $c_{\#}$, должны совпадать. При выполнении операции соседствующие сегменты не должны взаимодействовать, поэтому должны быть приняты меры по согласованию разрядностей. Форматы представления ЛП данных в системе должны быть организованы так, чтобы в сегментах $c_{\#}$ было изначально зарезервировано достаточно места для размещения результата операции над $a_{\#}$ и $b_{\#}$.

Выигрыш в производительности и характеристики точности системы конкурируют между собой. При фиксированной разрядности регистра производительность ЛП системы тем выше, чем больше число сегментов. Следовательно, для повышения производительности целесообразно сократить разрядность сегментов до минимума. При этом уменьшается число градаций, с которым могут быть представлены в системе соответствующие обрабатываемые величины. Таким образом, при проектировании систем с использованием принципов ЛП при выборе размера сегментов должны приниматься компромиссные решения по соотношению производительности вычислений и точности результатов.

Для операций должны быть разработаны соответствующие алгоритмы, подобные ЛП алгоритмам операций нечёткой логики, рассмотренным в [2, 3]. Имеются ограничения, определяемые длиной сегментов. В частности, рассмотренная ЛП операция суммирования реализуема только для положительных операндов ограниченного размера. Этому условию удовлетворяют, в частности, значения ФП, применяемые в нечетком теоретико-множественном описании. Применительно к ЛП представлению они масштабированы в соответствии с размерами сегментов и конкатенированы в РгП [2, 3]. Для ячеек КА так же имеется фиксированное число неотрицательных состояний. Следовательно, ЛП представление информации потенциально применимо и к КА.

3. Локально-параллельное представление КА

Упрощённо процессор ВУ Фон-Неймановского типа имеет два 32-разрядных регистра для хранения операндов и один 32-разрядный регистр для помещения результата. Процессор реализует стандартный набор арифметических и логических операций, включая поразрядную (побитовую) логику и регистровые сдвиги.

Как отмечалось (п. 2), при ЛП обработке предполагается хранение нескольких элементов информации в виде раздельных непересекающихся сегментов РгП. В случае ЛП представления КА логично разместить в отдельных сегментах РгП отдельные клетки КА. При этом формат размещения информации в РгП должен быть удобным для реализации ЛП алгоритмов обработки.

В обычной записи для хранения решётки ячеек (матрицы) КА требуется двумерный массив. При

размере поля M строк на N столбцов требуется $M{\cdot}N$ индивидуально адресуемых элементов хранения информации. В случае 32-разрядного ВУ Фон-Неймановского типа это составляет $32{\cdot}M{\cdot}N$ бит. В ЛП представлении ячейки КА хранятся в сегментах РгП. То есть индивидуально адресуемыми элементами хранения становятся РгП, а не отдельные ячейка КА. В случае БКА в РгП может быть до 32 однобитовых сегментов. Строка поля БКА может быть помещена в $[M/32]+1$ РгП (квадратные скобки обозначают взятие целой части от деления). В результате для представления всего БКА необходим массив из $([M/32]+1){\cdot}N \approx M{\cdot}N/32$ РгП (индивидуально адресуемых элементов хранения информации). Соответственно, если ячейка КА имеет n состояний, для её представления требуется k бит с соблюдением соотношения $2^k \geq n$, то есть $k \geq log_2 n$, то есть $k = [log_2 n]+1$ бит. Таким образом, в 32-разрядном ВУ для представления строки элементов КА требуется $[M{\cdot}k/32]+1$, а для представления всего массива – $M{\cdot}N{\cdot}k/32$ индивидуально адресуемых элементов хранения информации.

Для БКА наиболее плотным (компактным, экономным, с минимальным расходованием ресурсов ВУ) является однобитовое сегментное представление. Например, поле БКА размером 30×30 элементов может быть размещено в одномерном массиве A из 30 штук 32-разрядных чисел в формате integer: $(a_1, a_2,..., a_j,..., a_{29}, a_{30})$. Сложность проявляется при реализации работы БКА. Как отмечалось, алгоритм функционирования КА предполагает рассмотрение окружения каждой клетки и принятие в зависимости от этого решения, будет ли клетка «живой» (значение «1») или «мёртвой» (значение «0») в следующем цикле жизни КА. В частности, для БКА с 8-ю соседями (рис. 1 а), согласно правилам (рис. 2), при ЛП вычислениях для j-й строки (j-го РгП a_j) должны суммироваться 8 штук РгП:

$$(a_{(j\text{-}1)}{<<}p) + (a_{(j\text{-}1)}) + (a_{(j\text{-}1)}{>>}p) + (a_j{<<}p) +$$

$$+ (a_j{>>}p) + (a_{(j+1)}{<<}p) + (a_{(j+1)}) + (a_{(j+1)}{>>}p). \quad (6)$$

В (6) «$<<p$» и «$>>p$» означают процедуры регистрового сдвига на p бит (длину сегмента) в сторону старших или младших разрядов соответственно. В рассматриваемом случае БКА $p = 1$, а для представления результата по каждому сегменту требуется 4 бита, поскольку имеется 9 градаций возможных значений сегментной суммы: (0, 1, 2,..., 7, 8). Как отмечалось, соседние сегменты не должны взаимодействовать. Для этого ЛП система должна быть организована так, чтобы в сегментах было изначально зарезервировано достаточно места для размещения результата. Типовым подходом при реализации подобных ЛП вычислений является прореживание РгП A [2, 3] с использованием битовых масок.

$$E_1 = (000100010001...0001)_2;$$
$$E_2 = (001000100010...0010)_2;$$
$$E_3 = (010001000100...0100)_2;$$
$$E_4 = (100010001000...1000)_2. \quad (7)$$

При этом массив РгП A преобразуется в 4 массива A_i,: $(a_{i1}, a_{i2},..., a_{ij},..., a_{i29}, a_{i30})$; $a_{ij} = a_j \wedge E_i$; $i \in (1, 2, 3, 4)$; где «$\wedge$» – побитовое логическое И. В полученных 4 массивах все сегменты содержатся в точности по одному разу. При этом они не взаимодействуют при проведении операции. После прореживания процедура выполняется последовательно для каждого из A_i, с последующим сведениием результатов в единый РгП – сложением либо теоретико-множественным объединением. Эффективность ЛП алгоритма при этом снижается четырёхкратно.

Аналогично, при 6 соседях (рис. 1 б) имеется 7 градаций, а при 4 соседях (рис. 1 в) – 5 градаций, поэтому для хранения каждого из таких результатов потребуется 3 бита. Соответственно, для ЛП обработки потребуется прореживание A на три массива, с применением системы битовых масок

$$E_1 = (001001001...001)_2;$$
$$E_2 = (010010010...010)_2;$$
$$E_3 = (100100100...100)_2. \quad (8)$$

Эффективность ЛП алгоритма снижается при этом только трёхкратно.

Прореживание системой масок (7) или (8) и соответствующее кратное снижение эффективности ЛП обработки снимается полностью с применением *стекового* ЛП алгоритма.

4. Стековый локально-параллельный алгоритм

Сортировка материала, попавшего в k-й сегмент стека, реализуется «методом пузырька». Поясним работу ЛП стекового алгоритма на примере. Пусть имеются 8-битные числа

$$A_1 = 188_{10} = 10111100_2,$$
$$A_2 = 113_{10} = 01110001_2,$$
$$A_3 = 235_{10} = 11101011_2,$$
$$A_4 = 25_{10} = 00011001_2, \quad (9)$$

случайным образом выбранные в интервале от 0 до 255. Нижний индекс обозначает основание системы счисления представления числа. В двоичном представлении для наглядности лидирующие нули сохранены.

Интерпретируем двоичные позиции, как сегменты в ЛП представлении. Таким образом, выражения (9) означают, что имеется четыре 8-сегментных РгП. Размер каждого сегмента – 1 бит. Стеками являются столбцы сегментов. Имеется ЛП набор из 8 стеков. Работа данного ЛП набора стеков заключается в том, что после первоначального заполнения единичные сегменты «проваливают-

ся» максимально вниз, а нулевые, соответственно, «всплывают» вверх согласно алгоритму сортировки «методом пузырька», который применяется к РгП посегментно, одновременно (локально-параллельно) ко всем сегментам. С набором данных (9) это выглядит следующим образом:

$$
\begin{array}{cccc}
\begin{matrix}10111100\\01110001\\11101011\\00011001\end{matrix} & \}\rightarrow\{ & \begin{matrix}10111100\\01110001\\00001001\\11111011\end{matrix} & \}\rightarrow \\
\rightarrow\{\ \begin{matrix}10111100\\00000001\\01111001\\11111011\end{matrix} & \}\rightarrow\{ & \begin{matrix}00000000\\10111101\\01111001\\11111011\end{matrix} & \}\rightarrow \\
\rightarrow\{\ \begin{matrix}00000000\\10111101\\01111001\\11111011\end{matrix} & \}\rightarrow\{ & \begin{matrix}00000000\\00111001\\11111101\\11111011\end{matrix} & \}\rightarrow \\
 & \rightarrow\{ & \begin{matrix}00000000\\00111001\\11111001\\11111111\end{matrix} &
\end{array}
\qquad (10)
$$

Выражение (10) изображает последовательную смену 7 состояний набора РгП (1). Первое состояние соответствует исходному (1), последнее – ситуации с полностью отсортированными сегментными стеками: все единицы максимально смещены вниз. Сортировка (ЛП работа стеков) организована за 6 шагов – переходов между состояниями. Фигурные скобки и стрелки переходов показывают, какие именно из РгП сопоставляются на каждом следующем шаге алгоритма сортировки.

Рассмотренный процесс включает два вложенных цикла. Внешний цикл обеспечивает (n-1)-кратное (где n – число РгП, в примере $n = 4$) повторение внутреннего цикла с сокращением его объёма от (n-1) (три первые смены состояний) до 1 (последняя смена состояний).

Первый (самый правый, соответствующий младшим разрядам чисел $A_1 - A_4$) стек не претерпевал изменений, т.к. оказалось, что изначальная расстановка единиц в нём не требует сортировки. 4-е и 5-е состояния – тождественны, поскольку в результате сопоставления РгП A_3 и A_4 оказалось, что в этом цикле не происходит вертикального перемещения единиц («всплывания пузырьков»). Показателен 3-й справа разряд. В нём тремя последовательными обменами единица спускается из РгП A_1 в РгП A_4.

Рассмотрим бинарную процедуру сопоставления, обозначенную на (10) значком «→». Процедура включает арифметические операции «–» и «+», а так же побитовые (поразрядные) логические операции «END» и «XOR». Входными данными являются A и B, выходными – изменённые значения A и B. В процедуре используются константа $E_0 = 11111...11$ («ЛП единица») и промежуточная переменная M. Приведём описание процедуры в обозначениях, принятых в [7]:

```
Local_Parallel_Stack()
1  M ← A “END” ( B “XOR” E0 )          (11)
2  A ← A – M
3  B ← B + M
```

В ходе выполнения в посегментном битовом представлении для всех четырёх возможных комбинаций входных данных происходит следующее.

Исходное состояние:

$A = ...1...1...0...0...$;
$B = ...1...0...1...0...$.

Константа:

$E_0 = ...1...1...1...1...$.

Промежуточные состояния:

B “XOR” $E_0 = ...0...1...1...0...$;
$M = A$ “END” (B “XOR” E_0) $= ...0...1...0...0...$.

Состояние после завершения процедуры:

$A = ...1...0...0...0...$;
$B = ...1...1...1...0...$.

Как следует из представленного, в тех сегментных стеках, в которых имеется комбинация (1, 0), происходит преобразование (1, 0) → (0, 1), то есть при наличии «свободного места» единица «проваливается» вниз по стеку. Остальные стековые комбинации остаются без изменения.

После завершения сортировки стеков (столбцов текущего набора РгП (10)), имеется весь исходный материал для формирования результата – нового набора значений соответствующего фрагмента КА. Так, если рассматривать пример (10) как относящийся к работе БКА с четырьмя соседями (рис. 1 а), разность

$$(A_2 - A_1) = (00111001)_2 - (00000000)_2 = (00111001)_2$$

будет соответствовать правилу наличия трёх «живых» ячеек-соседей. В ней единицами отмечены те позиции (ячейки КА), которые «выжили» в новом поколении КА, поскольку в текущем поколении они были окружены тремя единичными соседями. Соответственно, нулями обозначены «не выжившие» ячейки, которые в текущем поколении были окружены меньшим либо большим числом соседей.

Аналогично, выглядит ситуация с другим числом соседей (рис. 1 б, в) и более сложным набором правил. Так, в случае БКА с 8 соседями (рис. 1 в) и «выживанием» ячейки при 3 или 4 соседях (рис. 2) стек состоит из восьми РгП $B_1, B_2,..., B_m,..., B_8$. Здесь, как и в (9), B_8 – основание стека, B_1 – его вершина. При упорядочении РгП аналогично представленному в (10), после прохождения сортировки в стеках, значение «1» в некоторой позиции m-го РгП обозначает наличие у соответствующей

ячейки не менее $(9-m)$ «единичных» ячеек-соседей. Поэтому ситуация с 3 или 4 «соседями» отображается в виде «1» в B_6 и B_5, соответственно, при наличии «0» в B_5 и B_4, соответственно. Результирующее выражение данного комбинированного решающего правила – $(B_4 - B_6)$.

5. Обсуждение результатов

Работа БКА с применением предложенного стекового алгоритма смоделирована в варианте с 8 соседями (рис. 1 а) на 32-разрядном процессоре, на языке Python. Размер поля модели 32×32 ячейки. Моделированием подтверждена работоспособность стекового алгоритма. В процессе разработки выявлены следующие особенности стекового варианта построения КА.

Позитивной стороной ЛП стекового алгоритма КА является универсальность подхода применительно к различным вариантам правил поведения КА. Универсальность проявляется в независимости от объёма и конфигурации окружения (соседства), а так же в простоте выбора порогового уровня или диапазона уровней при определении «выживающих» ячеек. Этим обеспечивается простота смены значений для ключевых параметров модели (числа ячеек-соседей, конфигурации соседства и правил перехода разметки ячеек) при экспериментальном подборе при исследовании применительно к конкретной моделируемой предметной области.

Достоинством стекового варианта алгоритма КА является существенное (в 3 – 4 раза) сокращение времени обработки по сравнению с рассмотренным (п. 3) альтернативным ЛП вариантом – с прореживанием РгП.

Ограничением выполненного моделирования ЛП стекового варианта БКА является размер поля: строка поля целиком помещена в одно РгП. Для размещения более длинных строк требуется несколько РгП и дополнительные алгоритмические решения по состыковке их концов (младшего сегмента i-й РгП и старшего сегмента $(i+1)$-й РгП) для правильной обработки соседства ячеек.

В разработанном варианте стековый алгоритм детерминировано реализует два вложенных цикла. В результате, возможны, хотя и редки, случаи «холостого» прохождения цикла, типа отмеченного выше, продемонстрированного в примере (10) – тождественность 4-го и 5-го состояний. Сокращение количества переборов за счёт изъятия этих «холостых» прохождений цикла – проблематично. Введение соответствующих дополнительных проверок только усложнит и замедлит алгоритм, поскольку возможные проверки в совокупности будут в лучшем случае соизмеримы по сложности и объёму кода с основной процедурой (11).

В связи со сказанным, последующими направлениями исследования могут явиться, в частности:

– модифицирование алгоритма на случай представления строки поля КА несколькими РгП;

– поиск оптимизационных алгоритмических решений, в частности по сокращению «холостых» циклов;

– распространение стекового алгоритма на более широкий класс КА (большее число состояний ячейки, конфигурации соседства, отличные от иллюстрируемых рис. 1, многомерные и многослойные структуры с послойными функциями влияния).

Кроме того ретроспективно целесообразны обобщения КА совместно с принципом ЛП и стековым подходом в реализации базового алгоритма:

– введение весовых коэффициентов для уровней (при задании уровня срабатывания в виде диапазона значений) и отдельных полей конфигурации окружения;

– интерпретация вводимых весовых коэффициентов как значений ФП;

– вероятностная интерпретация ФП.

Введение вероятностных представлений и элементов нечёткости – позволяет существенно разнообразить поведение КА и искать их устойчивые долгоживущие состояния (в смысле выживаемости КА) помимо зависимости от конкретных правил поведения КА.

Применительно к приложениям типа моделирования различных аспектов функционирования живой природы нечёткий и вероятностный КА подходы принципиально интересны тем, что ими вводится (учитывается, изначально задаётся) мера детерминированности (известности, измеримости, наблюдаемости) в поведении моделируемой системы. Таким образом, не исключаются из рассмотрения неизвестные влияющие факторы, всегда присутствующие в моделируемой системе.

Выводы

Локально-параллельное представление информации обеспечивает повышенную эффективность работы программного обеспечения. Принципы локально-параллельной обработки применимы к клеточным автоматам и позволяют существенно увеличить размеры модели. Предложен и продемонстрирован на тестовых примерах локально-параллельный стековый алгоритм бинарного клеточного автомата. Стековый подход в равной степени эффективен применении к различному числу и вариантом расположения ячеек-соседей; а также инвариантен по выбору порогового уровня или диапазона уровней для определения «выживающих» ячеек.

Список литературы: 1. *Бенаддия, Абдельлатиф.* Перспективы реализации келлюлярных авторматов на локально-параллельных алгоритмах [Текст] / *Бенаддия Абдельлатиф* // Материалы международной научно-практической

конференции студентов и аспирантов “Информационные технологии в экономике и образовании”. – М.: Российский университет кооперации, 2008. – С. 45–48. **2.** *Михаль, О.Ф.* Моделирование распределенных информационно-управляющих систем средствами локально-параллельных алгоритмов обработки нечеткой информации [Текст] / *О.Ф. Михаль* // Проблемы бионики: Всеукр. межвед. науч.-техн. сборник. – 2001. – Вып. 54. – С. 28–34. **3.** *Михаль, О.Ф.* Принципы организации систем нечеткого регулирования на однородных локально-параллельных алгоритмах [Текст] / *О.Ф. Михаль, О.Г. Руденко* // Управляющие системы и машины. 2001. № 3. С. 3–10. **4.** *Тоффоли, Т.* Машины клеточных автоматов [Текст] : Пер. с англ. / *Т. Тоффоли, Н. Марголус.* – М., «Мир» 1991. – 280 с. **5.** *Люггер, Д.Ф.* Искусственный интеллект: стратегии и методы решения сложных проблем [Текст] / *Д.Ф. Люггер* М.: Изд. дом «Вильямс», 2005. – 864 с. **6.** *Любищев, А.А.* Дисперсионный анализ в биологии [Текст] / *А.А. Любищев.* – М.: Изд-во Моск. ун-та, 1986. 200 с. **7.** *Кормен, Т.* Алгоритмы: построение и анализ [Текст] / *Т. Кормен, Ч. Лейзерсон, Р. Ривест* М.: МЦНМО, 2001. – 960 с.

Поступила в редколлегию 12.09.2011

УДК 519.87

Локально-паралельний стековий алгоритм бінарного клітинного автомату / Бенаддія Абдельлатіф, О.П. Міхаль // Біоніка інтелекту: наук.-техн. журнал. – 2011. – № 3 (77). – С. 112-118.

Локально-паралельне подання інформації забезпечує підвищення эфективности работи програмного забезпечення. Додання принципів локально-паралельної обробки до клітинних автоматів дозволяє суттєво підвищити розміри моделей. Запропоновано і продемонстровано на тестових прикладах локально-паралельний стековий алгоритм бінарного клітинного автомату.

Іл. 2. Бібліогр.: 7 найм.

UDK 519.87

Local-parallel stack algorithm of the binary cellular automaton. / Benaddia Abdellatif, O.Ph. Mikhal // Bionics of Intelligense: Sci. Mag. – 2011. – № 3 (77). –P. 112-118.

Local-parallel presentation of information provides raised efficiency of the work of software. Application of principle of local-parallel processing to cellular automaton allows greatly enlarge the sizes of models. Local-parallel stacked algorithm of the binary cellular automaton is offered and demonstrated on test examples.

Fig. 2. Ref.: 7 items.

УДК 519.87

Мохамад Али, О.Ф. Михаль
ХНУРЕ, г. Харьков, Украина, fuzzy16@pisem.net

ЛОКАЛЬНО-ПАРАЛЛЕЛЬНАЯ СОРТИРОВКА ОГРАНИЧЕННЫХ МАЛЫХ НАБОРОВ ДАННЫХ

Рассмотрены принципы локально-параллельного представления информации применительно к задаче сортировки данных. Процедура сортировки проанализирована на комбинаторном уровне на примере 3-, 4- и 8-элементных числовых последовательностей. Описан принцип работы алгоритма локально-параллельной сортировки. Моделированием на языке Python показано, что локально-параллельный алгоритм максимально эффективен применительно к малым выборкам, суммарным размером в пределах разрядности процессора.

ЛОКАЛЬНАЯ ПАРАЛЛЕЛЬНОСТЬ, СОРТИРОВКА ДАННЫХ

Введение

Сортировка данных – классическая задача теории алгоритмов, сочетающая в себе прозрачность общей концепции и простоту постановки с многовариантностью решений в зависимости от дополнительных условий, налагаемых в связи с конкретными особенностями обрабатываемой информации. Алгоритмы сортировки подробно изучены для *вычислительных систем* (ВС) последовательного действия. Параллельные ВС до недавнего времени были исключительно многопроцессорными, дорогими и громоздкими, поэтому они разрабатывались преимущественно для специальных применений, а задачи сортировки рассматривалась в них главным образом в контексте реализации общих принципов распараллеливания алгоритмов применительно к большим массивам данных. Ситуация изменилась в последние десятилетия с появлением многоядерных процессоров и широким распространением (массовым использованием) сетевых вычислительных структур [1]: приобрела актуальность обработка *ограниченных малых наборов данных* (ОМНД). Задача сортировки применительно к ОМНД [2, 3] эффективно решается с использованием алгоритмов, реализующих принципы *локальной-параллельной* (ЛП) обработки информации [4, 5]. Область применения ОМНД – задачи принятия решений при детерминированном наборе вариантов. К задачам этого типа относится, в частности, значительная часть тематики *искусственного интеллекта* (ИИ). Так, одна из парадигм ИИ – исследование и воспроизведение структур и функций человеческого интеллекта. Человек же в каждый конкретный момент своей практической деятельности эффективно оперирует *ограниченным малым* числом понятий [6]. Эффективность (скорость принятия решений) зачастую определяется правильной последовательностью рассмотрения вариантов, а реализация эффективности сводится к сортировке ОМНД. ЛП алгоритмы обеспечивают повышение скорости обработки без задействования дополнительных аппаратных ресурсов. Поэтому они целесообразны как структурный элемент при реализации обработки ОМНД. Сказанным определяется актуальность и практическая ценность данного направления.

В настоящей работе развиваются рассмотренные ранее [1-3] принципы сортировки на ЛП алгоритмах, проводится детальное сравнение с последовательными алгоритмическими процедурами и формулируются направления использования выявленных закономерностей в прикладных задачах.

1. Гносеологический аспект

Рассматриваемый предмет имеет глубокие корни в природе человеческого интеллекта. Эффективная человеческая деятельность характеризуется, в частности:

– ограниченным (не слишком большим) количеством одновременно выполняемых (отслеживаемых) заданий (процессов, объектов, дел, работ);

– правильным определением приоритетности (важности, последовательности) выполнения заданий.

Деятельность с избыточным числом выполняемых заданий, или с плохо организованной приоритетностью, как правило, малоэффективна. Данные аспекты базируются на ограниченности ресурсов реализации человеческого интеллекта. Факт, известный в психологии [6]: средний человек способен эффективно взаимодействовать (удерживать в памяти, отслеживать изменения, мысленно оперировать) одновременно не более, чем с 5–7 объектами (предметами, понятиями, блоками информации). Данная закономерность обусловлена ограниченным числом информационных каналов взаимодействия с объектами, а также ограниченными ресурсами параллельной обработки информации (мыслительной деятельности) по планированию выполнения действий с надлежащей приоритетностью.

Данная особенность человеческого интеллекта имеет отношение к компьютерной технике и информационным технологиям в следующем аспекте. Практика развития компьютерных си-

стем (hardware и software, включая ИИ) такова, что явно или неявно — мерилом прогресса в этой области является степень соответствия человеческому интеллекту. Данное положение соответствует, в частности, известному тесту Тюринга [7]. Исходя из этого, при разработке ВС целесообразно изучать «прототип» (hardware – человеческий мозг и software – человеческую психику) и следовать тенденциям, предначертанным в этой «уже разработанной» «эталонной» и «априорно совершенной» системе, являющейся «технической базой», на которой реализован человеческий интеллект. Известно, что «прототип» «аппаратно реализован» на естественных (живых) *нейронных сетях* (НС). Основной особенностью обработки информации на НС и ключевым преимуществом НС является *параллельность*. Параллельность реализуема в настоящее время преимущественно на многопроцессорных и многоядерных вычислительных структурах, а также в виде распределённых (локально- и глобально-сетевых) ВС. Аппаратная база искусственных НС развита в настоящее время в существенно меньшей степени. Исходя из этого, наряду с развитием аппаратных искусственных НС, целесообразны исследования и разработки в области алгоритмов, поддерживающих параллельную обработку информации на имеющихся наиболее развитых на настоящий момент типах аппаратного обеспечения. Перспективны ЛП алгоритмы, которыми распараллеливание обработки информации может быть реализовано на ВС последовательного действия (работающих в рамках концепции Фон-Неймана), то есть на компьютерах общего назначения. Таким образом, с использованием принципа ЛП параллельные вычисления претерпевают некоторое качественное изменение. Соответственно, требуют пересмотра в рамках концепции ЛП также и классические процедуры сортировки данных [8].

2. Основные определения

Будем рассматривать набор данных A как множество, состоящее из элементов:

$$A: (a_1, a_2, a_3,..., a_{i-1}, a_i, a_{i+1},..., a_{n-2}, a_{n-1}, a_n). \quad (1)$$

Каждое из данных, соответствующее $a_i, \in A$, $i \in (1, 2,...., n)$, – объект произвольной природы: число, фрагмент текста, запись в БД, картинка, фрейм и др. При этом элементы множества A могут быть самими объектами набора данных, либо указателями на них, позволяющими взаимнооднозначно соотносить $a_i, \in A$ и соответствующий объект. В первом случае операции, производимые над объектами множества A, связаны с перемещением самих объектов (данных) в памяти, во втором – объекты непосредственно не перемещаются, а операции над ними сводятся к работе с указателями. С точки зрения рассматриваемых далее алгоритмов ЛП сортировки данное различие не существенно.

Термин «ограниченный» предполагает конечное число n элементов, неизменное в процессе сортировки; термин «малый» – количество элементов, при котором не требуется использование видов (типов) памяти, *существенно* замедляющих информационный обмен в процессе выполнения сортировки. Разумеется, интерпретация термина «малый» может быть уточнена с учётом конкретной архитектуры ВС.

Отношение порядка (наличие *упорядоченности*) на множестве A предполагает указание для *некоторых* элементов $a_i, a_j \in A$, $i, j \in (1, 2, 3,..., n)$, $i \neq j$, одного из следующих соотношений: $a_i \prec a_j$, $a_i \succ a_j$. Символы «$\prec$» и «$\succ$» есть отношения следования: «предшествует» и «следует за». В частности они могут интерпретироваться как арифметические соотношения между числами – бинарные отношения «меньше» (<) и «больше» (>). При этом интерпретацией арифметического соотношения $a_i = a_j$ может быть «произвольное следование»: справедливо любое из отношений следования $a_i \prec a_j$, или $a_i \succ a_j$.

Множество A является *упорядоченным*, если отношение порядка определено для *любых* $a_i, a_j \in A$. Множество A является *частично упорядоченным*, если отношение порядка определено только для *некоторых* из пар $a_i, a_j \in A$. Элементы, для которых отношения порядка не определены, являются *несоизмеримыми*. Множество является *неупорядоченным*, если ни для одной пары элементов не определено отношение порядка.

Будем различать *множество* как совокупность элементов (1) и *последовательность* элементов множества как выборку некоторых элементов множества с учётом порядка их размещения. Для *упорядоченной* последовательности, включающей все элементы множества A (1), в арифметической интерпретации с учётом замечания о соотношении $a_i = a_j$ справедливо одно из выражений:

$$a_1 \le a_2 \le a_3 \le ... \le a_{i-1} \le a_i \le a_{i+1} \le ... \le a_{n-2} \le a_{n-1} \le a_n; \quad (2)$$

$$a_1 \ge a_2 \ge a_3 \ge ... \ge a_{i-1} \ge a_i \ge a_{i+1} \ge ... \ge a_{n-2} \ge a_{n-1} \ge a_n. \quad (3)$$

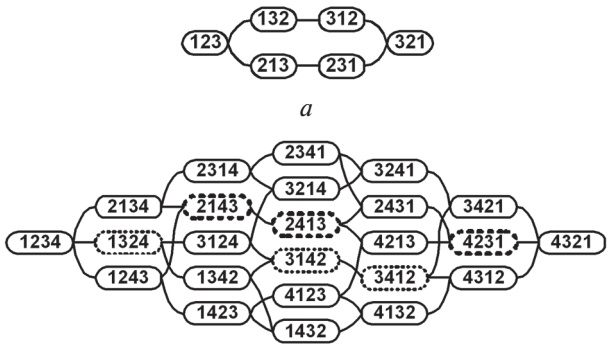
Последовательность является упорядоченной *по возрастанию* (2) (соответственно, *по убыванию* (3)), если для любой пары её элементов справедливо $a_i \le a_j$ (соответственно, $a_i \ge a_j$), где $i, j \in (1, 2, 3,..., n)$, $i < j$. Если хотя бы для одной пары элементов условие $a_i \le a_j$ ($a_i \ge a_j$) не выполняется, последовательность *не является упорядоченной* по возрастанию (убыванию), то есть является *неупорядоченной по возрастанию* (*неупорядоченной по убыванию*). Таким образом, последовательность, упорядоченная по возрастанию, является неупорядоченной по убыванию; последовательность, упорядоченная по

убыванию, является неупорядоченной по возрастанию. То есть упорядоченности по возрастанию и убыванию взаимно-обратно являются неупорядоченностями. Данные и подобные представления (упорядочение, разупорядочение или переупорядочение) продуктивны применительно к задачам сортировки ограниченных малых наборов элементов. Множество может быть изначально разупорядоченным. Процесс упорядочения – переход от одной последовательности (расстановки) элементов к другой. Может быть введена *мера разупорядоченности* множества как число перестановок соседних элементов, которое необходимо произвести чтобы прийти к упорядоченной последовательности элементов.

3. Упорядочение элементов

С учётом введенного понятия меры разупорядоченности, может быть показано, что множества, упорядоченные по возрастанию и убыванию, взаимно максимально разупорядочены по отношению друг к другу. В частности, для данного множества число перестановок в парах соседствующих элементов является максимальным при переупорядочении множества из упорядоченности по возрастанию в упорядоченность по убыванию, или наоборот: (2) → (3) или (3) → (2)

Графы, представленные на рис. 1 а, б иллюстрируют сказанное для множеств из трёх и четырёх *разных* элементов. Термин «разный» понимается в значении $a_i, \neq a_j$; $a_i, a_j \in A$; $i, j \in (1, 2, 3,..., n)$; $i \neq j$.



а

б

Рис. 1. Структуры частично упорядоченных множеств перестановок трёхэлементной (*а*) и четырёхэлементной (*б*) последовательностей.

При рассмотрении цепочек перестановок необходимо исключить циклы, ограничившись только продуктивными перестановками. Графы рис. 1 а, б позволяют проследить различные варианты таких цепочек, поскольку охватывают все возможные варианты связей по перестановкам в парах соседствующих элементов. Так, для 4-элементной последовательности (рис. 1 б) одна из цепочек преобразования (1, 2, 3, 4) → (4, 3, 2, 1) имеет следующий вид:

$$
\begin{array}{ccccccc}
 & & \lceil\rceil & & \lceil\rceil & & \\
 & & (1,2,3,4) & \rightarrow & (2,1,3,4) & \rightarrow & \\
 & & \lceil\rceil & & \lceil\rceil & & \\
\rightarrow & & (2,1,4,3) & \rightarrow & (2,4,1,3) & \rightarrow & \\
 & & \lceil\rceil & & \lceil\rceil & & \\
\rightarrow & & (4,2,1,3) & \rightarrow & (4,2,3,1) & \rightarrow & \\
\rightarrow & & (4,3,2,1) & & & &
\end{array}
\qquad (4)
$$

В (4) любые две соседние последовательности, разделённые знаком « → », различаются перестановкой местами одной (обозначенной) пары двух соседних элементов (цифр). Другая соседствующая пара, если она имеется и может быть переставлена, тем не менее остаётся незадействованной, поскольку процедура сортировки – последовательная. В результате, (1, 2, 3, 4) → (4, 3, 2, 1) реализуется за 6 шагов. Эта ситуация имеет место не только для (4), но и для любых вариантов цепочек перестановки элементов, что может быть прослежено непосредственно по графу (рис. 1 б), на котором имеется 7 вертикальных столбцов расположения 4-элементных последовательностей, различающихся перестановкой одной пары элементов.

В отличие от последовательной процедуры сортировки, ЛП процедура поддерживает обмен местами сразу нескольких (всех имеющихся) пар соседствующих элементов последовательности. Вследствие этого, как будет показано далее, число переходов « → » в процедуре сортировки может быть существенно сокращено.

4. Локальная параллельность

ЛП обработка информации на процессорах общего назначения допускает широкий круг приложений. Принцип ЛП обработки информации может быть проиллюстрирован на следующем вычислительным примером. Имеется n выражений $c_i = a_i + b_i$; $i = 1, 2,..., n$. Значения a_i и b_i заданы. Требуется найти значения c_i. Если используется ВС с одним процессором, задача разрешима за $4n$ шагов: загрузить a_i, загрузить b_i, произвести суммирование, выгрузить результат c_i. Если a_i и b_i положительные, имеют ограниченное число разрядов и если в результате сложении разрядность не нарушается (числа a_i, b_i и c_i имеют одинаковую разрядность), исходные данные могут быть представлены в виде:

$$A = a_1 \oplus a_2 \oplus ... \oplus a_i \oplus ... \oplus a_n;$$
$$B = b_1 \oplus b_2 \oplus ... \oplus b_i \oplus ... \oplus b_n, \qquad (5)$$

где символом $\oplus$ обозначена *конкатенация*. Если разрядность результата превышает разрядность слагаемых для переноса избыточного разряда при суммировании в формате представления данных требуется предусмотреть разделительные нули. В обоих случаях результат $C = A + B$ может быть интерпретирован как конкатенация аналогично (5), в которой сегменты – числа c_i – соответствуют

по формату a_i и b_i. При этом *весь* набор значений c_i может быть получен за 4 шага. Здесь n-кратный выигрыш достигнут без учёта «окаймляющих» операций конкатенации-деконкатенации, предназначенных для формирования A и B и извлечения результатов c_i из C. Если в реальном случае вычислительный блок достаточно сложный и включает последовательное применение нескольких разнообразных операций без промежуточных конкатенаций-деконкатенаций, выигрыш может быть значительным. Выигрыш растёт с ростом разрядности процессора и с сокращением размеров конкатенируемых сегментов. При этом снижается точность представления данных (система «загрубляется»), что в ряде конкретных приложений бывает допустимо.

Рост разрядности процессоров является долговременной устойчивой тенденцией развития процессорной техники. Известный закон Мура об удвоении числа транзисторов на кристалле на практике означает удвоение разрядности регистров процессора. Разрядность регистров, как известно, это объём непосредственно адресуемой оперативной памяти, т.е. допустимая размерность задач, разрешимых в данном ВС за приемлемое время. Поэтому разрядность вероятно, будет наращиваться и в дальнейшем, поскольку при этом стимулируется появление новых вычислительных задач, которые в свою очередь стимулируют рост требований к техническим характеристикам вновь разрабатываемых ВС. Данная «обратная связь» исправно работает на протяжении всей истории развития ВС и нет причин предполагать отход от данной тенденции. Соответственно, есть основания прогнозировать рост эффективности (производительности) применения ЛП алгоритмов за счёт увеличения числа конкатенируемых сегментов.

5. ЛП сортировка

Проиллюстрируем работу алгоритма ЛП сортировки ОМНД следующим образом. Пусть имеется множество элементов, на котором определено отношение порядка:

$$A: \{a_1, a_2, a_3, a_4,..., a_n\};\quad a_1 > a_2 > a_3 > a_4 >...> a_n. \quad (6)$$

Пусть из элементов множества A составлена *разупорядоченная* последовательность:

$$B: (b_n, b_{(n-1)},..., b_2, b_1),\ b_i = a_j,\ i,j \in \{1, 2,..., n\},$$
$$\exists\, (i, j): i \neq j. \quad (7)$$

В частности, в максимально разупорядоченном случае (7) может быть обратно упорядоченным набором элементов из (6): B: (a_n a_{n-1}, a_{n-2}, a_{n-3}, a_{n-4},..., a_3, a_2, a_1).

Требуется упорядочить последовательность (7) в порядке убывания величины элементов от младшего (правого) к старшему (левому), т.е. привести к упорядочению согласно (6).

Производим конкатенацию (5) элементов b_i согласно их расположению в последовательности (7). Далее в вербальном описании ЛП процедура сортировки включает следующие 4 шага.

Шаг 1. Скопировать текущее B для последующего сравнения: $B^1 = B$.

Шаг 2. Сравнить попарно нечётные элементы B с чётными, стоящими слева от них. В каждой сравниваемой паре больший из элементов поставить на нечётное место, а меньший – на чётное.

Шаг 3. Сравнить попарно чётные элементы B с нечётными, стоящими слева от них. В каждой сравниваемой паре больший из элементов поставить на чётное место, а меньший – на нечётное.

Шаг 4. Сравнить B и B^1. Если $B^1 = B$, ЛП процедура сортировки закончена. Результат содержится в B. Если $B^1 \neq B$ – перейти к Шагу.1.

Процедура иллюстрируется рисунком 2. Разделение сегментов на чётные и нечётные организовано посредством пары регистров $R1$ и $R2$. Заштрихованы фрагменты регистров, в которых расположены информационные сегменты. Разделение (прореживание) конкатенации вида (4) на четный ($R2$) и нечётный ($R1$) регистры программно реализуется применением процедуры побитового «И» с использованием пары битовых полумасок вида:

$$E1:\{...(11...1)(00...0)(11...1)(00...0)(11...1)\};$$
$$E2:\{...(00...0)(11...1)(00...0)(11...1)(00...0)\}. \quad (8)$$

Сегменты, соответствующие конкатенированным данным в (5), выделены в (8) круглыми скобками.

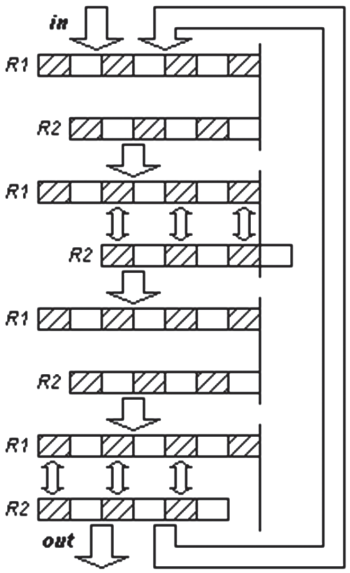


Рис. 2. Работа ЛП алгоритма сортировки

Регистры последовательно совмещаются для сопоставления (Шаги 2 и 3) применением регистровых сдвигов на число бит, равное длине сегмента. Посегментное сопоставление показано двунаправленными прозрачными стрелками. Однонаправленные прозрачные стрелки показывают общее

прохождение ЛП процедуры по Шагам 1 – 4. Загрузка исходных данных и вывод результата обозначены *in* и *out*.

6. Чередование обменов

Основные черты ЛП сортировки ОМНД согласно алгоритму (рис. 2) могут быть прослежены на примере пересортировки набора из 8 элементов:

$$(7, 6, 5, 4, 3, 2, 1, 0) \rightarrow (0, 1, 2, 3, 4, 5, 6, 7). \quad (9)$$

Соответствующая процедура представлен на рис. 3.

Существенным (и тривиальным) ограничением при выполнении обычной (не ЛП) процедуры сортировки является последовательный характер выполнения действий. Как отмечалось, в каждом акте сопоставления рассматривается исключительно одна пара соседствующих элементов (a_i, a_{i+1}). Предшествующие (a_1, a_2), (a_2, a_3),..., (a_{i-1}, a_i) и последующие (a_{i+1}, a_{i+2}), (a_{i+2}, a_{i+3}),..., (a_{n-1}, a_n) пары остаются вне рассмотрения. Таким образом проделывается n-1 однократных операций сопоставления. При этом, когда одна пара сопоставляется, n-2 не задействованы. Введением ЛП обработки эффективность существенно повышается: параллельно проводится ещё до $(n/2)$-1 сопоставлений.

Нумерация шагов процесса дана в левой колонке. Нижняя строка – результат сортировки, полученный после ЛП операций обмена в предпоследней строке.

Строками двунаправленных стрелок над соответствующими нумерованными строками обозначены ЛП операции обмена между сегментами. Каждая строка стрелок есть одна ЛП операция. Как показано, инвертирование последовательности реализуется за n шагов. В примере (рис. 3) в нечётных шагах процесса осуществляется обмен между чётными и нечётными сегментами (чётный сегмент стоит *слева* от нечётного), а в чётных строках – обмен между нечётными и чётными сегментами (чётный сегмент стоит *справа* от нечётного).

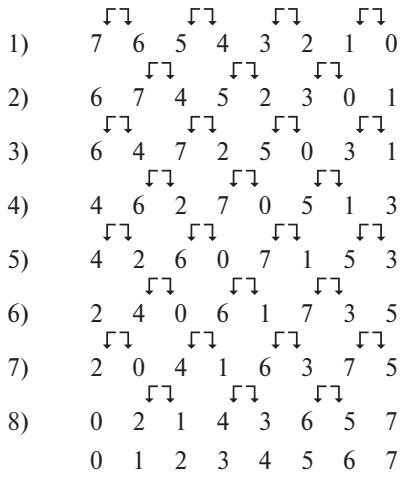


Рис. 3. ЛП сортировка последовательности из 8 элементов

В примере (рис. 3) ЛП сортировка начата со строки обменов «чётный–нечётный». Возможен также другой порядок: начало сортировки со строки «нечётный–чётный». При изменении порядка чередования обменов общее число шагов сортировки не изменяется: полная ЛП пересортировка 4 элементов (1, 2, 3, 4) → (4, 3, 2, 1) реализуется за равное число шагов.

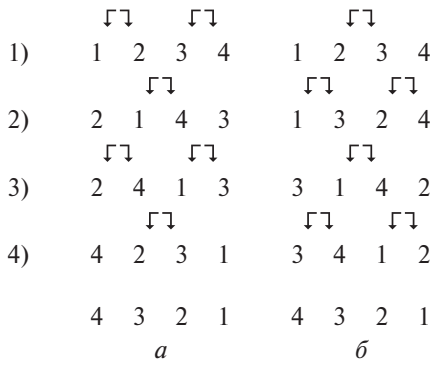


Рис. 4. ЛП сортировка последовательности из 4 элементов в двух вариантах: начиная с «чётный–нечётный» (*а*) и «нечётный–чётный» (*б*)

Результат сортировки также остаётся прежним, поскольку изменение касается только фаз сортировки по чередованию чётностей.

Пример (рис. 4) соответствует графу на рис. 1 б. При этом ЛП алгоритм преобразования (1, 2, 3, 4) → (4, 3, 2, 1) реализует пару цепочек (10, 11) типа (4), но с сокращённым числа промежуточных состояний:

$$(1, 2, 3, 4) \rightarrow (2, 1, 4, 3) \rightarrow (2, 4, 1, 3) \rightarrow \rightarrow (4, 2, 3, 1) \rightarrow (4, 3, 2, 1) \quad (10)$$

$$(1, 2, 3, 4) \rightarrow (1, 3, 2, 4) \rightarrow (3, 1, 4, 2) \rightarrow \rightarrow (3, 4, 1, 2) \rightarrow (4, 3, 2, 1). \quad (11)$$

Выражение (10) соответствует рис. 4 а – ЛП сортировка начинается с сопоставления «чётный–нечётный»; выражение (11) – рис. 4 б – начало сортировки с «нечётный–чётный». На графе рис. 1 б элементы (10) (кроме исходного и конечного) выделена жирным пунктиров, элементы (11) – мелким пунктиром. Сопоставление рис. 4 а, б с графом рис. 1 б позволяет видеть, каким образом реализуется выигрыш в производительности ЛП сортировки по сравнению с традиционной последовательной.

7. Обсуждение результатов

Граф (рис. 1 а) демонстрирует симметрию различных вариантов сортировки. Различные виды последовательностей из 3 элементов расположены в вершинах 6-угольника. Каждая вершина соединена с двумя другими, потому что связь соответствует обмену местами в паре соседствующих элементов, а у последовательности из трёх элементов имеется ровно 2 варианта пар обмена. Изменение порядка

элементов на обратный реализуется за 3 шага не только для (1, 2, 3) → (3, 2, 1), но и для любых пар последовательностей (вершин графа) с взаимно-инверсным расположением символов. Легко видеть, что на графе соответствующие парные вершины в 6-угольнике расположены диагонально.

Аналогично, граф (рис. 1 б) также демонстрирует симметрию. Каждая из вершин графа соединена с тремя другими, так как у последовательности из 4 элементов имеется три пары соседствующих элементов, что соответствует трём вариантам обмена. Изменение порядка элементов на обратный реализуется за 6 шагов не только для (1, 2, 3, 4) → (4, 3, 2, 1), но и для любых пар последовательностей (вершин графа) со взаимно-инверсным расположением символов. Это может быть непосредственно прослежено на графе. Изображение графа (рис. 1 б) укрупнено, имеет очертания ромба. Вершины графа, расположенные по левой верхней стороне ромба, в последовательности слева направо (т.е. (1, 2, 3, 4), (2, 1, 3, 4), (2, 3, 1, 4), (2, 3, 4, 1)) взаимно-инверсны вершинам графа, расположенным по правой нижней стороне ромба, в последовательности справа налево (т.е. (4, 3, 2, 1), (4, 3, 1, 2), (4, 1, 3, 2), (1, 4, 3, 2)). Аналогично, взаимно-инверсны вершины расположенные по правой верхней и левой нижней сторонам ромба.

Граф (рис. 1 б) – непланарный. При других его начертаниях (начиная, например, с изменения порядка расположения вершин второго слева столбца) другие его вершины будут «выведены» на стороны ромба. При этом в силу симметрии (каждая вершина соединена ровно с тремя другими) сохраняется взаимно-инверсное расположение вершин. Между взаимно-инверсными вершинами непосредственно и наглядно может быть прослежена на графе цепочки длиной ровно 6 шагов.

Описанная ситуация не является неожиданной, поскольку сводится к циклической замене символов или может рассматриваться как перекодировка записи выражений (1, 2, 3) → (3, 2, 1) и (1, 2, 3, 4) → (4, 3, 2, 1) и соответствующая замена символов в графах (рис. 1).

Таким образом, рассмотренное выше ЛП преобразование (1, 2, 3, 4) → (4, 3, 2, 1), проиллюстрированное графом (рис. 1 б) является достаточно общим, справедливым для любых пар взаимно-инверсных вершин.

В алгебраическом смысле рассматриваемые последовательности образуют конечные группы относительно операции перестановки пары элементов. В связи с этим представленные особенности, касающиеся симметрии в преобразованиях последовательностей, являются отражением теоретико-групповых свойств.

Работа ЛП процедуры сортировки (рис. 2) смоделирована в варианте с 8 3-битовыми сегментами. Заполнение сегментов выполнено наборами чисел (0, 1, 2, 3, ..., 7) со случайной перестановкой от генератора случайных чисел с равномерным законом распределения. Модель реализована на языке Python и демонстрирует работоспособность алгоритма.

8. Перспективы использования

Сортировка данных – классическая задача, исчерпывающе разработанная для ВС последовательного типа. С применением элементов параллельной обработки сортировка приобретает новые очертания. В случае ЛП целесообразна работа с ограниченными наборами данных. При этом скорость выполнения сортировки возрастает пропорционально числу конкатенированных сегментов, вследствие чего она может быть использована в приложениях с быстрым принятием оперативных решений. Применительно к ситуациям типа систем массового обслуживания, наряду с дисциплинами выборки заданий из очередей FIFO и FILO, получает право на существование очередь с заданиями, последовательность (очерёдность, важность, актуальность, срочность) выполнения которых изменяется в процессе нахождения заданий в очереди. Одним из приложений для подобных систем является Интернет: распределённая система хранения информации с децентрализованным управлением и мультиагентной обработкой. Другая важная прикладная область – интерфейс взаимодействия ВС с оператором, в частности, обучающие системы.

Выводы

Локально-параллельное представление информации обеспечивает повышенную эффективность работы программного обеспечения. Принципы локально-параллельной обработки применимы к задачам сортировки наборов данных. Процедуры сортировки рассмотрены на комбинаторном уровне для 3-, 4- и 8-элементных последовательностей. Разработано алгоритмическое обеспечение локально-параллельной сортировки. В ходе моделирования, проведенного на языке Python, выявлено, что локально-параллельный алгоритм максимально эффективен применительно к сортировке малых (в пределах разрядности процессора) выборок.

Список литературы: 1. *Мохамад, Али* Перспективы реализации локально-параллельных вычислений на многоядерных процессорах [Текст] / *Мохамад Али, О.Ф. Михаль* // Радиоэлектронные и компьютерные системы. – 2008. – № 6 (33). – С. 234–237. **2**. *Мохамад, Али*. Локально-параллельная сортировка данных с ограниченной разрядностью [Текст] / *Мохамад Али*. // Материалы международной научно-практической конференции студентов и аспирантов “Информационные технологии в экономике и образовании”. – М.: Российский университет кооперации, 2008. – С. 48–52. **3.** *Мохаммед, Али*

Локально-параллельная сортировка со встречной чередующейся упорядоченностью данных [Текст] / *Мохамед Али, О.Ф. Михаль* // Материалы 13-го международного молодёжного форума "Радиоэлектроника и молодёжь в XXI веке". Ч.2. – Харьков: ХНУРЭ, 2009. – С. 209. **4**. *Михаль, О.Ф.* Локально-параллельные алгоритмы определения степени включения и степени равенства нечетких множеств [Текст] / *О.Ф. Михаль* // Проблемы бионики. – Вып. 55. – 2001. – С. 80–90. **5**. *Михаль, О.Ф.* Принципы организации систем нечеткого регулирования на однородных локально-параллельных алгоритмах [Текст] / *О.Ф. Михаль, О.Г. Руденко* // Управляющие системы и машины. – 2001. – № 3. – С. 3-10. **6**. *Милнер, П.* Физиологическая психология [Текст] / *П. Милнер* М.: Мир, 1973. – 648 с. **7.** *Люггер, Дж.Ф.* Искусственный интеллект: стратегии и методы решения сложных проблем [Текст] / *Дж.Ф. Люгер* 4-е изд.: Пер. с англ. – М.: Изд. дом «Вильямс», 2005. – 864 с. **8**. *Кормен, Т.* Алгоритмы: построение и анализ [Текст] / *Т. Кормен, Ч. Лейзерсон, Р. Ривест* М.: МЦНМО, 2001. – 960 с.

Поступила в редколлегию 28.09.2011

УДК 519.87

Локально-паралельне сортування обмежених малих наборів даних / Мохамад Алі, О.П. Міхаль // Біоніка інтелекту: наук.-техн. журнал. – 2011. – № 3 (77). – С. 119-125.

Розглянуто локально-паралельне подання інформації стосовно задач сортування наборів даних. Процедури сортування проаналізовано на комбінаторному рівні для 3-, 4- и 8-елементних послідовностей. Подано алгоритмічне забеспечення локально-паралельного сортування. В ході моделюрования на мові Python виявлено, що локально-паралельний алгоритм максимально эфективний стосовно до сортування виборок в межах разрядности процесора.

Іл. 4. Бібліогр.: 8 найм.

UDK 519.87

Local-parallel sorting of limited small sets of data / Mohamad Ali, O.Ph. Mikhal // Bionics of Intelligense: Sci. Mag. – 2011. – № 3 (77). –P. 119-125.

Local-parallel presentation of information is considered referring to problem of the sorted data sets. The Procedures of the sorting is analysed on combinatorial level for 3-, 4- and 8-element sequences. Algorithmic provision of local-parallel sorting is presented. In the course of modeling in language Python is revealled that local-parallel algorithm is greatly efficient with reference to sorting the samples within processor register size.

Fig. 4. Ref.: 8 items.

УДК 004.93

Е. А. Гофман[1], А. А. Олейник[2], С. А. Субботин[3]

[1] Запорожский национальный технический университет, г. Запорожье, Украина, gofman_jenek@rambler.ru;

[2] Запорожский национальный технический университет, г. Запорожье, Украина, olejnikaa@gmail.com;

[3] Запорожский национальный технический университет, г. Запорожье, Украина, subbotin@zntu.edu.ua

ИНДУКЦИЯ ЛИНГВИСТИЧЕСКИХ ПРАВИЛ С ИСПОЛЬЗОВАНИЕМ ДЕРЕВЬЕВ РЕШЕНИЙ

Рассматривается задача индукции лингвистических правил. Разработан метод идентификации деревьев решений для индукции лингвистических правил. Создано программное обеспечение на основе предложенного метода.

ДЕРЕВО РЕШЕНИЙ, ИДЕНТИФИКАЦИЯ, ИНДУКЦИЯ ПРАВИЛ, ЛИНГВИСТИЧЕСКОЕ ПРАВИЛО, ОБУЧАЮЩАЯ ВЫБОРКА

Введение

В настоящее время экспертные системы, основанные на лингвистических правилах [1], успешно применяются в различных прикладных областях, таких как медицинская и техническая диагностика, финансовый менеджмент, распознавание образов, геологическая разведка, управление компьютерными сетями, управление технологическими процессами, анализ веб-контента в Интернет и др. Широкое применение таких систем обусловлено в первую очередь тем, что они являются прозрачными и относительно дешёвыми в реализации.

Поскольку базы правил в экспертных системах часто характеризуются большим объёмом, актуальной является задача индукции правил, суть которой заключается в том, что на основе начального набора правил необходимо сформировать новую базу правил меньшего объёма, которая в достаточной мере представляла бы начальную базу правил и была бы менее избыточной.

Существуют различные методы индукции правил [2], однако эти методы при обработке правил анализируют их качество по отдельности, не рассматривая и не учитывая качество всей базы в целом, что приводит к получению неоптимальных баз нечётких правил. Поэтому актуальной является разработка новых методов индукции правил, которые учитывали бы качество всей базы знаний, а не только отдельных правил. Для решения данной задачи предлагается создавать деревья решений [3–5], которые после их построения переводились бы в лингвистические правила. Выбор деревьев решений обосновывается их возможностью выявлять ненаблюдаемые связи внутри рассматриваемых объектов, процессов и систем.

Целью данной работы является разработка метода индукции лингвистических правил с использованием математического аппарата деревьев решений.

Для достижения поставленной цели необходимо решить следующие задачи:

– обзор математического аппарата деревьев решений;

– приведение основных этапов идентификации деревьев решений в соответствие с решаемой задачей;

– создание правил преобразования деревьев решений в лингвистические правила;

– сравнение разработанного подхода с существующими методами индукции лингвистических правил.

1. Постановка задачи

Пусть задана база лингвистических правил $RB = \{R^1, R^2 ..., R^{RN}\}$, описывающая объекты обучающей выборки $O = \{O^1, O^2, ..., O^N\}$. Тогда на основе обучающей выборки объектов O требуется сформировать такую базу лингвистических правил $RB^* = \{R^1, R^2, ..., R^{RN^*}\}$, $RN^* << RN$, которая обеспечивала бы приемлемое качество прогнозирования экспертной системы, построенной на основе полученной базы лингвистических правил RB^*:

$$Q(RB^*) \geq Q_{threshold} \cdot$$

где $Q(RB^*)$ – точность прогнозирования или классификации по базе правил RB^*; $Q_{threshold}$ – минимально допустимая точность прогнозирования или классификации.

2. Деревья решений

Деревья решений представляют собой графовые интеллектуальные модели, во внутренних узлах которых расположены функции принятия решений на основе значений входных переменных, а во внешних узлах (терминальных узлах, листьях) содержатся значения выходной переменной, соответствующие условиям внутренних узлов [2, 6, 7].

Благодаря своей древовидной структуре такие модели позволяют наглядно представлять результаты вычислений. Поэтому они хорошо интерпретируются людьми-специалистами в прикладных областях, которые, как правило, не имеют специальной математической подготовки и незнакомы с методами и моделями искусственного интеллекта. Деревья решений позволяют эффективно решать задачи классификации и прогнозирования, обеспечивая при этом высокую точность.

Для применения деревьев решений на практике в целях классификации или прогнозирования значений выходных параметров исследуемых объектов по наборам значений входных характеристик необходимо с помощью данных обучающей выборки сформировать дерево решений таким образом, чтобы оно наилучшим образом описывало исследуемый объект.

Построение деревьев решений связано с извлечением правил из обучающих выборок. Каждый путь от корня дерева к одному из его листьев может быть преобразован к логическому высказыванию – правилу типа «если А, то В», где его антецедент получается путем использования всех условий, представленных во внутренних узлах от корня к выходному листу, а правая часть правила получается из соответствующего листа дерева.

Процесс построения дерева решений, как правило, содержит такие этапы: разрастание, ветвление, вычисление значения выходного параметра для листа, сокращение.

В результате этапа разрастания (увеличения, growing) некоторая вершина заменяется поддеревом, полученным путем ветвления этой вершины. На данном этапе происходит разделение выбранной вершины на несколько новых (в случае дихотомического дерева вершина разбивается на две новых). При этом перебираются все признаки и все возможные варианты ветвления по каждому из признаков. В результате остается вариант разбиения, при котором значение критерия качества разбиения является наилучшим. Если новые вершины являются перспективными для последующего разделения (критерии завершения разрастания не удовлетворены), то выполняется их ветвление. В случае невозможности дальнейшего разделения вершины она становится листом, и для нее выполняется процедура вычисления значения выходного параметра. Если ветвление вершины приводит к ухудшению качества дерева, то вершина также объявляется листом.

Процедура ветвления (разделения, splitting) дерева вызывается рекурсивно при выполнении этапа разрастания. Ветвление подразумевает создание для выбранной вершины заданного количества (для дихотомических деревьев – две) вершин-потомков.

Вычисление значения выходного параметра происходит путем передвижения по синтезированному дереву решений от корневого узла к листу в зависимости от значений входных параметров.

Этап сокращения (усечение, обрезка, pruning) используется для упрощения построенного дерева путем отсечения потомков у выбранной вершины, которая впоследствии становится листом с определенным значением. Усечение узла выполняется в случае, если оно не приведет к существенному ухудшению аппроксимационных и обобщающих характеристик дерева решений.

Таким образом, этап усечения дерева выполняется снизу вверх: движение начинается от листов дерева и происходит вверх до тех пор, пока аппроксимационные способности дерева решений остаются приемлемыми.

3. Индукция лингвистических правил на основе построения деревьев решений

Существующие методы построения деревьев решений [2–7] не учитывают особенностей задачи индукции лингвистических правил. В связи с этим разрабатывается новый метод построения деревьев решений для индукции правил. Подобно известным методам построения дерева решений, предлагаемый метод состоит из основных фаз: рост дерева и его сглаживание (сокращение), после чего выполняется преобразование дерева решений в лингвистические правила. Наиболее важными аспектами предлагаемого метода являются следующие: использование модифицированной энтропии как оценочной меры, и использование сглаживания для отсечения.

Таким образом, предлагаемый метод состоит из следующих этапов:

– рост дерева;

– сглаживание дерева;

– преобразование дерева решений в лингвистические правила.

На этапе роста дерева предлагается использовать жадный подход. В каждом узле, соответствующем подмножеству T обучающей выборки, выбирается признак f и значение v таким образом, что данные из T разделяются на два подмножества $T^1_{f,v}$ и $T^1_{f,v}$ исходя из условий $x_{i,f} \le v : T^1_{f,v} = \{x_i \in T : x_{i,f} \le v\}$ и $T^2_{f,v} = \{x_i \in T : x_{i,f} > v\}$. Такое разбиение разделяет множество объектов обучающей выборки на такие, для которых значение признака f меньше значения v, и на те, для которых значение признака f больше значения v.

С целью разбиения дерева решений для каждого возможного разбиения (f, v) рассчитывается оценочная функция (1):

$$Q(f,v) = p_{f,v} g(p^1_{f,v}) + (1 - p_{f,v}) g(p^2_{f,v}), \qquad (1)$$

где $p^1_{f,v} = P(y_i = 1 \mid x_i \in T^1_{f,v})$, $p^2_{f,v} = P(y_i = 1 \mid x_i \in T^2_{f,v})$ и $p_{f,v} = P(x_i \in T^1_{f,v} \mid x_i \in T)$; $g(p)$ – модифицированная энтропия для вероятности отнесения выходной переменной y к рассматриваемому классу при условии, что x больше или меньше значения v ($p^1_{f,v}$ и $p^2_{f,v}$ соответственно):

$$g(p) = -r(p)\ln(r(p)) - (1 - r(p))\ln(1 - r(p)), \qquad (2)$$

где $r(p)$ преобразует оценку вероятности:

$$r(p)=\begin{cases}\frac{1}{2}(1+\sqrt{2p-1}), \text{ если } p>0{,}5;\\ \frac{1}{2}(1-\sqrt{1-2p}), \text{ если } p>0{,}5.\end{cases}, \qquad (3)$$

Таким образом, чем ближе значение вероятности к 0,5, тем выше модифицированное значение, а чем дальше от 0,5, тем значение ниже.

Оценочная функция рассчитывается для всех возможных разбиений и выбирается разбиение с наименьшим значением оценочной функции. Разбиение начинается от корневого узла и продолжается до тех пор, пока не возникнет ситуация, когда невозможно произвести новое разбиение.

После выполнения первого этапа может возникнуть ситуация «переобучения» дерева, что может привести к не совсем корректной работе дерева на тестовых выборках. В связи с этим на втором этапе производится усечение большого дерева, чтобы дерево меньших размеров давало более стабильные оценки вероятности и было более интерпретабельным.

Далее описывается подход, который вместо урезания полного дерева будет производить переоценку вероятности каждого листового узла путем усреднения оценки вероятности по пути следования от корневого узла к листовому узлу. Для достижения данной цели была взята идея «утяжеления дерева» [8]. Если используется дерево для сжатия бинарного классового признака y_i, основанного на x_i, то в таком случае метод утяжеления дерева гарантирует, что коэффициент сжатия переоцененной вероятности не будет хуже, чем в успешно усечённом дереве. Так как предлагаемый метод применяется в большей степени к трансформированной оценке вероятности $r(p)$, чем непосредственно к p, то теоретически результат может быть следующим: путем использования переоцененной вероятности можно достичь ожидаемую классификацию обучающего множества с не худшим результатом, чем у правильно усеченного дерева.

Следует отметить, что данный подход также является сжатием, поскольку при помощи такого подхода оценка сжимается от дальних узлов дерева по направлению к оценкам узлов, которые находятся ближе к корню дерева.

Пусть узлы T_1 и T_2 являются элементами одного уровня с общим родительским узлом T. Пусть $p(T_1)$, $p(T_2)$ и $p(T)$ будут соответствующими оценками вероятности. Локальная переоцененная вероятность это $w_T p(T)+(1-w_T)p(T_1)$ для T_1 и $w_T p(T)+(1-w_T)p(T_2)$ для T_2. Локальная значимость w_T и сопутствующая функция $G(T)$ рассчитываются рекурсивно, основываясь на следующих формулах:

$$\frac{w_T}{1-w_t}=\frac{c\cdot\exp(-|T|\,g(p(T)))}{\exp(-|T_1|\,G(T_1)-|T_2|\,G(T_2))},$$

$$G(T)=\begin{cases}g(p(T))+\frac{1}{|T|}\log((1+\frac{1}{c})w_T), \text{ если } w_T>0{,}5,\\ \frac{|T_1|}{|T|}G(T_1)+\frac{|T_2|}{|T|}G(T_2)+\\ \quad+\frac{1}{|T|}\log((1+c)(1-iw_T)), \text{ в противном случае.}\end{cases}$$

Параметр c устанавливается априорно и показывает Байесову «оценку» разбиения. Для листового узла T устанавливается: $G(T)=g(p(T))$ и $w_T=1$.

После вычисления значимостей w_T для каждого узла рекурсивным методом (используется нисходящая рекурсия) необходимо рассчитать глобальную оценку вероятности для каждого узла дерева сверху вниз. Данный этап усредняет все оценки $r(p)$ от корневого узла T_0 к узлу T_h по пути T_0, ..., T_h, основываясь на значимости w_T. Следует отметить, что значимость w_T является лишь локально важной. Это означает то, что глобальная значимость узла T_h является $w_T{}^*=\prod_{i<k}(1-w_i)w_k$ на всём пути. По определению, $\sum_{i=1}^{h} w_i=1$ для любого направления, ведущему к листу. По следующим рекурсивным формулам вычисляется глобальная переоценка подчиненных узлов T_1 и T_2 в родительском узле T:

$$\hat{w}_{T_i}=\hat{w}_T(1-w_T), \qquad (4)$$

$$r^*(T_i)=r^*(T)+\hat{w}_{T_i}w_{T_i}r(p(T_i)), \qquad (5)$$

где $r(p(T))$ – преобразование по формуле (3) из оценки вероятности $p(T)$ в узле T. В корневом узле устанавливается: $\hat{w}=1$. После вычисления $r^*(T_h)$ для листового узла T_h в качестве оценки вероятности можно использовать следующее: $r^{-1}(r(T_h))$. Метка класса для T_i будет равна единице, если $r(T_i)>0{,}5$, в противном случае – нулю. Усечение дерева выполняется, начиная с основания по направлению вверх путем проверки идентичности узлов одного уровня. Если идентичность узлов выявлена, то они удаляются и используется значение родительского узла. Данная процедура будет продолжаться до тех пор, пока она не станет невозможной. Метод сглаживания последовательно улучшает работу дерева. Оценка временной сложности – $O(M)$, где M – количество узлов неусеченного дерева.

Неотъемлемой частью предложенного метода является этап преобразования дерева решений в эквивалентный набор легко поддающихся толкованию лингвистических правил. Важность такого преобразования объясняется двумя причинами.

1. Любому человеку легче понять и изменить набор правил, чем понять и изменить дерево решений. Потребность в таком изменении очевидна. Например, может возникнуть ситуация, когда имеется некоторое несоответствие между обучающей выборкой и реальной системой, что требует

ручной модификации автоматически созданной системы, и, таким образом, в системе, основанной на правилах, такую модификацию можно выполнить путём простого изменения соответствующих правил.

2. Тот факт, что набор правил логически эквивалентен соответствующему дереву решений для данной обучающей выборки, гарантирует, что любой математический анализ эффективности работы дерева решений относится не только к дереву решений, но и к соответствующему набору правил.

Самый простой способ преобразования дерева в эквивалентный набор правил заключается в том, чтобы создать набор правил из правил, каждое из которых соответствует отдельному листу дерева путём формирования логического объединения условий на пути от корня дерева к листу.

Предлагается подход, преобразующий дерево решений в набор логически эквивалентных правил. Целью предлагаемого подхода не является получение доказуемо минимального набора правил. Вместо этого с помощью предлагаемого подхода производится логическое усечение правил.

1. Проверка условий “>” и “<” во всех правилах с целью устранения избыточности в описании условий правил. Таким образом, выполняется, например, следующее преобразование: $(x < 3) \cap (x < 5)$ заменяется на $(x < 3)$.

2. Удаление условий, которые являются логически избыточными в контексте всего набора правил, т.е. удаление условий, которые идентифицируются исходя из структуры полученного дерева решений. Такое упрощение изменяет правило, связанное с конкретным листом дерева, при этом сохраняя полную адекватность всего набора правил.

Для каждого листа, отнесённого к классу *X,* создаётся правило о том, что объект относится к классу *X* путём конъюнкции условий, находящихся на пути следования от корня к *X*, но используя только те условия, которые соответствуют следующему правилу: для каждого узла *N* на пути от корня к листу с меткой *X*, условие, соответствующее родителю *N*, является частью конъюнкции только в том случае, если срабатывает условие соседства для узла *N*. Условие соседства для *N* считается успешным, если: узел *N* не является корнем, и соседний узел относительно *N* не является листом с меткой *X*.

Таким образом, результирующий набор правил является логически эквивалентным базовому дереву решений.

Предложенный метод индукции лингвистических правил с использованием деревьев решений был программно реализован на языке программирования C#.

Для экспериментов использовались тестовые данные, которые были взяты из общедоступных репозиториев [9]. Экспериментальные исследования проводились на основании выборки, которая содержала информацию об эхокардиограммах пациентов с сердечными приступами. Выборка содержала информацию о 132 пациентах, каждый из которых характеризовался 12 признаками. Кроме того, для каждого пациента указывалось, жив он или умер.

Предложенный метод индукции нечётких правил сравнивался с мультиагентным методом и каноническим методом эволюционного поиска. Исходя из проведенных экспериментов, были получены базы лингвистических правил, характеризующиеся следующим качеством классификации пациентов: 81,3%, 79,1% и 92,7% для мультиагентного, эволюционного и предложенного методов, соответственно.

Таким образом, можно отметить, что предложенный метод построения деревьев решений для индукции лингвистических правил обеспечивает более точные результаты прогнозирования по сравнению с другими известными методами индукции лингвистических правил.

Выводы

В работе решена актуальная задача индукции лингвистических правил.

Научная новизна работы заключается в том, что разработан новый метод построения деревьев решений, позволяющий выполнять индукцию лингвистических правил, что достигается за счёт введения дополнительных функций преобразования при росте дерева, путём сглаживания дерева решений для его усечения и за счёт введения критерия соседства при преобразовании дерева решений.

Разработанный метод идентификации деревьев решений для индукции лингвистических правил позволяет произвести преобразование и объединение правил, что обеспечивает возможность разработки экспертных систем на основании более логически прозрачных и простых баз лингвистических правил.

Практическая ценность полученных результатов заключается в том, что на основе предложенного метода разработано программное обеспечение, позволяющее производить индукцию баз правил для получения баз лингвистических правил, на основании которых можно создавать экспертные системы с меньшей ошибкой классификации.

Список литературы: 1. *Субботін, С. О.* Подання й обробка знань у системах штучного інтелекту та підтримки прийняття рішень : навч. посібник [Текст] / С. О. Субботін. – Запоріжжя: ЗНТУ, 2008. – 341 с. **2.** *Encyclopedia* of artificial intelligence / Eds.: J. R. Dopico, J. D. de la Calle, A. P. Sierra. – New York : Information Science Reference, 2009. – Vol.

1–3. – 1677 p. **3.** *Rokach L.* Data Mining with Decision Trees. Theory and Applications / L. Rokach, O. Maimon. – London : World Scientific Publishing Co, 2008. – 264 p. **4.** *Quinlan J. R.* Induction of decision trees / J. R. Quinlan // Machine Learning. – 1986. – № 1. – P. 81–106. **5.** *Quinlan J. R.* Decision trees and decision making / J. R. Quinlan // IEEE Transactions on Systems, Man and Cybernetics. – 1990. – № 2 (20). – P. 339–346. **6.** *Liu X.* A decision tree solution considering the decision maker's attitude / X. Liu, Q. Da // Fuzzy Sets and Systems. – 2005. – № 152 (3). – P. 437–454. **7.** *Classification* and regression trees / L. Breiman, J. H. Friedman, R. A. Olshen, C. J. Stone. – California : Wadsworth & Brooks, 1984. – 368 p. **8.** *Willems F. M. J.* The Context Tree Weighting Method: Basic Properties / F. M. J. Willems, Y. M. Shtarkov, T. J. Tjalkens // IEEE Transactions on Information Theory. – 1995. – № 3. – P. 653–664. **9.** UCI Machine Learning Repository [electronic resource] / Center for Machine Learning and Intelligent Systems. – Access mode : http://archive.ics.uci.edu/ml/datasets.html.

Поступила в редколлегию 20.06.2011

УДК 004.93

Індукція лінгвістичних правил з використанням дерев рішень / Є. О. Гофман, А. О. Олійник, С. О. Субботін // Біоніка інтелекту: наук.-техн. журнал. – 2011. – № 3 (77). – С. 126-130.

Розглядається завдання індукції лінгвістичних правил. Розроблено метод ідентифікації дерев рішень для індукції лінгвістичних правил. Створено програмне забезпечення на основі запропонованого методу.

Бібліогр.: 9 найм.

UDC 004.93

Linguistic Rules Induction with Decision Trees / Ye. A. Gofman, A. O. Oliinyk, S. A. Subbotin // Bionics of Intelligense: Sci. Mag. – 2011. – № 3 (77). –P. 126-130.

The problem of linguistic rules induction is considered in this paper. A method of decision trees identification for the linguistic rules induction is developed. The software based on the proposed method is created.

Ref.: 9 items.

УДК 004.93

С.А. Зайцев[1], С.А. Субботин[2]

[1] Запорожский национальный технический университет, г. Запорожье, zaitsev.serge@gmail.com

[2] Запорожский национальный технический университет, г. Запорожье, subbotin@zntu.edu.ua

МОДЕЛЬ ОТРИЦАТЕЛЬНОГО ОТБОРА С ИСПОЛЬЗОВАНИЕМ МАСКИРОВАННЫХ ДЕТЕКТОРОВ И МЕТОД ЕЁ ОБУЧЕНИЯ ДЛЯ РЕШЕНИЯ ЗАДАЧ ДИАГНОСТИРОВАНИЯ

Исследовалось применение модели отрицательного отбора в диагностировании. Предложена модель отрицательного отбора, использующая маскирование детекторов, и разработан метод ее обучения, что позволило повысить скорость работы модели и улучшить интерпретабельность получаемых результатов. Проведены эксперименты, подтверждающие эффективность предложенной модели.

ОТРИЦАТЕЛЬНЫЙ ОТБОР, БИТОВАЯ СТРОКА, ДЕТЕКТОР, ПРАВИЛО СОПОСТАВЛЕНИЯ, МАСКИРОВАНИЕ

Введение

При решении задач технического и медицинского диагностирования возникает необходимость определения дефектных изделий или опасных состояний объектов диагностирования, что предполагает наличие диагностической модели. Для обеспечения удобства применения диагностическую модель целесообразно представлять в виде набора продукционных правил вида “если-то”, которые могут быть получены посредством индуктивного обучения по прецедентам.

В простейшем случае каждое правило основывается на бинарном представлении антецедентов, где значением “1” кодируется наличие признака, а значением “0” – его отсутствие у данного экземпляра. Исходя из значений битовой строки, описывающей экземпляр, принимается решение об отнесении его к классу годных или дефектных.

Используемые для решения данной задачи подходы имеют ряд недостатков. В частности, нейронные сети требуют дополнительной процедуры вербализации для упрощения правил [1]. Деревья принятия решений часто сходятся на локальных оптимумах, при большом количестве признаков полученные правила значительно усложняются, а также деревья решений плохо поддаются переобучению [2].

В целях устранения перечисленных выше недостатков при решении данной задачи целесообразно использовать принцип отрицательного отбора в искусственных иммунных системах. Он обладает таким рядом свойств, как способность работать с бинарным представлением признаков, возможность обучаться на экземплярах только одного класса, распределенность вычислений [3].

Среди недостатков существующих реализаций модели отрицательного отбора [4] стоит отметить тот факт, что они требуют предварительной оценки и отбора информативных признаков, а также характеризуются высокой сложностью извлечения знаний из полученных результатов работы модели.

Цель работы заключается в разработке такой модели отрицательного отбора, которая бы позволяла проводить отбор информативных групп признаков и формировать на их основе продукционные правила для проведения диагностирования, а также разработать метод ее обучения.

1. Постановка задачи

Пусть мы имеем выборку S', состоящую из экземпляров, описанных битовыми строками фиксированной длины l. Будем считать, что набор всех возможных битовых строк длиной l формирует пространство признаков U. Множество U можно разделить на два комплементарных подмножества, описывающих “свои” (годные) и “чужие” (дефектные) экземпляры соответственно: $U = S \cup N$, $S \cap N = \varnothing$, где S – множество годных экземпляров, а N – множество дефектных. Обучающая выборка состоит только из годных экземпляров $S' \subset S$. В этом случае задача построения модели отрицательного отбора заключается в генерации такого набора правил, представленного множеством детекторов D, на основании которого каждый $x \in U$ можно однозначно отнести к классу годных или дефектных экземпляров.

2. Метод обучения модели отрицательного отбора с цензурированием

Рассмотрим модель [5], реализующую парадигму отрицательного отбора.

Как правило, для определения принадлежности экземпляра к множеству S или N модель отрицательного отбора в процессе обучения добавляет в набор D такие детекторы, которые не соответствуют “своим” экземплярам. Поскольку множества S и N комплементарны, то предполагается, что любая битовая строка x принадлежит множеству N, если ей соответствует хотя бы один детектор из набора D. Определение соответствия экземпляра детектору происходит на основании правила сопоставления $match(d,x)$, которое принимает значение “истина”, если детектор соответствует экземпляру, и “ложь” – в противном случае.

Таким образом, работа данной модели осуществляется в два этапа.

1. Генерация набора детекторов. Для этого случайно сгенерированные кандидаты в детекторы C, представленные в виде битовых строк, подлежат цензурированию, и те из них, которые не отсеиваются на данном этапе, попадают в набор детекторов D. В результате цензурирования отсеиваются те кандидаты в детекторы $c \in C$, для которых существует такой $x \in S$, чтобы $match(c,x)=1$.

2. Классификация. На этом этапе экземпляр x, поступающий на вход модели, сравнивается с детекторами из набора D, используя правило сопоставления. Если хотя бы один из детекторов при этом активизируется, т.е. $\exists d \in D : match(d,x)=1$, считается, что $x \in N$, в противном случае – $x \in S$.

На практике оказывается [5], что множество D относительно небольшой мощности способно обеспечить достаточно высокую точность классификации диагностируемых данных. Более того, при неизменном количестве детекторов объем годных экземпляров может увеличиваться без снижения точности диагностирования.

Метод генерации детекторов представляется чрезвычайно ресурсоемким, а потому очень важно своевременно осуществить останов во избежании генерации избыточного количества детекторов.

3. Бинарные метрики

В качестве правила сопоставления двух бинарных детекторов применяются различного рода метрики, позволяющие определить степень подобия двух битовых строк [5-8].

Правило r-последовательных битов (r-contiguous rule, RCB rule) [5-7] использовалось изначально в модели отрицательного отбора. Метод генерации детекторов для модели отрицательного отбора оперировал строками фиксированной длины. Так, для двух строк, представленных в виде последовательности из n битов $x=\{x_1,x_2,\ldots,x_n\}$ и $d=\{d_1,d_2,\ldots,d_m\}$ правило выглядит следующим образом:

$$match(d,x)=\begin{cases}1, \exists i, i \le n-r+1, \forall i \le j \le i+r-1 : x_j = d_j;\\ 0, \text{ в противном случае.}\end{cases}$$

Иными словами, две строки совпадают, если существует такое окно размером r, в пределах которого все биты обеих строк совпадают.

Данная метрика отличается своей простотой и используется в оригинальном методе генерации детекторов для модели отрицательного отбора, получившем свое дальнейшее развитие в линейном и “жадном” методах генерации детекторов [6]. Все эти методы ограничиваются использованием бинарной формы представления детекторов и RCB-правила.

Метрика R-chunks [7] является более общей по сравнению с RCB-метрикой, т.е. любой детектор, оперирующий RCB-правилом, может быть представлен в виде множества *r*-chunks детекторов. Для двух строк $x=\{x_1,x_2,\ldots,x_n\}$ и $d=\{d_1,d_2,\ldots,d_m\}$ длиной n и m соответственно, $n \le m$, правило *r*-chunks можно представить как:

$$match(d,x)=\begin{cases}1, \ \forall i \le j \le i+m-1 : x_j = d_j;\\ 0, \text{ в противном случае,}\end{cases}$$

где i определяет позицию начала отрезка строки (chunk).

Правило *r*-chunks позволяет повысить точность работы метода отрицательного отбора.

Метрика Хемминга применялась в [8] в качестве правила сопоставления:

$$match(d,x)=\begin{cases}1, \ \sum_{i=1}^{n}\left|\overline{x_i \oplus d_i}\right| \ge r;\\ 0, \text{ в противном случае,}\end{cases}$$

где n – длина битовой строки, $\oplus$ – операция “исключающее ИЛИ”, r – порог срабатывания правила, $0 \le r \le n$.

Метрика Левенштейна [8] может считаться обобщением метрики Хемминга. Её значение определяется минимальным количеством изменений, необходимых для преобразования одной бинарной строки в другую. При этом изменения могут быть следующего вида: вставка разряда, удаление разряда, замена одного разряда (бита) другим из алфавита. В некоторых вариациях метрика Левенштейна считает замену нескольких смежных разрядов одной операцией.

Использование метрики Левенштейна целесообразно, если экземпляры обладают различным количеством признаков.

Важно отметить, что в случае использования метрики rcb и r-chunks необходимо заранее учитывать информативность признаков и возможно произвести перестановку битов в строках таким образом, чтобы биты, соответствующие информативным признакам образовывали последовательность – только в таком случае, метрики rcb и r-chunks смогут учитывать их значения наиболее эффективно.

4. Метод обучения модели отрицательного отбора, использующий перестановку битов

Использование бинарных детекторов часто приводит к такому явлению, как “дыры”. “Дырой” (hole) называют такую бинарную строку, описывающую экземпляр чужого класса, для которой невозможно сгенерировать корректный детектор (т.е. любой сгенерированный детектор, который соответствует этой строке, будет также соответствовать какой-то строке, описывающей “свои” экземпляры). Иными словами, чужая строка $a \in N$ образовывает “дыру”, тогда и только тогда, когда $\forall x \in U, match(x,a)=1 : \exists s \in S, match(s,a)=1$. “Дыры” образовываются при использовании любой метри-

ки с фиксированной вероятностью соответствия [6]. В [9] предлагается решение этой проблемы. Каждый детектор должен иметь несколько способов представления в бинарном виде. Например, для строк $s_1 = 01101011$, $s_2 = 00010011$ зададим правило перестановки бит $L = 1-6-2-5-8-3-7-4$. Применив это правило к строкам, получим: $L(s_1) = 00111110$ $L(s_2) = 00001011$. Используя rcb-метрику с параметром $r = 3$, можно увидеть, что $match(s_1, s_2) = 1$, т.к. последние три бита этих строк совпадают. Однако $match(L(s_1), L(s_2)) = 0$.

Фактически использование маски перестановок позволяет изменять форму детектора в пространстве признаков, в то время как форма "своего" пространства остается неизменной. Каждое фиксированное представление формирует свои "дыры", а использование нескольких представлений снижет вероятность того, что одна и та же "чужая" строка приведет к формированию "дыр" одновременно во всех представлениях детектора.

Хотя такой подход и позволяет повысить точность работы модели отрицательного отбора за счет устранения "дыр", а также снизить количество детекторов в наборе, для каждого детектора производится несколько вычислений функции сопоставления, таким образом вычислительная сложность метода эквивалентна отрицательному отбору с цензурированием при большем количестве детекторов. Также усложняется процесс генерации набора детекторов, поскольку требуется случайным образом сгенерировать такую пару битовых строк (сам детектор и его представление после применения правила перестановки), чтобы ни одна из них не совпадала ни с каким "своим" экземпляром.

Существенным недостатком метода является то, что он применим только для метрики rcb.

5. Метод обучения с маскированием

С целью устранения недостатков метода, описанного выше, авторами предлагается использовать детекторы с маскированием. Для этого необходимо расширить алфавит, на основе которого формируются детекторы, дополнив его еще одним символом – символом маски Z: $\Omega = \{0,1,Z\}$.

Значение Z играет особую роль – оно соответствует любому значению $\{0,1\}$ бита в битовой строке. Учитывая этот факт, существующие метрики должны быть модифицированы.

Так, метрика Хемминга для порогового значения r и битовых строк s и d длиной l может быть представлена в виде:

$$match(d,s) = \begin{cases} 1, \sum_{i=1}^{l} \{1 \mid d_i \neq Z \wedge d_i \neq s_i\} \geq r; \\ 0, \text{ в противном случае.} \end{cases}$$

Иными словами, детектор d и битовая строка s совпадают, если у них совпадают r незамаскированных битов.

Метрика rcb для маскированных детекторов вычисляется следующим образом:

$$match(d,s) = \begin{cases} 1, \exists i, i = 1 \ldots l - r : \sum_{j=i}^{l} \{1 \mid d_i = Z \wedge d_i = s_i\} \geq r; \\ 0, \text{ в противном случае.} \end{cases}$$

В тех случаях, когда в результате обучения диагностической модели требуется получить набор правил, по которым будет проводиться классификация, рекомендуется применять модификацию метрики Хемминга:

$$match(d,s) = \begin{cases} 1, \sum_{i=1}^{l} \{1 \mid d_i = Z \vee d_i = s_i\} = l; \\ 0, \text{ в противном случае.} \end{cases}$$

Такая метрика предполагает, что детектор соответствует строке, если все незамаскированные биты детектора соответствуют битам строки.

Жизненный цикл детектора включает в себя следующие стадии:

– формирование полностью замаскированного детектора, т.е. такого детектора, у которого значение всех битов равно Z;

– замена замаскированных значений битов детектора;

– добавление детектора в набор.

Ниже представлен метод обучения модели отрицательного отбора, использующей маскирование детекторов.

1. Сформировать замаскированный детектор $d = \{Z\}^n$.

2. Если $\exists s \in S : match(d,s) = 1$, тогда перейти к этапу 3, в противном случае – к этапу 5.

3. Выбрать произвольным образом бит d_i, $i = 1,\ldots,l$, $d_i = Z$. Если такого бита не существует, тогда перейти к этапу 1, в противном случае – перейти к этапу 4.

4. Установить значение i-го бита детектора: $d_i = \neg s_i$. Перейти к этапу 2.

5. Добавить детектор d в набор детекторов: $D = D \cup \{d\}$. Если достигнуто достаточное для необходимого уровня точности количество детекторов, тогда перейти к этапу 6, в противном случае – перейти к этапу 1.

6. Останов.

В результате работы метода будет получен набор детекторов, у которых незамаскированные биты определяют правила, по которым можно проводить дальнейшую классификацию. Например, детектор $\{001Z1\}$ можно рассматривать следующим образом: если у экземпляра отсутствует 1-ый и 2-ой признаки, однако присутствуют 3-ий и 5-ый, то экземпляр считается чужим.

Так как метод не проверяет наличие подобных детекторов в наборе, то в процессе работы метода могут быть сгенерированы идентичные детекторы. По той же причине могут быть получены такие

пары детекторов, у которых все незамаскированные биты совпадают, например $\{010Z\}$ и $\{01ZZ\}$.

Чтобы ускорить работу обученной модели предлагается по окончании обучения удалить из набора детекторов дубликаты. Также для детекторов, отличающихся только замаскированными битами, целесообразно оставлять только те, у которых большее количество замаскированных битов. Таким образом можно сократить набор детекторов без потери точности классификации.

6. Модифицированный метод обучения с маскированием

Предложенный выше метод позволяет сократить количество детекторов в наборе, тем самым повысив скорость процесса классификации с использованием модели отрицательного отбора. Однако формирование такого набора детекторов остается ресурсоемкой задачей, поскольку каждый новый детектор после изменения некоторого бита требуется сопоставить с каждым “своим” экземпляром.

Этого можно избежать, если сохранять промежуточные детекторы, не вошедшие в состав модели, и использовать их в качестве кандидатов при генерации нового поколения детекторов.

Модифицированный метод обучения модели отрицательного отбора с маскированием состоит из следующих этапов.

1. Сформировать замаскированный детектор $c_0 = \{Z\}^n$.

2. Создать набор кандидатов в детекторы $C = \{c_0\}$.

3. Выбрать произвольным образом кандидат в детекторы из набора $c \in C$.

4. Если $\exists s \in S : match(c,s) = 1$, тогда перейти к этапу 5, в противном случае – перейти к этапу 8.

5. Выбрать произвольным образом бит c_i, $i = 1,\ldots,l$, $c_i = Z$. Если такого бита не существует, тогда перейти к этапу 1, в противном случае – перейти к этапу 6.

6. Установить значение i-го бита детектора: $c_i = \neg s_i$.

7. Добавить модифицированный кандидат в набор $C = C \cup \{c\}$. Перейти к этапу 3.

8. Добавить кандидат c в набор детекторов: $D = D \cup \{c\}$. Если достигнуто достаточное для необходимого уровня точности количество детекторов – перейти к этапу 9, в противном случае – перейти к этапу 1.

9. Останов.

В первоначальной реализации метода, кандидат в детекторы состоял исключительно из замаскированных битов, а следовательно, всегда совпадал с экземплярами из набора S. Данная модификация снижает вероятность совпадения, поскольку большая часть кандидатов в детекторы уже содержит незамаскированные биты. Модифицированный метод предполагает наличие набора кандидатов в детекторы, мощность которого увеличивается по мере обучения модели.

В качестве дальнейшего развития метода можно выбирать кандидата в детекторы из набора не случайным образом, а основываясь на значении некоторой фитнесс-функции, которая своим значением определяет насколько данный кандидат пригоден для формирования детекторов. Пригодность может определяться количеством замаскированных битов, а также количеством детекторов, сформированных на основе данного кандидата.

7. Эксперименты и результаты

Предложенный метод синтеза диагностической модели тестировался как на синтетических выборках, так и на практических задачах диагностирования [10] с использованием программной реализации метода на языке Python.

Синтетические тесты включали в себя наборы детекторов длинной $l = 4,\ldots,12$. Обучение проводилось на экземплярах “своего” класса, затем проводилась классификация и вычислялось значение ошибки первого рода, определяемое числом неверно распознанных чужих экземпляров выборки.

На рис. 1 представлены графики, отображающие зависимости количества детекторов в наборе D от длины битовых строк l для метода с цензурированием и для предложенной модификации метода с маскированием соответственно.

Рис. 2 отображает количество проверок соответствия детекторов и экземпляров для каждого из методов.

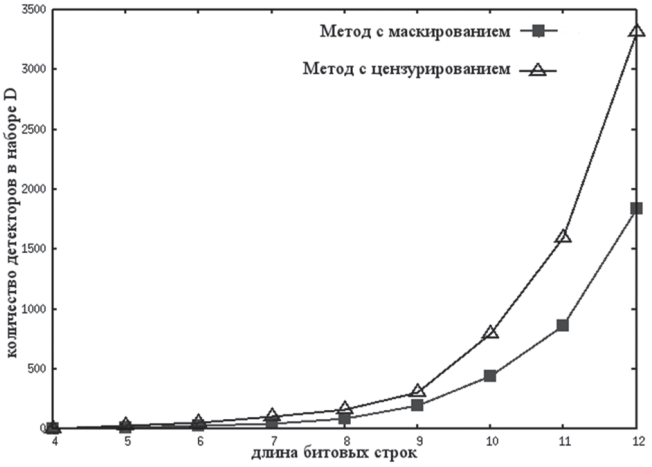


Рис. 1. График зависимости размера набора детекторов от размерности задачи

Как видно из рис. 1 и рис. 2, предложенный метод требует значительно меньше вычислений при построении модели отрицательного отбора, хотя при этом точность работы метода выше и составляет 95–100%.

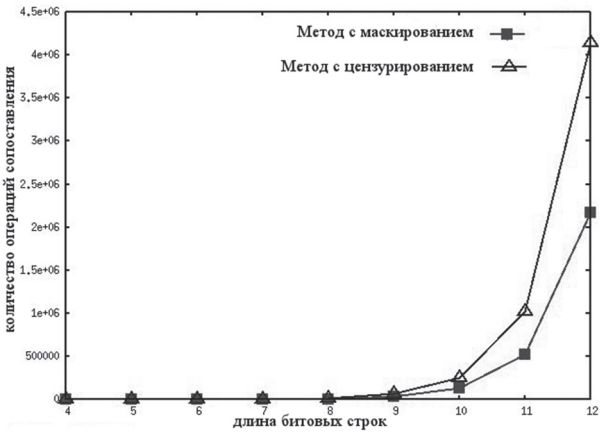


Рис. 2. График зависимости количества проверок правила сопоставления от размерности задачи

Выводы

С целью решения актуальной задачи автоматизации процесса диагностирования объектов, характеризуемых набором бинарных признаков, разработано математическое обеспечение, позволяющее проводить определение класса состояний объектов диагностирования на основе иммунокомпьютинга.

Научная новизна работы заключается в том, что впервые предложена модель отрицательного отбора, использующая маскированные детекторы, а также метод ее обучения, основная идея которого состоит в том, чтобы замаскировать максимальное количество битов, отвечающих за признаки с низкой информативностью. Это позволяет одновременно с построением диагностической модели осуществлять отбор групп информативных признаков, за счет чего повышается скорость работы метода обучения, а также упрощаются генерируемые продукционные правила посредством сокращения числа условий в антецедентах, что, в свою очередь, упрощает диагностическую модель, повышает скорость ее работы и интерпретабельность.

Практическая ценность работы заключается в том, что разработано программное обеспечение для проведения диагностирования объектов с помощью предложенной модели отрицательного отбора.

Тестирование предложенной модели отрицательного отбора показало высокую точность классификации, что позволяет рекомендовать ее использование для решения практических задач.

Дальнейшие исследования могут быть сосредоточены на развитии метода обучения предложенной модели, в частности, на разработке критериев отбора кандидатов в детекторы.

Список литературы: 1. *Миркес, Е.М.* Логически прозрачные нейронные сети и производство явных знаний из данных [Текст] / Е.М. Миркес, А.Н. Горбань, В.Л. Дунин-Барковский, А.Н. Кирдин и др. // Нейроинформатика. Новосибирск: Наука. Сибирское предприятие РАН, 1998. – С. 296. **2.** *J. Ross Quinlan* C4.5: Programs for Machine learning // Massachusetts, USA: Morgan Kaufmann Publishers, 1993. P. 324. **3.** *Gonzalez F., Dasgupta D., Gomez J.* The effect of binary matching rules in negative selection // Proceedings of the Genetic and Evolutionary Computation Conference GECCO–2003 (9-11 July, 2003). Berlin-Heidelberg: Springer-Verlag, 2003. P. 195–206. **4.** *Ji Z., Dasgupta D.* Revisiting negative selection algorithms // Evolutionary Computation. 2007. №15. P. 223–251. **5.** *Forrest S., Perelson A.S., Cherukuri R., Allen L.* Self-Nonself Discrimination in a Computer // Proceedings of the 1994th IEEE Symposium on Research in Security and Privacy (1994). CA: IEEE Computer Society Press, 1994. P. 202–212. **6.** *D'haeseleer P., Forrest S., Helman P.* An immunological approach to change detection: algorithms, analysis, and implications // Proceedings of the 1996th IEEE Symposium on Computer Security and Privacy (6-8 May, 1996). CA: IEEE Computer Society Press, 1996. P. 110–119. **7.** *Balthrop J., Esponda F., Forrest S., Glickman M.* Coverage and generalization in an artificial immune system // Proceedings of the Genetic and Evolutionary Computation Conference GECCO–2002 (9-13 July, 2002). San Fransisco, USA: Morgan Kaufmann, 2002. P. 3–10. **8.** *Farmer J., Packard N., Perelson A.* The immune system, adaptation, and machine learning // Physica D: Nonlinear Phenomena. 1986. №2. P. 187–204. **9.** *Hofmeyr S., Forrest S.* Architecture for an Artificial Immune System // Evolutionary Computation. 2000. №8(4). P. 443–473. **10.** *Герасимчук, Т.С.* Использование искусственных иммунных систем для прогнозирования риска развития реккурентных респираторных инфекций у детей раннего возраста [Текст] / Т.С. Герасимчук, С.А. Зайцев, С.А. Субботин // Матеріали міжрегіональної науково-практичної конференції "Діагностика та лікування інфекційно опосередкованих соматичних захворювань у дітей" : конференція (10-11 лютого 2011 р.). Донецьк: Норд-пресс, 2011. – С. 27–29.

Поступила в редколлегию 27.06.2011

УДК 004.93

Модель негативного відбору з використанням маскованих детекторів та метод її навчання для вирішення задач діагностування / С. О. Зайцев, С. О. Субботін // Біоніка інтелекту: наук.-техн. журнал. – 2011. – № 3 (77). – С. 131-135.

Проводилось дослідження моделі негативного відбору в задачі діагностування. Запропоновано модель негативного відбору, що використовує маскування детекторів, та розроблено метод її навчання, що дозволило підвищити швидкість роботи моделі та покращити інтерпретабельність результатів. Проведено експерименти, що підтверджують ефективність запропонованої моделі.

Іл.: 2. Бібліогр.: 10 найм.

UDC 004.93

Negative selection model with masked detectods and its training method in diagnostics / S. A. Zaitsev, S. A. Subbotin // Bionics of Intelligense: Sci. Mag. – 2011. – № 3 (77). – P. 131-135.

Negative selection model application in diagnostics have been investigated. A new negative selection model based on detector masking has been proposed, which allows to increase model speed and improve results interpretation. The experiments have been carried to approve the efficiency of the suggested model.

Fig.: 2. Ref.: 10 items.

УДК 004.652.4, 004.652.6

С.С. Танянский [1], Ю.А. Мальков [2]

[1]ХНУРЭ, г. Харьков, Украина, tanyansky_ss@yahoo.com;
[2] ХНУРЭ, г. Харьков, Украина, malkov@smtp.ru

ОТОБРАЖЕНИЕ ЭЛЕМЕНТОВ РЕЛЯЦИОННОЙ МОДЕЛИ ДАННЫХ В ЭЛЕМЕНТЫ ДЕДУКТИВНОЙ МОДЕЛИ

Рассматривается соответствие понятий реляционной и дедуктивной моделей данных. Представлены отображения компонентов реляционной модели в соответствующие компоненты дедуктивной модели данных. Рассмотрены конструкции логического программирования, описывающие основные операции над данными реляционной модели.

БАЗЫ ДАННЫХ, РЕЛЯЦИОННАЯ МОДЕЛЬ ДАННЫХ, ДЕДУКТИВНАЯ МОДЕЛЬ ДАННЫХ, DATALOG, ЛОГИЧЕСКОЕ ПРОГРАММИРОВАНИЕ, ДЕДУКТИВНЫЕ БАЗЫ ДАННЫХ

Введение

На сегодняшний день реляционная модель является стандартом для баз данных. Широкое распространение она получила из-за своей простоты и декларативности. Ведущие производители СУБД, такие как Oracle, Microsoft, IBM, DB2 выпускают именно реляционные системы.

Под моделью данных подразумевают механизм хранения и обработки данных, а также язык, обеспечивающий взаимодействие между пользователем и базой данных.

Несмотря на свою широкую распространенность, реляционная модель не идеальна и имеет ряд ограничений, одним из которых является слабая выразительная мощность стандартного для реляционной системы языка SQL. Так как в нем отсутствует рекурсия, транзитивное замыкание не может быть определено без использования внешнего процедурного языка.

Другим существенным ограничением являются строгие требования реляционных систем к типам данных, что является недостатком при использовании неоднородных источников данных.

Также к ограничениям реляционной модели можно отнести отсутствие возможности работы с неопределенной информацией.

Следовательно, для решения описанных выше задач является целесообразным использование другой модели данных. Наиболее подходящей, с точки зрения описанных недостатков, является дедуктивная модель представления данных. В отличие от реляционных систем, дедуктивные базы данных (ДБД) используют доказательно-теоретическое представление данных вместо модельно-теоретичского.

Доказательно-теоретический подход к базам данных рассматривает базу данных как комбинацию данных (аксиом) и набор теорем, которые должны быть выведены из аксиом. Выполнение запроса к дедуктивной базе данных является процессом доказательства теоремы.

В отличие от модельно-теоретического, доказательно-теоретическое представление позволяет описывать конструкции базы данных (базовые данные, запросы, ограничения целостности) единообразно – при помощи общего языка. Это позволяет создавать более понятные и унифицированные интерфейсы.

На данный момент в литературе не существует четкого определения соответствий понятий реляционной и дедуктивной моделей данных. В связи с этим, целью данной работы является определение соответствия компонентов реляционной и дедуктивной модели данных, рассмотрение языковых конструкций логического программирования, используемых в ДБД, выражение стандартных операций реляционной алгебры в терминах логического программирования.

2. Основные компоненты реляционной и дедуктивной моделей данных

2.1. Реляционная модель

В реляционной модели база данных рассматривается как набор явных именованных переменных-отношений, каждое из которых содержит явный набор кортежей и явный набор ограничений целостности.

Согласно Дейту [1] реляционная модель состоит из трех частей:

$$M_R = < S, IC, O >, \quad (1)$$

где S – структура данных; IC – ограничения целостности и O – средства манипулирование данными.

Структурная часть включает данные и описание объектов, рассматриваемых реляционной моделью:

$$S = < D, R > \quad (2)$$

Постулируется [2], что единственной структурой данных, используемой в реляционной модели, являются нормализованные n-арные отношения. Каждое отношение r характеризуется схемой $R(A_1, A_2, \ldots, A_n)$, состоящей из множества имен атрибутов $A_1, A_2, \ldots, A_n$, каждому из которых ставится в соответствие множество D_i, называемое доменом атрибута A_i.

Целостная часть описывает ограничения, которые должны выполняться для любых отношений в любых реляционных базах данных. Ограничения целостности реляционной базы данных (*IC*) можно определить формулой:

$$IC =< B, F >, \quad (3)$$

где B – бизнес-правила – ограничения, которые зависят от семантики элементов домена, а F– функциональные зависимости.

Бизнес-правила ограничивают значения атрибута отношения, например, рост человека не может быть равен 100 метрам, а функциональные зависимости определяют зависимость между соответствующими атрибутами.

Манипуляционная часть реляционной модели описывает способы определения и манипулирования данными и выражается формулой:

$$O =< DDL, DML >, \quad (4)$$

где DDL – язык определения данных (ЯОД), а DML – язык манипулирования данными (ЯМД).

2.2. Дедуктивная модель данных

Дедуктивная модель данных интерпретирует базу данных как множество предложений логики первого порядка, а под выполнением запроса или удовлетворением ограничения здесь рассматривается доказательство того, что некоторая логическая формула является логическим следствием из базовых данных.

Дедуктивная модель данных представлена формулой:

$$M_D =< EDB, P >, \quad (5)$$

где EDB – множество данных (экстенсионал), именуемых как факты, и P – множество правил вывода, являющихся логической программой.

Под *логической программой P* подразумевают конечное множество правил вывода вида:

$$p \leftarrow p_1, p_2, \ldots, p_m, not\ p_{m+1}, \ldots, not\ p_n, \quad (6)$$

где $m,n \geq 0$, а p и p_i – литералы: p называют головой правила, множество литералов p_i образует тело правила, а символом $\leftarrow$ обозначена операция импликации. Приведенная запись интерпретируется следующим образом: «Если истинно $p_1, p_2, \ldots, p_m, not\ p_{m+1}, \ldots, not\ p_n$, то истинно p».

Как видно из формулы (5), дедуктивная модель, в отличие от реляционной, обладает дополнительными возможностями, реализуемыми с помощью машины вывода логического программирования. Этот механизм, используя алгоритмы логического вывода, такие как обратный вывод и резолюция [3], способен продуцировать новые факты на основе применения правил вывода к фактам, заданным в *EDB*.

Машину вывода ДБД можно представить как функцию f:

$$EDB' = f(EDB), \quad (7)$$

где EDB – множество фактов экстенсионала; f – функция продуцирования фактов и EDB' – множество фактов, полученных вследствие логического вывода.

Логическую программу P в контексте баз данных можно представить формулой:

$$P =< IDB, IC >, \quad (8)$$

где IDB – интенсионал – часть дедуктивной БД, состоящая из множества правил вывода, а IC множество ограничений целостности.

В ДБД представление и манипулирование данными осуществляется при помощи специальных конструкций, называемых дизъюнктами (9). Дизъюнкт является конечным списком литералов вида $p(a_1, a_2, \ldots, a_n)$ или отрицаний литералов $-p(a_1, a_2, \ldots, a_n)$. Приведенные литералы являются предикатными функциями, возвращающими множество значений {0,1} (или «ложь» и «истина») и определены на множестве фактов *EDB*:

$$p(a_1, a_2, \ldots, a_n) \leftarrow p_1(a_1, a_2, \ldots, a_k), \ldots, p_m(a_1, a_2, \ldots, a_n) \quad (9)$$

Любое выражение, включая данные (факты), правила вывода, ограничения целостности и запросы, определяется при помощи дизъюнктов. В соответствии с (9) рассмотрим несколько типов дизъюнктов.

– Факты. Если в дизъюнкте будет отсутствовать тело, а все значения аргументов предиката p будут константами, то он будет представлять основную аксиому, т.е. утверждение, которое однозначно является истинным.

– Правила вывода имеют вид (9) и могут рассматриваться как дедуктивные аксиомы, которые дают определение предиката в голове правила в терминах, представленных в теле правила.

– Ограничения целостности выражаются дизъюнктами, в которых отсутствует голова.

– Запросы или цели – это дизъюнкты, тело которых содержит предикатный символ, определяющий множество фактов, над которым будет осуществляться запрос, а голова состоит из знака “?”.

Интерпретацией логической программы P называют комбинацию пространства рассуждений, отображения индивидуальных констант из этой программы на объекты в этом пространстве и набора заданных значений для предикатов и функций, содержащихся в P.

Если дизъюнкт p удовлетворяется в интерпретации τ – то говорят, что τ является моделью для p. Аналогично, если множество дизъюнктов P удовлетворяется в интерпретации τ, говорят, что τ является моделью для P.

Поскольку логическая программа может допускать несколько интерпретаций, то, следовательно, она может иметь более одной модели. Таким обра-

зом, с теоретико-доказательственной точки зрения база данных в общем случае может иметь несколько различных моделей, в отличие от модельно-теоретической, где модель всегда одна.

После введения базовых понятий логического программирования, применяемого в дедуктивных базах данных, рассмотрим соответствие основных компонентов реляционной и дедуктивной модели.

3. Отображение компонентов реляционной модели данных в компоненты дедуктивной

Рассмотрение приведенных моделей данных показало, что реляционная модель данных (1) состоит из компонентов, описывающих структуру данных (2), целостную часть (3), представленную ограничениями целостности, и языков определения и манипулирования данными (4).

В то же время дедуктивная модель данных представляется в виде множества фактов *EDB*, и логической программы (5), состоящей из множеств правил вывода и ограничений целостности (8).

Проведенный выше анализ моделей данных позволяет сделать вывод о возможности отображения составных частей реляционной модели данных в компоненты дедуктивной (рис. 1).

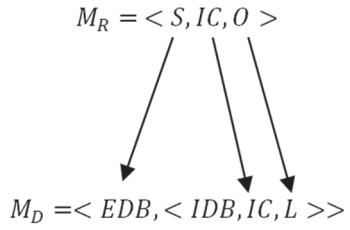


Рис. 1. Отображение компонентов реляционной модели данных в компоненты дедуктивной

Для структурной части реляционной модели можно построить отображение в множество фактов экстенсионала *EDB* дедуктивной модели:

$$\rho : S^{M_R} \to EDB^{M_D}. \qquad (10)$$

Множество ограничений целостности реляционной модели можно представить множеством правил вывода логической программы, являющейся частью дедуктивной модели:

$$\beta : IC^{M_R} \to IC^{M_D}. \qquad (11)$$

Манипуляционную часть реляционной модели данных можно выразить с помощью эквивалентных логических формул и конструкций логического программирования:

$$\theta : O^{M_R} \to L^{M_D}. \qquad (12)$$

Рассмотрим более подробно отображение каждого компонента реляционной модели.

3.1. Отображение структурной части

Структурная часть реляционной модели определяет отношение как средство хранения данных. В то же время предикат можно представить как функцию, описывающую реляционное отношение: если значения аргументов предиката являются кортежем некоторого отношения, то предикат будет возвращать истинное значение.

Определение: Пусть *r* – реляционное отношение со схемой $R(A_1, A_2, \ldots, A_n)$, которое содержит конечное множество кортежей $\{t_1, t_2, \ldots, t_k\}$. Каждый кортеж $t_i(A_1, A_2, \ldots, A_n)$ представляется в виде литерала $p_i(a_1, a_2, \ldots, a_n)$, где каждый атрибут A_j кортежа t_i отображается в аргумент a_j предиката p_i:

$$\phi : A_j^{t_i} \to a_j^{p_i}. \qquad (13)$$

Поскольку предикат является функцией, вышесказанное можно переформулировать следующим образом: предикат p_i будет принимать значения “истина” лишь в том случае, если его аргументы будут отображением значений некоторого кортежа t_i из отношения *r*.

Рассмотрим пример: пусть дано отношение *r* (рис.2), состоящее из двух кортежей $\{a_1, b_1, c_1\}$ и $\{a_1, b_2, c_2\}$. Представим данное отношение в терминах дедуктивной модели с помощью предиката *p*. Тогда предикат *p* будет возвращать значение “истина” лишь при подстановке в качестве его аргументов $\{a_1, b_1, c_1\}$ и $\{a_1, b_2, c_2\}$.

A	B	C
a_1	b_1	c_1
a_1	b_2	c_2

$p(a_1, b_1, c_1).$
$p(a_1, b_2, c_2).$

Рис. 2. Отображение реляционного отношения в множество фактов

Кортеж отношения *r*, представленный в виде *n*-арного предикатного символа $p(a_1, a_2, \ldots, a_n)$, где *n* арность исходного отношения *r*, а a_i – константа, называют *фактом*. Множество всех фактов дедуктивной базы данных принято называть *экстенсионалом*.

В реляционной модели данных понятие отношения непосредственно связано с понятием домена [2]. В связи с этим рассмотрим соответствующие домену понятия дедуктивной модели и построим отображение.

Определение: пусть $D = \{D_1, D_2, \ldots, D_n\}$ – множество доменов, где $D_i = \{d_1^i, d_2^i, \ldots, d_k^i\}$, где d_j^i - значение домена $(1 < j < k)$. Определим предикат $p(a_1, a_2, \ldots, a_n)$ и множество значений $a_j^i = \{a_1^i, a_2^i, \ldots, a_k^i\}$ его аргумента a_i. Тогда, можно построить отображение γ множества элементов домена D_i в множество значений аргумента a_i:

$$\gamma : d_j^i \to a_j^i. \qquad (14)$$

Представим систему отображений структурной части реляционной модели в экстенсионал дедуктивной при помощи диаграммы (рис. 3).

Пусть S^{M_R} – множество отношений, определенных в реляционной модели M_R, а EDB^{M_D} –

множество фактов экстенсионала M_D. Пространство допустимых состояний, выразимых в M_R и M_D, обозначим, соответственно B_R и B_D. Тогда ρ – отображение вида (10), $\omega: B_R \to B_D$ – отображение пространства состояний M_R в пространство состояний M_D, а μ^{Def} – семантическая функция модели языка определения данных.

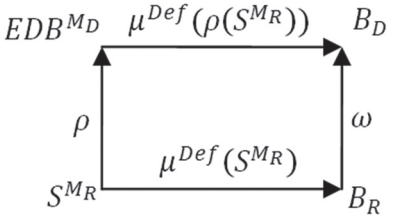


Рис. 3. Диаграмма отображения компонент M_R в M_D

3.2. Отображение манипуляционной части

Как было сказано выше, в основе дедуктивной модели лежит доказательно-теоретический подход к базам данных, расширяющих классическое понимание базы данных за счет применения логического программирования.

Наиболее распространенным языком, применяемым в ДБД, является Datalog.

Семантика Datalog определена на алфавитах $Var, Const\ 8\ Pred$, обозначающих переменные, константы и предикаты.

Переменные обозначают строками буквенно-цифровых символов конечной длины, начинающихся с прописной буквы. Следует отметить, что существует специальная переменная «_», называемая анонимной. Ею обозначают любую неименованную переменную. Анонимные переменные используют для отсечения, некоторых аргументов предиката, что аналогично операции проекции в реляционной алгебре.

Константой является текстовая строка или число. Например: “строковая константа” или “12”.

Предикаты обозначают строками алфавитно-цифровых символов конечной длины, которые начинаются со строчной буквы. Как говорилось выше, предикаты являются булевыми функциями над множествами данных.

К функциям Datalog, помимо формулирования дедуктивных правил вывода, относят описание запросов. Для выражения запросов, называемых в логическом программировании целями, на языке Datalog используют дизъюнкты вида (15). Голова цели состоит из символа “?”, а тело – из предиката, над которым выполняют запрос и условий отбора:

$$?\leftarrow p(A_1, A_2, \ldots, A_k, \ldots, A_n), A_k\, \Omega C, \qquad (15)$$

где, p – предикат, определенный на множестве фактов EDB^{M_D}; A_i – переменные, связанные с значениями аргументов предиката; $A_k\, \Omega C$ условие выборки; Ω – множество логических функций сравнения $(>,<,=,\neq)$; C – константа. Следовательно, результатом запроса будут являться все факты p, у которых аргумент A_k будет отвечать заданному условию отбора.

Цели, как и любые дизъюнкты, не содержат свободных переменных. Т.е. все появляющиеся в цели переменные связаны квантором всеобщности. Из этого следует, что переменная, появляющаяся в некоторой цели G, имеет смысл только в этой цели и ее значение не распространяется на другие составляющие логической программы.

Синтаксис логического программирования основан на использовании операций булевой алгебры, к которым относятся отрицание, импликация, конъюнкция и дизъюнкция. В Datalog для выражения булевых операций используются символы и соглашения, определенные в табл. 1.

Таблица 1

Выражение булевых операций в Datalog

Дизъюнкция	«;»
Конъюнкции	«,»
Отрицание	«not»
Импликация	« ← »

Синтаксис языка Datalog является более выразительным чем, языки реляционных систем, и все операции реляционной алгебры и реляционного исчисления можно выразить с помощью эквивалентных логические операций и конструкций логического программирования. Рассмотрим их подробно (табл. 2).

Таблица 2

Соответствие формул Datalog операциям реляционной алгебры

Операция реляционной алгебры	Обозначение	Равнозначное выражение на Datalog
Объединение	$R \cup S$	$q(X_1, X_2, \ldots, X_n) \leftarrow r(X_1, X_2, \ldots, X_n).$ $q(X_1, X_2, \ldots, X_n) \leftarrow s(X_1, X_2, \ldots, X_n).$
Пересечение	$R \cap S$	$q(X_1, X_2, \ldots, X_n) \leftarrow r(X_1, X_2, \ldots, X_n), s(X_1, X_2, \ldots, X_n).$
Разность	$R - S$	$q(X_1, X_2, \ldots, X_n) \leftarrow r(X_1, X_2, \ldots, X_n), not\ s(X_1, X_2, \ldots, X_n).$
Декартово произведение	$R \times S$	$q(X_1, X_2, \ldots, X_{k+s}) \leftarrow r(X_1, X_2, \ldots, X_k), s(X_{k+1}, X_{k+2}, \ldots, X_{k+s}).$
Селекция	$\sigma_C(R)$	$q(X_1, X_2, \ldots, X_k \ldots, X_n) \leftarrow r(X_1, X_2 \ldots, X_k, \ldots, X_n), X_k > C.$
Проекция	$\pi_{X,Y,\ldots,Z}(R)$	$q(X_1, X_2, \ldots, X_k) \leftarrow r(X_1, X_2, \ldots, X_k, \_, \ldots, \_).$

Операции соединения и деления являются зависимыми операциями, и следовательно могут быть выражены с помощь рассмотренных операций разности и декартова произведения.

Представленные выражения на Datalog позволяют выражать все операции манипулирования данными, используемые в реляционной модели. В связи с этим можно построить диаграмму отображения языка манипулирования данными реляционной модели в дизъюнкты Datalog (рис. 4).

Пусть O^{M_R} – множество операторов языка манипулирования данными модели M_R, L^{M_D} – множество рассмотренных выражений на Datalog, представляющих ЯМД модели M_D, эквивалентных операциям реляционной алгебры. Тогда θ – отображение вида (12), $\omega: B_D \to B_R$ – отображение пространства состояний M_R в пространство состояний M_D, а μ^{Man} - семантическая функция модели языка манипулирования данными.

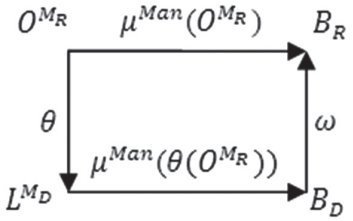


Рис. 4. Диаграмма отображения операторов ЯМД

3.3. Отображение целостной части

Под ограничениями целостности подразумевают соответствие имеющейся в базе данных информации её внутренней логике, структуре и всем явно заданным правилам. Каждое правило, налагающее некоторое ограничение на возможное состояние базы данных, называется ограничением целостности.

Ограничения целостности, используемые в реляционной модели, можно классифицировать следующим образом [4]:

1. Первичные ключи.
2. Ограничения ссылочной целостности.
3. Ограничения области значений.

В общем случае ограничения целостности на Datalog выражают при помощи правил вида (16). Голова такого правила является пустой, а тело состоит из предиката, который нарушает целостность базы при истинности условия:

$$\leftarrow p(A_1, A_2, \ldots, A_k, \ldots, A_n), A_k\, \Omega C, \qquad (16)$$

где p – предикат, определенный на множестве фактов EDB^{M_D}; A_i – переменные, связанные с значениями аргументов предиката; $A_k\, \Omega C$ условие; C – константа. Из приведенного выражения следует, что целостность базы данных будет нарушена, если найдется хоть один факт p, у которого аргумент A_k будет отвечать заданному условию.

Первичным ключом называют атрибут или множество атрибутов, уникальным образом идентифицирующих объект в его классе. Никакие два объекта класса не могут совпадать по своим значениям для каждого множества атрибутов, формирующих ключ. Применительно к ДБД, приведенное определение означает, что во множестве EDB^{M_D} не должно существовать двух фактов с одинаковыми значениями аргументов, определяющих уникальность факта.

Для представления первичных ключей в дедуктивной модели данных вводят правила вида:

$$unique\, key(p, \gamma), \qquad (17)$$

где p – имя ограничиваемого ключом предиката и γ подмножество аргументов p, определяющих состав ключа. Подмножество аргументов γ задают в виде перечисления позиций аргументов, разделенных запятой. Например: *unique key*(p, [1,3]) .

Ограничение ссылочной целостности в реляционной модели заключается в том, что каждому кортежу отношения, содержащему внешний ключ, должен соответствовать кортеж в другом отношении, содержащем первичный ключ. Отношение, содержащее внешний ключ, называют дочерним, а первичный – родительским.

В дедуктивной модели ограничение ссылочной целостности в общем виде представляют правилом вида:

$$\leftarrow p_1(A_1, \ldots, A_k, \_\ldots, \_), not\ p_2(A_1, \ldots, A_k, \_, \ldots, \_), \qquad (18)$$

где p_1 – предикат, на который накладывается условие ссылочной целостности; p_2 – родительский предикат для p_1; $A_1, A_2, \ldots, A_k$ – атрибуты, входящие в состав внешнего ключа.

Ограничения области значений требуют, чтобы значение атрибута реляционного отношения выбиралось из определенного множества значений или находилось в определенных границах. Ограничения такого вида можно представить правилами вида (19, 20):

$$\leftarrow p(A_1, \ldots, A_k, \ldots, A_n), not\ A_k = C_1, \ldots, not\ A_k = C_m, \qquad (19)$$

где p – предикат; A_k – аргумент предиката, значение которого, не входящее в множество значений $\{C_1, C_2, \ldots, C_m\}$, нарушает целостность БД,

$$\leftarrow p(A_1, A_2, \ldots, A_k, \ldots, A_n), A_k < C_1, C_2 < A_k, \qquad (20)$$

где p – предикат; A_k – аргумент предиката, значение которого, выходящее за пределы диапазона $\{C_1, C_2\}$, нарушает целостность БД.

Кроме представленных выше ограничений целостности, накладываемых на факты *EDB*, дедуктивная система позволяет ограничивать вывод фактов, полученных в результате логического вывода (EDB'). Следовательно, ограничения целостности в дедуктивной модели можно представить формулой:

$$IC =< IC^{EDB}, IC^{EDB'} >, \qquad (21)$$

где IC^{EDB} – ограничения целостности, накладываемые на множество фактов EDB, а $IC^{EDB'}$ – ограничения целостности, накладываемые на множество продуцируемых фактов EDB'. Поскольку в реляционной модели отсутствует механизм целостности хранимых запросов, механизмы, аналогичные $IC^{EDB'}$, в ней не представлены.

4. Использование дедуктивных баз данных в задачах интеграции реляционных систем

Достаточно часто реляционные системы используют для хранения табличных документов с нарушением требований реляционной модели. Это связано с ошибками и недоработками при проектировании, к которым можно отнести отсутствие нормализации и различную степень абстракции данных. Интеграция таких систем является достаточно сложной задачей и требует более мощных и выразительных средств, нежели реляционные языки.

Для решения этой задачи предлагается использовать аппарат дедуктивных баз данных, значительно расширяющий операционную спецификацию реляционных систем. Как было показано выше, все основные структурные компоненты реляционной модели выразимы средствами логического программирования, следовательно, их можно однозначно отобразить в дизъюнкты ДБД. Применение дедуктивных правил вывода к фактам, представляющим различные неоднородные системы баз данных, позволит представить их в виде единого виртуального множества данных, над которым можно осуществлять запросы. Общую схему работы предлагаемого метода представим на рис. 5.

4.1. Архитектура системы интеграции неоднородных баз данных на основе ДБД

Как было показано выше, любую реляционную базу данных можно представить в виде логической программы *P* и множества фактов *EDB*. Следовательно, множество интегрируемых систем баз данных $\{DB_1, DB_2, \ldots, DB_n\}$ можно представить как совокупность множеств фактов $\overline{EDB} = \{EDB_1, EDB_2, \ldots, EDB_n\}$ и логической программы $\overline{P}$, включающей правила, заданные в $P_1, P_2, \ldots, P_n$.

Поскольку, интегрируемые базы данных зачастую имеют различную степень нормализации и абстракции данных, а также могут содержать нарушения требований реляционной модели, необходимо задать механизм приведения локальных структур данных к глобальной виртуальной схеме. Это достаточно просто реализовать при помощи правил вывода вида (6).

Рассмотрим пример. Пусть даны две базы данных, представленные отношениями R_1 и R_2, R_3 (рис. 6). Представим их с помощью предикатов p_1, p_2, p_3 соответственно.

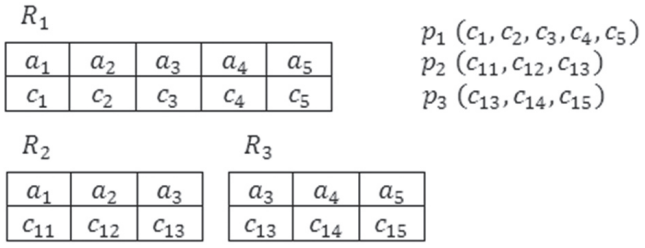


Рис. 6. Отношения R_1, R_2, R_3 и их представление в виде фактов p_1, p_2, p_3

Поскольку предикаты p_1, p_2, p_3 обладают различным количеством аргументов, определим правила вывода, которые приведут описанные предикаты к глобальной схеме. Глобальную схему обозначим при помощи предиката *p*:

$$p(X_1, X_2, X_3, X_4, X_5) \leftarrow p_1(X_1, X_2, X_3, X_4, X_5).$$
$$p(X_1, X_2, X_3, X_4, X_5) \leftarrow p_2(X_1, X_2, X_3), p_3(X_3, X_4, X_5). \qquad (22)$$

Подробнее правила преобразования различных вариантов неоднородных структур к глобальной схеме рассмотрены в [8].

Рассмотрим схему осуществления запроса над глобальной схемой (рис. 7).

Рис. 5. Схема интеграции реляционных СУБД на основе ДБД

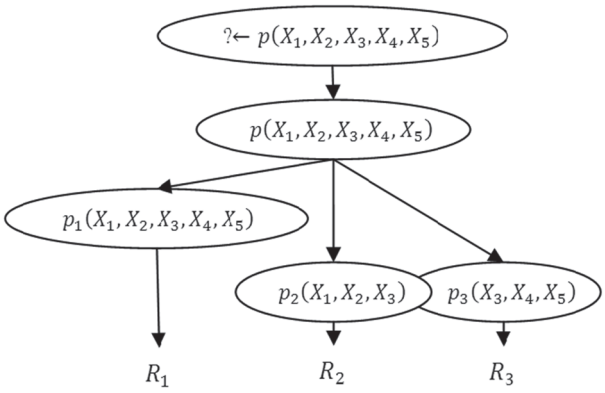


Рис. 7. Схема осуществления запроса над глобальной схемой $\overline{EDB}$

При осуществлении запроса над глобальной схемой $\overline{EDB}$ машина вывода Datalog, используя алгоритмы нисходящего вычисления [3], конструирует дерево доказательства запроса, начиная с предиката заданного в глобальной схеме и заканчивая на нижнем уровне, содержащем факты локальных экстенсионалов.

Выводы

В силу исторически сложившихся обстоятельств реляционная модель данных является доминирующей на рынке современных систем управления базами данных. Рассмотренные недостатки не позволяют решать ряд определенных задач из-за слабой выразительности реляционной модели. В то же время дедуктивная модель имеет более мощный манипуляционный компонент за счет использования логического программирования. Описанный в работе ряд отображений компонентов реляционной модели в элементы дедуктивной позволяет утверждать о возможности однозначной трансляции одной модели в другую. Представление реляционных систем в дедуктивном виде позволит решить ряд проблем, связанных с интеграцией неоднородных баз данных в силу большей выразительной мощности Datalog и отсутствия жестких требований к типизации данных.

Полученные результаты дают основания для дальнейшего изучения механизма отображения ограничений целостности, заданных в локальных системах, в ограничения целостности глобальной схемы. Поскольку ограничения целостности, описывающие локальную схему, не всегда допустимы для глобальной схемы, а их удовлетворение может привести к потере данных.

Список литературы: 1. *Дейт, К.Дж.* Введение в системы баз данных [Текст]: пер. с англ. – М.: Вильямс, 2006. – 1071 с. **2.** *Кодд, Э.Ф.* Реляционная модель данных для больших совместно используемых банков данных [Текст] / Э.Ф. Кодд // Системы управления базами данных. – 1995. – № 1. – С. 145-160. **3.** *Чери, С.* Логическое программирование и базы данных [Текст] : пер. с англ. / С. Чери, Г. Готлоб, Л._Танка.– М.: Мир, 1992. – 352 с. **4.** *Ульман, Дж. Д.* Введение в системы баз данных [Текст]: пер. с англ. – М.: Лори, 2000. – 374 с. **5.** *Калиниченко, Л.А.* Методы и средства интеграции неоднородных баз данных [Текст] / Л.А. Калиниченко. – М.: Наука, 1983. – 424 с. **6.** *Zaniolo, C.* Key Constraints and Monotonic Aggregates in Deductive Databases [Текст] / C. Zaniolo // Computational Logic: Logic Programming and Beyond. –2002. – С. 109-134. **7.** *Гарсиа-Молина Г.* Системы баз данных. Полный курс [Текст]: пер. с англ. / Г. Гарсиа-Молина, Дж. Ульман, Дж. Уидом.– М.: Вильямс, 2003. – 1088 с. **8.** *Танянский, С.С.* Организация запросов к распределенным данным средствами логического программирования [Текст] / С.С. Танянский, Ю.А. Мальков // Бионика интеллекта: науч.-техн. журнал. – 2010. – №1(72). С. 118-121.

Поступила в редколлегию 14.09.2011

УДК 004.652.4, 004.652.6

Відображення елементів реляційної моделі даних в елементи дедуктивної моделі / С.С. Танянський, Ю.А. Мальков // Біоніка інтелекту: наук.-техн. журнал. – 2011. – № 3 (77). – С. 136-142.

В статті розглянуто відображення компонентів реляційної моделі в компоненти дедуктивної, що дає підстави для побудови системи інтеграції з локальнонезалежними даними, заснованої на використанні логічного програмування та дедуктивного представлення бази даних.

Іл.: 7. Бібліогр.: 8 найм.

UDK 004.652.4, 004.652.6

Mapping relational model items to deductive model items / S.S. Tanyansky, Y.A. Malkov // Bionics of Intelligense: Sci. Mag. – 2011. – № 3 (77). – P. 136-142.

This article was focused on mapping items of relational model to deductive model items. This mapping provides a basis for constructing integration system with locally independent data, which based on application of logic programming and deductive presentation of databases.

Fig.: 7. Ref.: 8 items.

УДК 004.853

О.М. Почанский
ХНУРЭ г. Харьков, Украина, pochansky.oleg@yandex.ru

ИЗВЛЕЧЕНИЕ ЧАСТИЧНО-СТРУКТУРИРОВАННОЙ (ЗНАЧИМОЙ) ИНФОРМАЦИИ ИЗ ДИНАМИЧЕСКИХ WEB-ДОКУМЕНТОВ

В данной статье рассматривается модель системы, которая отвечает за извлечение значимой информации из web-документов. В основе модели лежит метод разделения web-страницы на структурные блоки. Каждый из них должен проходить проверку на наличие шумов. Те данные, которые прошли проверку и не содержат шумов, сохраняются в базе знаний системы.

ИЗВЛЕЧЕНИЕ ЗНАНИЙ, БАЗА ЗНАНИЙ, ЗНАЧИМАЯ ИНФОРМАЦИЯ, ШУМ, ОНТОЛОГИЯ, ДИНАМИЧЕСКИЙ WEB-ДОКУМЕНТ

Введение

Проблеме извлечения значимой информации из web-документов посвящается огромное количество научных статей и прикладных разработок в течение длительного промежутка времени (начиная с конца 20 века и по наши дни). Но в большинстве существующих решений присутствует ряд ограничений или допущений, связанных с написанием эффективных алгоритмов работы специализированных систем, способных проанализировать и обработать почти любой источник, содержащий необходимые данные [1]. В первую очередь, это связано со сложностями понимания ими сути значимой информации и отличием ее от другой информации, которая не удовлетворяет требованиям или критериям, поставленным перед системой. Таким образом, это позволит утверждать о том, что данная проблема в полной мере не решена, а значит, исследование в данной области не утратило своей актуальности.

1. Причины возникновения данной проблемной области

Попробуем разобраться в причинах возникновения указанной проблемы. Итак, для начала остановимся на определении общего понятия термина информация. Данный термин означает сведения, передаваемые источником получателю (приемнику). Он всегда связан с материальным носителем, с материальными процессами и имеет некоторое представление. Информация, представленная в какой-либо форме, называется сообщением. Сообщения представляются в виде сигналов и данных. Сигналы используются для передачи информации в пространстве между источником и получателем, а данные — для хранения (т. е. для передачи во времени) [2].

В нашем случае значимая информация будет рассматриваться в виде произвольного набора данных, сгруппированных в рамках заданной общей предметной области. По способу упорядочивания данных в различных типах документов их можно разделить на три вида: неструктурированные, частично-структурированные и структурированные [3].

Как правило, электронные документы составлены в произвольной форме на естественном языке и содержат неструктурированные данные. Примером такого документа может быть статья с произвольной тематикой, опубликованная в журнале или газете. Частично-структурированные данные содержат в основном электронные Web-документы различных расширений (php, aspx, jsp, htm и т.д.), оформленные в текстовом формате HTML, где они описываются с помощью специальных тэгов. И наконец, структурированные данные содержат XML-документы и базы данных [1]. Их относят к данному типу благодаря наличию специфических инструментов. В случае XML это – DTD (преамбула документа, где определяются его компоненты и структура) и XML schema (язык описания структуры XML документа) [1]. В случае базы данных это – функциональная особенность программной оболочки, в которой она была создана, и специализированный язык запросов – SQL, с помощью которого можно обратиться к любому существующему элементу базы данных.

Следовательно, при исследовании вопроса извлечения значимой информации необходимо учитывать не только соответствие извлекаемых данных одной предметной области, но и тип источника, в котором они находятся.

При этом к основным критериям значимости любой информации можно отнести: актуальность, достоверность, полноту и «чистоту» находящихся в ней данных. Первые три критерия можно отнести к субъективным понятиям, которые могут быть выявлены экспериментальным путем. В свою очередь, последний критерий можно отнести к объективным характеристикам, так как он отвечает за содержательную часть [4] и отражает качество получаемой информации – сигнализирует о зашумленности (отсутствуют данные, которые не несут смысловой нагрузки в рамках заданной тематики). Следовательно, информация, содержащая данные, не удовлетворяющая поставленному критерию – незначима.

Также стоит учитывать, что Internet делает потенциально доступными огромные объемы ин-

формации и, тем самым, ставит новые проблемы эффективной работы с такими объектами. В ситуации «информационной перегрузки» особенно актуальными становятся автоматические методы работы с большими объемами информации [5].

Исходя из этого, можно предположить, что любая специализированная система, целью которой является извлечение значимой информации из определенного источника, должна заранее определить, с каким типом документа ей предстоит работать. А для этого системе необходимо скорректировать свой алгоритм работы автоматически под определенную структуру данных. При этом она должна разбить процесс обработки и извлечения знаний из источников информации на несколько этапов (рис. 1).

Как видно из рис. 1, специализированной системе необходимо выполнить ряд дополнительных действий, прежде чем перейти непосредственно к извлечению значимой информации из документа. Причем данные действия должны быть выполнены в четко определенной последовательности, в противном случае это может привести к сбою всей системы. Это оказывает негативное влияние на производительность почти любой системы с указанным выше принципом работы. В конечном счете, автоматическое определение типа документа увеличивает время поиска значимой информации для каждого исследуемого документа. Но если заранее определить его тип и выполнять анализ нескольких источников информации параллельно, то можно повысить производительность всей системы и тем самым уменьшить временные затраты на поиск необходимого документа (рис. 2)

Как видно из рис. 2, специализированная система за одну и ту же единицу времени может обработать в 3 раза больше документов при условии схожести алгоритмов и равном объеме извлекаемых данных. Хотя при этом область применения данной системы сужается из-за работы только с определенным типом документов.

В настоящие время существуют системы, способные работать как с документами любых типов, так и с одним, заранее определенным. Далее рассмотри некоторые из них.

2. Обзор существующих решений

Примером системы, которая способна работать с документами любых типов, является KIM – Semantic annotation platform [6]. Данная система отвечает за извлечение и обработку данных, получаемых из различных информационных источников.

Рис. 1. Этапы работы специализированной системы по извлечению информации из заданного документа любого типа

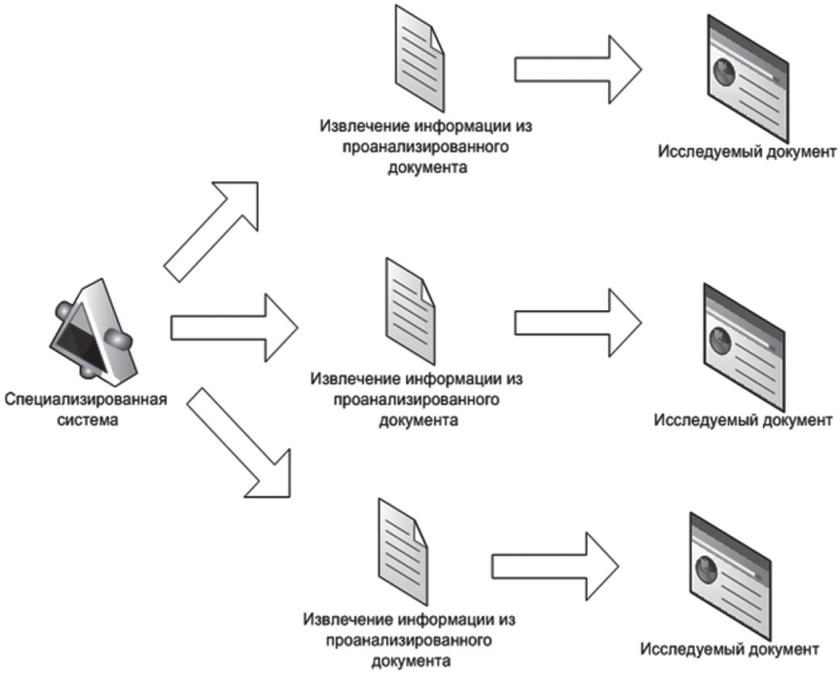


Рис. 2. Параллельная работа специализированной системы по извлечению информации из заданных документов заранее определенного типа

Она обеспечивает выполнение поставленных задач путем автоматического индексирования документов любого типа и построения их семантических аннотаций на основании проведенного анализа полученных данных. Для этого формируется база знаний, основанная на онтологии высшего порядка, в которой хранятся семантические аннотации в виде ключевых объектов (опорных слов) проиндексированных документов. Таким образом, в любом документе выделяются данные, которые соответствуют определенным классам, описанным в созданной онтологии. Каждый из классов, в свою очередь, делится на подклассы. Названия классов и подклассов соответствуют определенному общему термину, к которому можно отнести каждый из ключевых объектов (ключевой объект может соответствовать только одному классу или подклассу).

Все проаннотированные документы могут быть разделены на группы (на основании проанализированной meta-информации), каждая из которых соответствует определенной предметной области. Причем все данные в рамках одной группы связаны между собой. Это обеспечивает доступ ко всем ключевым объектам соответствующей тематики из документа любого типа, который был предварительно обработан данной системой.

К достоинствам KIM – Semantic annotation platform можно отнести:

1. Работу с любыми типами документов.

2. Взаимосвязь всех ключевых объектов различных документов, данные которых хранятся в базе знаний системы.

3. Возможность выполнения поиска документов по ключевым объектам.

К недостаткам KIM – Semantic annotation platform можно отнести:

1. Необходимость выполнения дополнительной обработки документа при создании семантических аннотаций.

2. Отсутствие проверки на зашумленность и повторяемость информации в исследуемых документах.

3. Неиспользование ключевых объектов для определения тематики документа. Ключевые объекты несут чисто информативный характер.

4. Отсутствие автоматизации составления логических правил, по которым ключевые объекты относят к какому-либо классу или подклассу, а также онтологии высшего порядка

5. Отсутствие возможности вносить в структуру онтологии высшего порядка изменения в процессе роботы системы.

Примером системы, которая способна работать с документами определенного типа, является программа, основанная на методе извлечения значимой информации из web-страниц путем их разделения на содержательную и навигационную часть. Данная система использует алгоритм, основанный на выделении повторяющихся фрагментов страниц одного сайта. Для этого на вход алгоритму подается директория с файлами, которая соответствует страницам одного сайта. После этого он анализирует данные файлы и выделяет в них повторяющиеся фрагменты, которые считаются навигационной частью. В зависимости от настроек, алгоритм либо удаляет навигационную часть из файла, либо выделяет навигационную часть специальными тегами. В свою очередь неповторяемые фрагменты относят к содержательной части, которая используется при информационном поиске документа, соответствующего формализованному запросу пользователя [4].

К достоинствам данной системы можно отнести:

1. Выполнение функций поиска данных, соответствующих запросу, сформированному пользователем, только в содержательной части web-документа.

2. Эффективную обработку информации на сайтах форумов, блогов, web-конференций, которые имеют стандартную структуру.

3. Возможность периодического мониторинга фиксированного списка сайтов.

К недостаткам данной системы можно отнести:

1. Необходимость выполнения дополнительной обработки web-документа при его разделении на содержательную и навигационную часть.

2. Низкая эффективность поиска web-документа, в случае если в его навигационной части содержится информация, релевантная сформированному запросу.

3. Отсутствие проверки на зашумленность содержательной части web-документа

4. Наличие случаев, в которых навигационную часть невозможно выявить или она выявлена неправильно на основе анализа совпадающих частей страниц.

5. Отсутствие функции лексического анализа содержательной части web-документа при поиске релевантных данных по сформированному запросу.

6. Работу только с одним типом документов.

Исходя из этого, можно сделать вывод, что независимо от количества поддерживаемых типов документов различными системами, они обладают рядом характерных недостатков. Главным образом они заключается в отсутствии возможности выявления и исключения шумов при извлечении данных из информационных источников.

Решением указанной проблемы посвящена данная статья.

3. Общие проблемы рассмотренных систем

Основываясь на приведенных выше исследованиях, было выявлено, что эффективного решения проблемы деления данных из любого информационного источника на значимую и незначимую

часть не предложено. В основном это вызвано наличием ряда следующих причин:

1. Большинство систем опираются на ключевые слова, которые могут иметь несколько значений и относится к разным тематикам, и поэтому возможно появление документов, не связанных со сформированным запросом.

2. Часто при анализе страницы web-документа исследуется только meta-данные, при этом другая информация не рассматривается, как следствие, возникают шумы в полученных данных.

3. В системах, направленных на извлечение информации из документов, отсутствуют критерии, характеризующие качество получаемой информации.

4. В процессе работы систем, направленных на извлечения информации, не формируются список «надежных» источников, данные из которых содержат наименьшее количество шумов. Это может отрицательно сказаться на эффективности работы всей системы в целом.

5. Большинство систем не выделяет значимую информацию из проанализированного источника и не хранит её в формате базы знаний. Тем самым данные системы сталкиваются с необходимостью повторной обработки данных во время появления новых поисковых запросов.

4. Постановка задачи

Исходя из изложенных выше причин, необходимо создать интеллектуальную систему, способную извлекать значимую информацию из документов с минимальным количеством шумов. Также она должна быть лишена основных недостатков и проблем, выявленных при рассмотрении схожих систем.

Для повышения эффективности будущей системы было принято решения обрабатывать динамические web-документы только с частично-структурированными данными (html-документами) в связи с тем, что она рассчитана на работу преимущественно с Internet ресурсами, где данный тип документов наиболее распространен.

5. Метод решения

На основании сформированной выше задачи предлагается рассмотреть модель работы системы извлечения значимой информации, которая способна повысить критерии качества получаемых данных. Данная модель отталкивается от предположения, что любой современный информационный web-документ можно разбить на различные структурные блоки. Каждый из этих блоков в свою очередь содержит определенные данные, посвященные заданной тематики, и выделен произвольным набором повторяемых html-тегов. В качестве примера одного из таких блоков возьмем часть html-кода, взятого из сайта футбольного клуб “Металлист” (http://www.metallist.kharkov.ua). Итак, данный блок имеет следующую структуру:

* <h2>МЕТАЛЛИСТ ОПРОС</h2> *

* Как, по Вашему мнению, завершится матч Металлист - Таврия? *

* голосовать *

Его границы были выделены по следующему принципу: ключевой html-тег, в данном случае это , не может быть частью другого тега. К примеру, тег <h2> является частью рассмотренного выше ключевого тега, следовательно, он не может быть началом другого структурного блока. Закрытие ключевого тега ( ) символизирует окончание описания данного структурного блока. Исключение составляют теги, отвечающие за формирование и описание общей структуры любой web-страницы.

Если учесть, что структуру современного динамического web-документа можно представить в форме ориентированного графа [7], корнями которого являются гипертекстовые ссылки, представленные в виде меню-навигации, то блок, который содержит ссылки на внешний источник или другую страницу из другого домена (за исключением ссылок встречаемых в обширных текстовых описаниях), можно принять за шум. А значит, они не должны учитываться при обработке web-страницы данной системой. Остальные информационные блоки анализируются и записываются в базу знаний в соответствии с критериями, которые были предварительно заданы пользователем.

Общий принцип модели системы, отвечающей за извлечения значимой информации из web-страницы, рассмотрен ниже (рис. 3).

Далее остановимся подробнее на критериях, которые предварительно задаются пользователем. В общем случае они могут выглядеть в форме простых пожеланий в формате предоставляемых данных. К примеру, пользователи могут сформировать списки любимых источников и обмениваться ими между собой в форме rdf-файлов (рис. 4).

Также в процессе работы данной системы осуществляется подсчет качества информации, которая содержится на текущей web-странице, по формуле:

$$I_q = I_{al} - I_b,$$

где I_q – процент значимой информации на текущей web-странице; I_{al} –процент всей информации на текущей web-странице (обычно равен 100%); I_b – процент шумов на текущей web-странице.

Основываясь на данной формуле, можно выстроить внутренний рейтинг приоритета обработ-

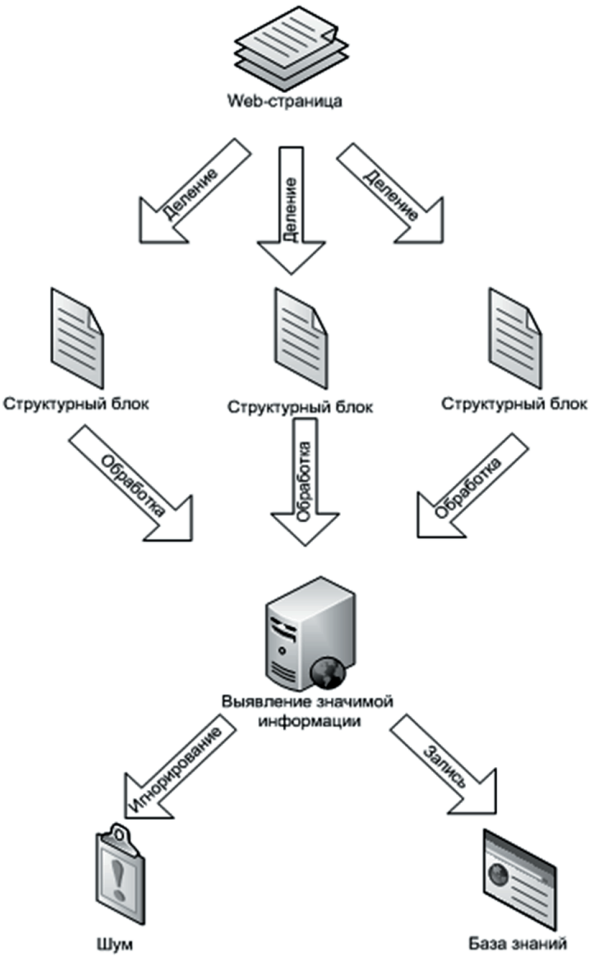


Рис. 3. Общая структурная схема модели системы, отвечающей за извлечение значимой информации из динамической web-страницы

ки информационных источников с наименьшим количеством шумов при условии схожести их тематик.

5. Анализ полученных результатов

В рамках рассмотренной модели системы, отвечающей за извлечения значимой информации из динамического web-документа, была сформирована база знаний сайта футбольного клуба “Металлист”. Каждый класс полученной онтологии соответствует пункту навигации данного сайта. Причем его название сформировано с использованием терминологический словаря онтологий по соответствующей предметной области [8].

Стоит отметить, что экземпляры каждого из этих классов хранят информацию, полученную при анализе web-страниц сайта футбольного клуба “Металлист” (рис. 5). Причем в этой онтологии данные могут быть представлены как текстом, так и картинками или видео. Также для каждой web-страницы формируются «белые» и «черные» списки, информация из которых автоматически считается значимой в первом случае или шумом, во втором.

Далее рассмотрим некоторые из основных полей, которые отображены на данном рисунке:

1. hasadress – хранит URL страницы, из которой были взяты данные;

2. hasname – хранит имя страницы, из которой были взяты данные;

3. haspictures – хранит URL картинок страницы, из которой были взяты данные;

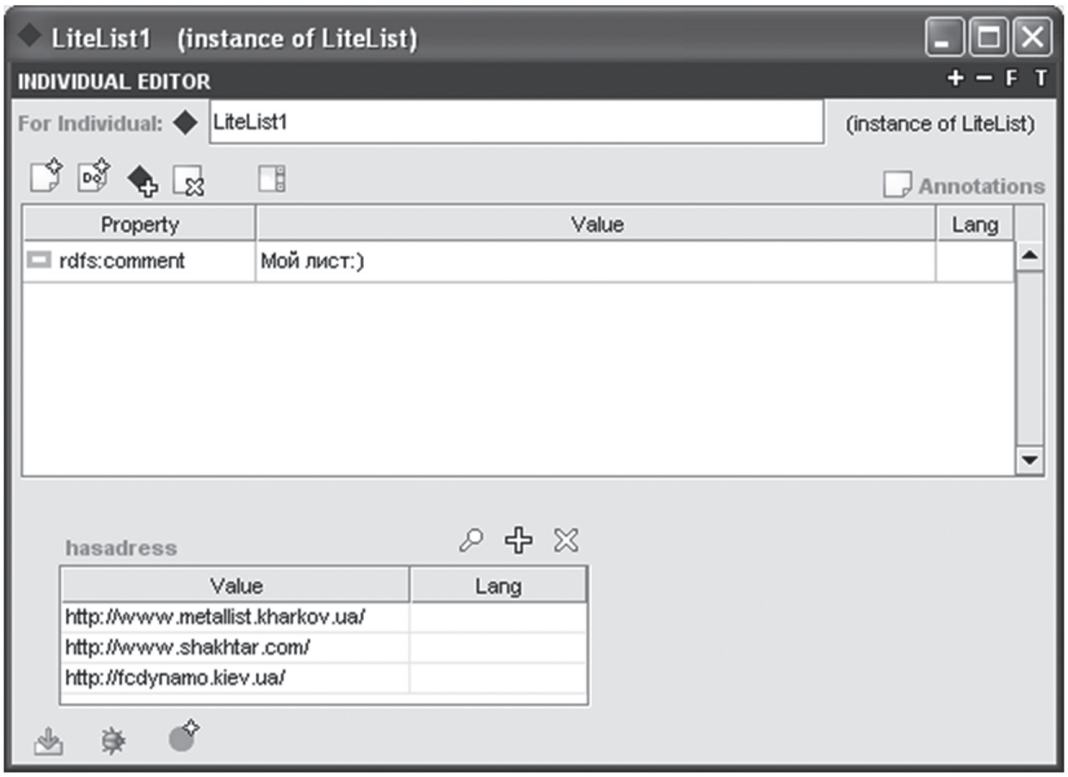


Рис. 4. Пример отображения списка любимых источников в редакторе онтологий Protйgй

4. hasLiteList – хранит URL страниц, которым доверяет система;

5. hasBlackList – хранит URL страниц, которым не доверяет система;

6. hasquality – хранит процент значимой информации от общего числа;

7. hasresorse – хранит значимую информацию, которая была взята из данной web-страницы (рис. 6).

В заключение можно сделать вывод, что полученная онтология позволяет сократить время поиска нужной информации за счет структуриро-

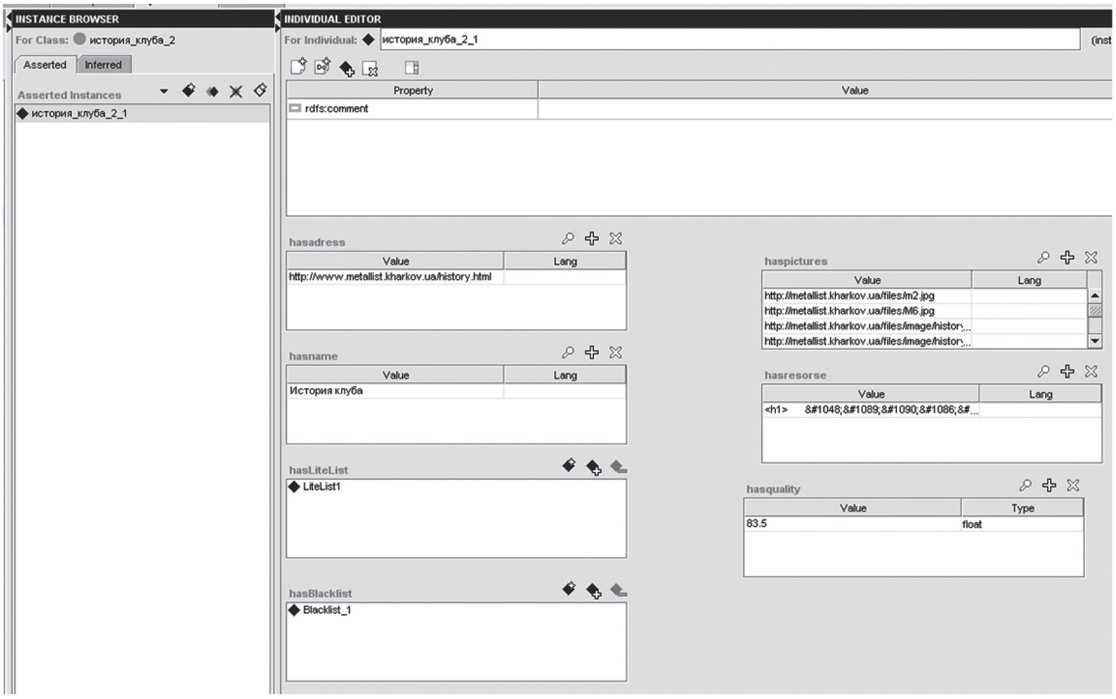


Рис. 5. Форма представление данных взятых из web-страницы сайта футбольного клуба “Металлист”

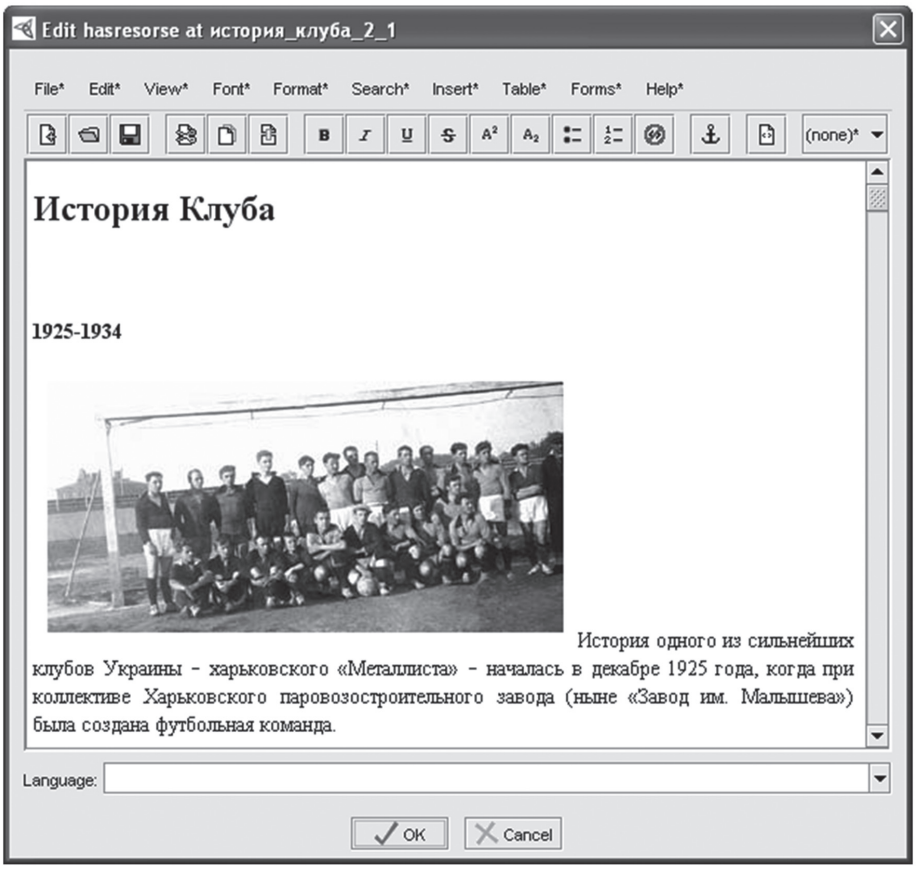


Рис. 6. Выделенная значимая информация из web-страницы (http://www.metallist.kharkov.ua/history.html)

вания источников данных. Причем в полученном результате будут отсутствовать шумы, что ускорит процесс его обработки и скажется на эффективности применения сформированной базы знаний в смежных системах.

Выводы

Полученная интеллектуальная модель системы, отвечающей за извлечения значимой информации из web-страниц, обладает рядом следующих преимуществ:

1. Извлекаемая информация проверяется на наличие шумов и повторяющихся данных, которые не записываются в базу знаний системы.

2. В процессе работы системы формируется белый список ссылок, которые должны быть проработаны в первую очередь, и черный список, ссылки из которого не анализируются, а полученные данные из них автоматически считаются шумом.

3. Хранение и обновления данных из проанализированных динамических web-документов осуществляется в формате единой базы знаний, через которую также выполняется поиск информации, релевантной сформированному запросу.

4. Для учета лексической взаимосвязи данных используется терминологический словарь онтологий.

5. Значимая информация в базе знаний разделяется за формой представления (картинки, текст, видео).

Подводя окончательную черту в описании модели системы, отвечающей за извлечения значимой информации из web-страницы, нужно отметить, что сформированная база знаний может использоваться при создании специализированных поисковых систем [8], а именно – в процессе анализа заданного web-документа, применяя и используя описанные выше особенности.

В последующих разработках планируется реализовать механизм проверки ссылок, встречаемых в обширных текстовых описаниях, а также применить описанную выше модель в рамках создание единой поисковой системы.

Список литературы: 1. *Chia-Hui, Ch.* A survey of Web Information Extraction [Text] / Ch. Chia-Hui, K. Mohammed, R.G. Moheb, F. S. Khaled. // IEEE Transactions on Knowledge and Data Engineering – 2006. №18/10. – С. 1411-1428. **2.** Информация [Электронный ресурс] / Википедия – интернет энциклопедия. – Режим доступа: URL: http://ru.wikipedia.org/ – 18.09.2011. **3.** *Беленький, А.* Текстомайнинг. Извлечение информации из неструктурированных текстов [Электронный ресурс] / А. Беленький // Журн. "КомпьютерПресс". – 2008. №10. Режим доступа: URL: http://www.compress.ru/article.aspx? id=19605&iid=905 – 18.09.2011. **4.** *Агеев, М. С.* Извлечение значимой информации из web-страниц для задач информационного поиска [Текст] / М. С. Агеев, И. В. Вершинников, Б. В. Добров // Интернет-математика 2005. Автоматическая обработка веб-данных. – М.:"Яndex", 2005. – С. 283-301. **5.** *Браславский, П.* Автоматическое реферирование веб-документов с учетом запроса [Текст] / П. Браславский, И. Колычев // Интернет-математика-2005. Автоматическая обработка веб-данных. – М. : "Яndex", 2005. - С. 485-501. **6.** *Popov, B.* KIM – Semantic Annotation Platform [Text] / B. Popov, A. Kiryakov, D. Manov, D. Ognyanoff, M. Goranov // Journal of Natural Language Engineering, №10/3-4. – С 375-392. **7.** *Ланде, Д.В.* Поиск знаний в Internet. Профессиональная работа [Текст] : пер. с англ. – М.: Издательский дом "Вильямс", 2005. – 272 с. **8.** *Почанский, О.М.* Модель построение адаптивных Web-страниц на основе интеллектуального анализа сети Internet [Текст] / О.М. Почанский // Журн. восточно-европейский журнал передовых технологий. – 2010. - № 4/7(46). – С. 66-69.

Поступила в редколлегию 19.09.2011

УДК 004.853

Витяг частково-структурованої (значущої) інформації з динамічних Web-документів / О.М. Почанський // Біоніка інтелекту: наук.-техн. журнал. – 2011. – № 3 (77). – С. 143-149.

В даній роботі описана модель інтелектуальної системи, що відповідає за вилучення значущої інформації з Web-сторінок. Це виконується шляхом поділу кожної сторінки аналізованого динамічного Web-документа на структурні блоки. Потім ці сторінки перевіряються на наявність шумів. А далі ті з них, які пройшли перевірку, зберігаються в базі знань. Результатом роботи системи є база знань, заповнена якісною інформацією (відсутні шуми) по заданій тематиці, яка може бути використана при створенні ефективних пошукових систем.

Іл. 6. Бібліогр.: 7 найм.

UDC 004.853

Removing partially-structured (significant) information from dynamic Web-documents / OM Pochansky // Bionics of Intelligense: Sci. Mag. – 2011. – № 3 (77). – P. 143-149.

The article describes the model of the intelligent system which is responsible for extracting meaningful information from Web-pages. Its main task is to divide each page of the analyzed dynamic Web-documents into different parts. Then they tested for the presence of noise, after that they saved into a knowledge base. The result of the system is the knowledge base that filled with quality information (without any noise), according to the chosen topic, which can be used to create effective search engines.

Fig. 6. Ref.: 7 items.

УДК 007.681.5:519.7

М.Ф. Бондаренко, З.Д. Коноплянко, Г.Г. Четвериков
ХНУРЕ, г. Харків, Україна, chetvergg@gmail.com

КОНЦЕПЦІЇ УНІФІКАЦІЇ ІНФОРМАЦІЙНО-ІНТЕЛЕКТУАЛЬНИХ ТЕХНОЛОГІЙ В СИСТЕМАХ МОВЛЕННЯ

Стаття присвячена аналізу проблеми створення систем штучного інтелекту, які дозволяють моделювати на логічному та апаратному рівнях процеси інтелектуального управління, що описані математичними операціями над природною мовою, і які є елементами *k*-значної структурної організації інформаційно-інтелектуальних технологій. Показана необхідність і можливість розробки загальної теорії побудови інтелектуальних систем штучного інтелекту, яка могла б стати методологічною основою створення нових інформаційних технологій.

ЛОГІКА, ЗНАННЯ, ПРИРОДНА МОВА, ЛІНГВІСТИЧНІ ТЕХНОЛОГІЇ, *k*-ЗНАЧНА СТРУКТУРА, АСП-СТРУКТУРА, ШТУЧНИЙ ІНТЕЛЕКТ

Вступ

Стаття є логічним продовженням досліджень авторів в галузі систем штучного інтелекту, що були розглянуті раніше, наприклад, [1–3]. Так, в умовах роботи реальних систем із високим рівнем невизначеності інформації для побудови інтелектуальних систем неминуче використання нових інформаційних технологій, зорієнтованих на потоки контекстно-залежної інформації; фактично необхідна розробка природно-мовних принципів побудови інтелектуального управління – теорії інтелектуальних систем управління (ІСУ) – для систем вищих рівнів системної складності. Отже, для правомірного використання скінченного автомата (комп'ютера) у складі інтелектуальної системи теорія повинна розглядати можливість побудови абстрактних конструкцій, що реалізують не обчислювані в звичному значенні об'єкти. Все, що дотепер винайдене, всі узагальнені функціональні перетворення можна застосувати тільки для подання зліченних сукупностей процесів, поданих потоками, хоча і нескінченними, але однорідними, що складаються з нескінченно малих невиразних сутностей. У разі відкритих (інтелектуальних) систем ми маємо справу з незчисленною множиною потоків, кожний із яких може розкритися в більш ніж зчисленну сукупність потоків, що складаються з нескінченної різноманітності структур[4].

Мета роботи. Основним завданням цієї роботи є викладення новоствореної концепції уніфікації методів та засобів побудови просторових багатозначних структур мовних систем. Предметом досліджень є моделювання інтелектуальної діяльності людей як у зовнішньому її прояві (вирішення складних завдань, розуміння природної мови, інтерпретація візуальної інформації та мови), так і у внутрішньому (накопичення, надання і використання знань).

Згідно з завданням роботи та враховуючи основні аксіоми теорії інтелектуальних систем управління інтегруємо необхідні та уже розроблені природномовні принципи побудови інтелектуального управління і систем штучного інтелекту. У цій роботі, перш за все, хотілося б показати необхідність і можливість розробки загальної теорії побудови інтелектуального управління і систем штучного інтелекту, яка могла б стати методологічною основою безпосередньо для створення нових інформаційних технологій[1–5].

1. Принципи побудови природно мовних систем штучного інтелекту

Морфологічний аналіз. Задача морфологічного аналізу [1, 3] полягає в ідентифікації словоформ та присвоєнні кожній словоформі комплексу морфологічної інформації (КМІ). Такий комплекс складається із морфологічно-інформаційних рядків (МІ-рядків) з наступною структурою:

– номер, <(основа чи ознаки основи), МІ >, (де номер – порядковий номер даної словоформи у фразі);

– основа (ознака основи) – код семантичної ознаки, номер синтаксичної чи семантичної моделі керування, що присвоєні даній основі в словнику основ;

– МІ – частина мови та її граматичні категорії: рід, число, відмінок, час, особа тощо.

Існує два методи реалізації [1, 3] морфологічного аналізу (МА): словниковий (декларативний) (використовується для аналізу мов із нерозвинутим відмінюванням слів (англійська, французька тощо)); алгоритмічний (процедурний) морфологічний аналіз. При МА здійснюється розчленування словоформ на основу та закінчення і в словниках зберігаються як основи, так і їх закінчення. МА здійснюється шляхом пошуку в складі словоформи, що аналізується, деякої словникової основи та певного словникового закінчення. Потім виконують порівняння інформації про основу та закінчення і отримують комплекс морфологічної інформації для всієї словоформи.

Під час МА змінюваної словоформи її кінцеву частину за черзі порівнюють із закінченнями словника. Після порівняння ту частину словоформи,

що співпала, відокремлюють і отримують припустиму основу (ПОС), припустиме закінчення (ПЗК) та припустиму морфологічну інформацію (ПМІ).

Дані про ПЗК (ПМІ) зчитують із словника закінчень (морфологічної інформації). Потім переходять до пошуку інших ПЗК, ПОС та ПМІ.

На другому кроці аналізу словоформи виконується ідентифікація її можливих основ шляхом перевірки збіжності отриманих припустимих основ із вмістом машинного словника основ.

На третьому кроці МА словоформи порівнюється інформація з тими припустимими основами та ПЗК, що отримали підтвердження за допомогою словника основ.

Ефективність МА суттєво залежить від виду подання машинних словників у пам'яті ЕОМ та способу їх кодування. При цьому доцільно мати окремий допоміжний словник перенумерованих основ, що наявні у одному примірнику та розташовані в алфавітному порядку.

Для подання значень граматичних категорій будь-якої словоформи використаємо 9-ти розрядний 10-значний код. Порозрядно у $p(1)$, $p(2)$ – закодовано частину мови словоформи, $p(3)$ – тип та клас прийменника чи розрядів за значенням (іменника, повного прикметника); $p(4.i)$ – дієслово 1–3 особи відповідно; $p(5)$ – код значення числа (однина, множина); $p(6)$ – код відмінка (називний, родовий, давальний ...); $p(7)$ – код категорії пасивності-активності; $p(8)$ – код часу (теперішній, минулий, майбутній); $p(9)$ – код категорії виду (доконаний, недоконаний) закінчення [1].

Для формування одного МІ-рядка до всієї словоформи порівнюють код основи та код закінчення на відповідність їх перших п'яти розрядів, якщо співпадання немає, то дані несумісні. Для порівняння вибирають черговий код закінчення. Якщо відповідність встановлена, то решту розрядів результуючого коду формують за правилами 10-значної диз'юнкції значень відповідних розрядів кодів основи та закінчення. При цьому попередньо перевіряють умову співпадання операндів чи рівність одного з них нулеві.

Таким чином, описаний алгоритм дозволяє інтерпретувати різноманіття граматичного оброблення українських флексій (аналіз, синтез, нормалізація, корегування помилок тощо) за допомогою розв'язків канонічних рівнянь виду $L_{\varphi}(X, Y) = 1$.

Синтаксичний та семантичний аналіз. Тепер ми переходимо до розгляду необхідних відомостей про контекстно-залежні (КЗ) мови, тобто про реалізацію інтелектуального управління, коли нас цікавить можливість мінімізації яких би то не було втрат при використанні скінченого автомата як основи КЗ-мови.

КЗ-мова (природна мова людини) в галузі наукової термінології володіє великою невизначеністю, що пояснюється частково поліморфізмом і контекстно-залежним поданням наукової інформації, а іноді (і для інформатики і для інтелектуального управління це особливо важливо) – недбалістю використання термінів.

Семантика визначає відношення між знаками і їх концептами, тобто задає зміст чи значення конкретних знаків[7].

Слова в мові не йдуть у довільному порядку і закони їх упорядкування є предметом синтаксису. Синтаксис описує структуру можливих фраз. Опис синтаксичних структур використовує наступні граматики (формалізми) [8–14]:

– дерева синтаксичного підпорядкування;

– системи складових;

– розширені мережі переходів.

Таким чином, синтаксичний аналіз використовує заготовлені за допомогою граматики шаблони вхідних фраз із метою виявлення (встановлення) відповідності між послідовністю, що аналізується, та значущими синтаксичними структурами. Основним формальним засобом математичного опису природної мови є алгебра скінченних предикатів (АСП), оскільки мова є скінченою, дискретною та k-значною. АСП у процесі її дії використовує процедури розв'язування рівнянь, а не алгоритмів. У роботі [8] в процесі синтаксичного розбору природномовних висловлювань розроблено метод побудови синтаксичних дерев для аналізу простих речень. Для встановлення інтегральних закономірностей обробки природної мови проаналізуємо, що власне відбувається у процесі аналітичних досліджень вищих лінгвістичних механізмів дії російської мови.

Семантика. Вихідним матеріалом для семантичного аналізу природної мови є синтаксична структура фрази чи її фрагмента, а також дані про значення словоформ. Основна задача семантичного аналізу – це зняття неоднозначності, морфологічної та лексичної багатозначності словоформ та синтаксичних структур речень.

У роботі [8] об'єктом математичного моделювання є словосполучення, що мають інструментальне значення. Для аналізу семантики повідомлення на природній мові необхідно визначити *значення одиниць повідомлення.* Значення слів класифікують згідно з набором апріорних ознак: дія – інструмент дії або, іншими словами, дієслово (конкретної дії чи акційне) – іменник у певному відмінку (називний, родовий, давальний тощо).

Метод, що покладений в основу – метод семантичного аналізу. Для аналізу семантики повідомлення на природній мові необхідно визначити значення одиниць повідомлення. Значення слів класифікують згідно з набором апріорних ознак: дія – інструмент дії або, іншими словами, дієслово (конкретної дії чи акційне) – іменник у певно-

му відмінку (називний, родовий, давальний тощо). Для дослідження семантики словосполучень такого виду використовують семантичні мережі та зв'язаний з ними математичний апарат, у даному випадку для кожного словосполучення у вигляді двох графів. У якості формального апарату подання семантики використовують АСП [9].

Якщо побудова результуючого графа і відповідного йому предиката АСП можлива, то це означає, що розглянута комбінація слів утворює осмислене словосполучення, а також можливо встановити чи володіє об'єкт деякою властивістю; визначити якими властивостями повинен володіти інструмент для завершення дії та відновити іменник чи дієслово за набором ознак тощо.

Наступною фазою досліджень стала робота [10] про змістовну інтерпретацію алгебри ідей. Тут об'єктом математичного моделювання стали: смислова однозначність; ситуаційно-предикатна; ситуаційно-множинна; ситуаційно-кодова ідея.

У роботі [11] об'єктом математичного моделювання обрана семантика похідних слів із модифікаційними значеннями.

Відповідно у роботі [12] досліджено міжморфемні семантичні зв'язки, які виникають у процесі словоутворення між префіксними та кореневими морфемами, кореневими та суфіксними морфемами, а також між основами та закінченнями.

І нарешті робота [13, 14] присвячена процедурам приписування словоформам семантичних ознак і моделюванню семантики похідних слова (мідь-мідний, залізо-залізний).

У всіх випадках ми маємо управління як результат "оптимізованого інформаційного пошуку", мета якого – вироблення управляючого рішення, тобто відповідного повідомлення на основі аналізу структури даних, закладеної в машину при проектуванні інформаційної системи, і її відповідного наповнення.

"Апріорна семантика" присутня лише у "власне даних", тобто в значеннях первинних сигналів і в "словнику", в наборі термінів, які "стали константами, що забезпечують життєздатність системи".

У випадку відкритих інтелектуальних систем управління структурно-динамічне, мета якого – формування деякої "структури знань", змінної в часі іменованої структури зв'язків, організація "інформаційного резонансу". Власне вироблення того чи іншого рішення є завданням найважливішої, але вторинної, похідної від основного завдання системи – "бути в курсі всіх змін і в постійній готовності до сприйняття сенсу запиту чи повідомлення за результатами попередньої інформаційної посилки".

Далі семантика накопиченої інформації вже забезпечить у потрібний момент вироблення структури зв'язків, що може служити для перетворення в будь-які дії: організаційні, правові, судові, особові, моральні, чим, власне кажучи, і визначається інтелектуальне управління в такій інженерній постановці.

Складається враження, що семантика – довільна вигадка теоретиків, а в природі її і немає зовсім.

Насправді семантика нікуди не пропала, просто адекватно реалізований апарат "здвоєної W-граматики [4] акуратно і послідовно "розрізає" її на дві частини – "константну", яку вона вкладає в БД у вигляді літералів і зв'язків і іменує після цього ієрогліфом словника, і "плинну", змінну, яка "залишається в розпорядженні" ПЗ і користувача. Велика частка народів Землі успішно чинить так само – користується ієрогліфами, до яких ми повинні відносити і всю термінологію професійних сленгів. Звідси випливають два висновки.

Теоретичний – семантика по суті динамічний об'єкт зі всіма витікаючими наслідками.

Практичний – чи варто використовувати в реалізації програмних продуктів "функції семантичних оцінок" тощо, бо це не більше ніж часткова статистика. Саме часткова, зроблена конкретно і на конкретному матеріалі і що має дуже опосередковане відношення до решти всіх випадків. Користь від неї сумнівна, зате неприємності – гарантовані, у чому ми пересвідчились на прикладах робіт.

Намагаючись досягти "чистої абстракції", ми розриваємо деякі взаємодії, можливо "знищуємо" процеси, які відповідальні за виникнення та існування досліджуваних феноменів. Тим самим цілком імовірно знищується сама можливість рішення початкової проблеми, відбувається її неусвідомлена підміна на довгі роки, до тих пір, поки знов не проявлять себе факти, що не вклалися в поточний варіант "чистої абстракції".

Другий варіант – варіант відмови від проблеми або, що те ж саме – введення революційних перебудов, фактично є логічним доповненням першого, принаймні до тих пір, поки ми орієнтуємося на будь-які раціональні чи ірраціональні методи *винайдення* аксіоматики.

Дійсно, чому б не припустити, що істина навіть не посередині, а в нерозривному зв'язку вказаних позицій? Повторимо тут висновки про необхідність постійної зміни правил формальної логіки, але вже з посиланням на Л. Керрола – "суть полягає в тому, що правила (гри) постійно змінюються".

Можливо, причина якраз у "правилах виникнення зміни правил" і зовсім не потрібно нескінченних сходів "правил-над-правилами" саме через динаміку системи. А нескінченні ієрархії "правил зміни правил" у багатьох дослідженнях виникають виключно від того, що розглядаються "мертві" статичні моделі чи довільно вирізані шматки систем.

Інакше кажучи, якісь із цих "правил над правилами", об'єктивно існуючі закони відкритих систем

можуть проявляти свою дію (або взагалі виникати!) тільки в деякій "мінімальній сукупності взаємодій". Достатньо незначного спрощення "моделі", відходу від реальної ситуації, щоб або ніколи їх не знайти, або одержати цілком реальні і несподівані наслідки їх дії. Саме ця ситуація найбільш типова для всіх модельних реалізацій інформаційних систем [15]. Практично весь попередній матеріал є коментарем цієї проблеми і пошуком інженерних шляхів її рішення.

Взагалі кажучи, сказане в деяких аспектах вже давно всім знайоме, нікого не дивує, що в світі елементарних частинок, у квантовій механіці діють закони окремі, не схожі на макромеханіку. В світі елементарних частинок взаємодія, розпад одних структур і утворення інших відбуваються "в перебігу одного кванта часу", тобто "усередині процесу нічого немає" (або ми поки що не уміємо уявити собі, що там є). Навпаки, весь зміст існування інформаційних систем полягає у процесі перетворення одних структур в інші [15, 16].

Звернемо увагу, що при цьому для термодинаміки і статистичної механіки, тобто для подання ентропійних процесів, виявляється зручним і адекватним подання процесів у вигляді потоків однорідних нерозрізнених сутностей, які можна роздрібнити "до нескінченно малого стану". Навпаки, сутність інформаційних систем, що самоорганізуються, полягає у взаємодії різних потоків, у нескінченній різноманітності структур, які можливо представити в скінченному вигляді, у вигляді скінченних алфавітів, знаків, складених із кінцевого числа розрізнимих елементів – "бітів".

Шляхом аналізу значної кількості текстів на проблемно-орієнтованій мові деякої галузі науки чи практичної діяльності можна виділити і стандартизувати обмежену, але достатню групу відношень R і правил побудови на їх основі логічних висновків, що практично виконуються на всьому інформаційному масиві.

Можна виділити декілька рівнів мовного подання знань у його зв'язку з даними:

• *Рівень інтелектуальної системи* – знань і дані існують у формі мовної моделі предметної галузі (моделі, в багатьох, якщо не у всіх, випадках невід'ємної від самої предметної галузі, що існує як мовний опис) і як опис складових цієї системи на рівні контекстно-залежної мови;

• *Рівень інформаційної системи* – знання і дані існують у формі мовної моделі (саме моделі, яка виділяє з реальної системи дещо, що визнається "істотним") на основі використання контекстно-незалежних мов;

• *Рівень математичної моделі* – дані, формалізовані на рівні мови формул і передавальних функцій, містять у собі знання як формалізовані правила і апарат продукування висновків;

• *Рівень фактографічної моделі* – текстові записи з фіксованою на рівні мови їх подання системою відношень між ними (наприклад, табличний запис).

Наведений перелік характеризує різні рівні роботи з інформацією, виділяє якісно різні групи інформаційних технологій, особливо підкреслюючи можливості роботи із знаннями на рівні контекстно-залежного опису предметної галузі, тобто на рівні семантики і контекстної залежності трактування кожної інформаційної посилки. Таким чином, між першим і подальшими рівнями подання знань проходить стіна, що відокремлює інтелектуальні системи від неінтелектуальних, а саме поняття знань не є винятковою приналежністю інтелектуальних систем і кардинально змінюється на кожному рівні його подання.

2. Формалізація принципу уніфікації багатовходових k-значних структур

Для того щоб в узагальненому вигляді аналітично описати та сформулювати принцип симбіозу в просторових k-значних структурах, відзначимо, що присвоювання значень алфавіту $E_k \in \{0, 1, 2, ..., k-1\}$ здійснюється за рахунок вхідних сигналів $X \in \{0, 1, 2, ..., k-1\}$. У загальному випадку кожному станові вхідного сигналу X відповідає певне значення вихідного сигналу $Y \in \{0, 1, 2,...,k-1\}$. Функціональний перетворювач є багатополюсником, що має n входів і один вихід, який відображає прямий добуток n множин $\{x\}_i$ ($i = 1, 2, ..., n$) на множину $\{y\}$. Об'єднання множин $\sum_{i=1}^{n}\{x\}_i$ називається вхідним алфавітом, а множина Y – вихідним [6].

В узагальненому вигляді універсальні просторові k-значні структури, згідно з узагальненим способом завдання функцій k-значної логіки з допомогою таблиць істинності та принципу симбіозу дво- та багаторівневого кодування і засобів, включають до свого складу такі компоненти: паралельний аналого-цифровий перетворювач (елемент розпізнавання k-значної змінної: ОБПЕ – оборотний багатозначний пороговий елемент); дешифратор (ДШ), селектор, комутатор, паралельний цифро-аналоговий перетворювач (ключовий комутатор, або підсумовувач струмів) [5,6].

Вирішення задач формалізації принципів організації багатовходових k-значних структур забезпечує побудову новітньої концепції синтезу, зокрема, тривходового універсального просторового функціонального перетворювача (УПФП) рис. 1 для високошвидкісних обчислювальних систем; застосування просторового та часового паралелізму на структурному й алгоритмічному рівнях та k-значних методів кодування; створення процедурних і функціональних мов, паралельних машин баз знань і логічного виводу [5, 15–17]. Отже, зрос-

тання значності та числа вхідних змінних універсальної k-значної структури веде до суттєвих змін у побудові компонентів, що входять до її складу.

Зокрема, на вході структури зростає пропорційно k число елементів розпізнавання; структури селекторів і комутаторів перетворюються в n-вимірні об'ємно-просторові утворення, а на виході структури зростає пропорційно k число ключів. Таким чином, в узагальненому вигляді універсальна просторова структура може бути описана системою ознак $S \to P \to V \to P \to S$, де V – об'єм (вимірність простору селектора і комутатора); S – статична ознака (кожному зі символів багатозначного структурного алфавіту ставиться у відповідність один із рівнів напругу чи струму); P – просторова ознака (символи алфавіту зображаються збудженим станом одного з k просторових полюсів).

Зауважимо, що структурні побудови універсальних k-значних функціональних перетворювачів на засадах принципу уніфікації з відповідними операційними засобами (значність, специфічні функтори, об'ємно-просторовий паралелізм, багатомісність функцій) утворюють нову паралельну обчислювальну математику, аналогів якої на нинішній день не існує у світі, а фундаментальні дослідження її щодо детальнішого аналізу можливостей застосування під час побудови високошвидкісних обчислювальних систем ще навіть, по суті, і не починалися й залишаються завданням ближчого і подальшого майбутнього.

Загалом виявляється [4], що метаструктура рішення будь-якої задачі складається з трьох рівнів деяких структур даних. І не лише метаструктура кінцевого стану, тобто деякого "статичного опису" рішення, але й структура самого процесу побудови цього рішення. Звернемо увагу на наступні дві обставини:

- процес отримання рішення по суті рекурсивний, зобов'язаний бути таким, оскільки апріорні оцінки критерію зупинки, за визначенням, відсутні.
- поняття структури даних ніяк не визначене апріорі, структура ніяк не задана.

З існування загального закону рекурсії структур і метаструктур виходить цікавий висновок: кожна проблемна галузь має не більше трьох рівнів метаструктур над структурою "елементної бази". Назвемо цей висновок "правилом трійки".

Усе сказане еквівалентне припущенню про існування деякого *закону рекурсії структур, метаструктур і процесів.* Процесів це стосується в тому сенсі, що процеси самоорганізації *не можуть не бути рекурсивними.*

3. Перспективи розвитку систем ШІ та комп'ютерна абсолютно універсальна система (КАУС)

Отже, слід наголосити на двох напрямах, за якими робляться спроби вирішення проблеми штучного інтелекту:

Перший з них – традиційний, в рамках розробки штучного інтелекту. Загальним для робіт цього напряму є те, що в них робляться спроби побудувати інтелект на основі дослідження і моделювання функцій інтелекту людини, наприклад, [17].

Другий з даних напрямів, який сформувався недавно [18], – це спроба на основі дослідження механізмів самоорганізації популяцій бактерій (простих, у порівнянні з людиною, організмів) від-

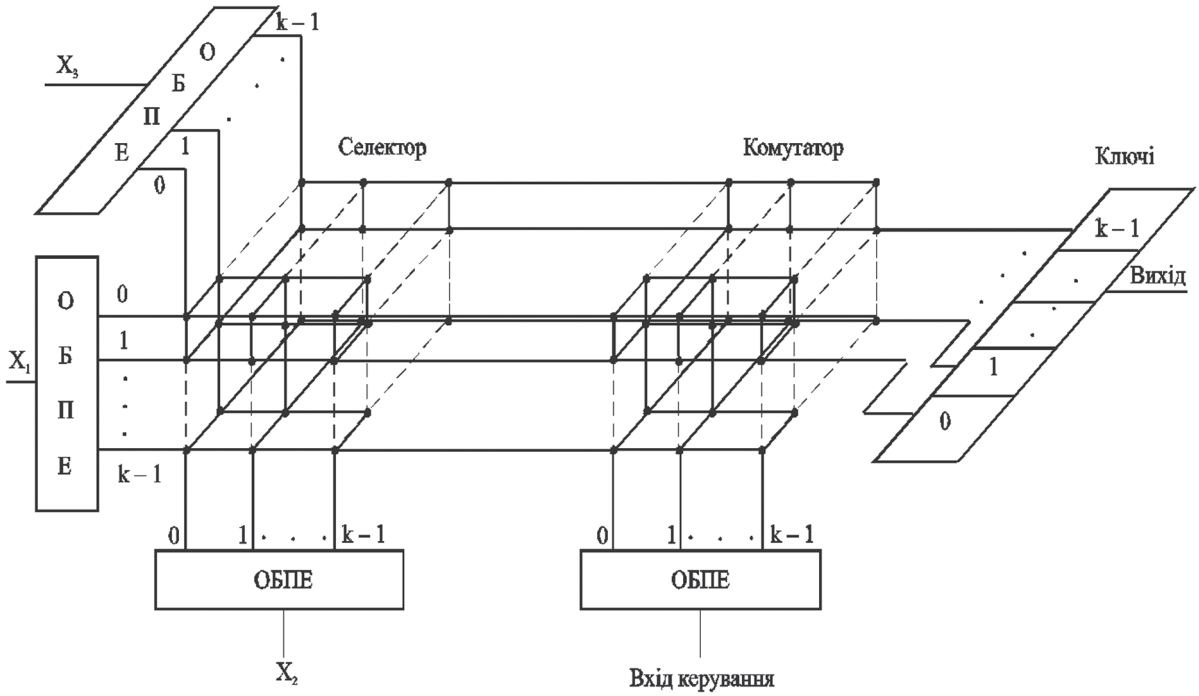


Рис.1. Узагальнена схема тривходового універсального просторового функціонального перетворювача

повісти на запитання: чому наш «повільний» мозок вирішує, наприклад, задачі розпізнавання в мільйон раз швидше, ніж найшвидші комп'ютери?

У цьому підході досліджувався механізм переходу від складної системи до простого організму, тобто була поставлена задача розробити псевдоорганізм. Для реалізації псевдоорганізму було розроблено *комп'ютерну абсолютну універсальну систему* (КАУС) [18].

Наголосимо на основних чинниках, важливих для аналізу проблеми.

Емерджентна властивість – це (якщо відштовхуватися від визначення емерджентності як властивості раптово виникати) несподіване, невідоме раніше, нова властивість. Використовуючи математичну термінологію, можна сказати, що емерджентними властивостями (рішеннями) володіють нелінійні рівняння. Тоді емерджентність можна заміряти ступенем нелінійності. Емерджентність нелінійності полягає у тому, що нелінійна модель здатна породжувати безліч якісно різних рішень. Із зростанням нелінійності кількість рішень стає настільки великою, що поведінка системи наближається до поведінки стохастичної системи. Саме це явище привело до введення поняття детермінованого хаосу. З появою у поведінці системи нелінійних ефектів порушується принцип суперпозицій. Ціле перестає бути тим, що складається з окремих елементів. Емерджентність – це одна з ознак середовища носія інтелекту, а оскільки інтелект невіддільний від середовища існування, то надалі говоритимемо, що емерджентність – це необхідна властивість інтелекту.

Іншою властивістю інтелектуальних систем і самого інтелекту є його іманентність. *Іманентне* – те, що наявне в чому-небудь, властиве чому-небудь. До цього необхідно додати – і невіддільне від цього чого-небудь.

Парадоксальність проблеми розробки ШІ полягає в наступному. З одного боку, інтелект є останнім досягненням еволюції людини як найвисокоорганізованішою матерією. Це останнє, що з'явилося в організмі в процесі його розвитку. У зв'язку з цим намагатися створити рукотворний інтелект, не маючи рукотворних простих організмів, здається безглуздим. З іншого боку, якщо цей феномен (мислення і інтелект) з'явився останнім, то він є наймолодшим в організмі і, як наслідок, його простіше змоделювати. Таким чином, проблема розробки ШІ одночасно і проста, і складна, тобто постановка самого питання про його розробку, з одного боку, виправдана, а з іншого, безглузда, неправомірна або передчасна. Цей парадокс виявляє єство механізмів, здатних забезпечити існування чогось подібного до інтелекту.

Відзначимо також, що коли математик стверджує, що світ нелінійний, він не помиляється, а просто вкладає в поняття "нелінійність" набагато більше, ніж людина, що зводить нелінійність до рівняння ступеня вище першого. Нелінійність - це перш за все перехід (зв'язок) від детермінізму до стохастики. Звідси витікає, що нелінійність - це нетривіальний, діалектичний зв'язок, це власне «життя», як перехід від народження до смерті, це процес становлення, що поглиблюється і породжує нове.

Однією з найяскравіших ознак нелінійності системи є той факт, що породжена нею реакція не міститься ні в одній з її початкових складових частин до дії. Іншими словами, нелінійна система здатна змінювати свою структуру, породжувати нові рішення. З опису КАУС відомо, що вона здатна візуалізувати власні уявлення про реальні об'єкти у вигляді реакцій (комп'ютерних образів чи структур, складених з комбінацій пікселів екрану, що знаходяться в різних станах) на збурення, джерелом яких ці об'єкти є.

Псевдоорганізм, як і будь-який організм, володіє своїми властивостями і параметрами. В нашому сприйнятті це видимі графічні образи, що синтезуються на екрані. Їх параметри і властивості можна вимірювати, розпізнавати, вивчати. Для комп'ютера, що працює за фіксованою програмою, вони є метафізикою, оскільки багато які з них просто не передбачені програмою. Для комп'ютерної системи істотно і доступне тільки те, що ми вже «заклали» – формалізували і зробили доступним для неї, перетворивши на якусь, доступну тільки їй, програму. Для математика - це лише наші уявлення, виражені за допомогою математичної символіки, які безпосередньо не можна побачити і зміряти.

Суть новітньої концепції в області розробки ШІ складають навчання і когнітивна графіка. Обидві ці складові є іманентними властивостями організму (живого) або реальної системи, розглянутої іншою реально живою системою на рівні поняття, яка сама є живим організмом. З іншого боку, якщо реалізувати дану концепцію в автономній системі, це ні що інше, як створення реального робота. Дана схема близька до відображення власне механізму функціонування живого, що природньо розвивається, еволюціонує й самоорганізовується.

Висновки

Сфери предметних галузей, де найдоцільніше працювати з даними і знаннями, поданими мовними моделями – це галузі з переважанням емпіричного знання, де складність фактів і їх описів виключає використання мови математики і, зокрема, інформаційно-інтелектуальних та лінгвістичних технологій. При цьому найбільш загальна проблема побудови системи управління семантичного чи семантико-прагматичного

рівня взаємодії пов'язана з вибором технології контекстно-залежного подання знань, побудовою інформаційних баз (даних і знань) про предметну галузь і механізму висновку для отримання необхідних рішень. Зауважимо, що єдино відомим нам, об'єктивним носієм знання та інтелекту є людина, а виразником, засобом до зовнішнього спілкування та носієм інтелекту є людська мова, що й складатиме об'єкт і напрямок досліджень у наступних частинах роботи.

Подання моделей декларативних мов здійснюється предикатами, рисунками, кресленнями, графами тощо. Математична структура даних у декларативних мовах базується на системах предикатних рівнянь в алгебрі скінченних предикатів або на аксіоматичній теорії множин, у якій теорія множин інтерпретується як структура даних. Отже, проблема створення ШІ полягає не в тому, щоб «будувати штучних людей», а в тому, щоб пізнати природні організми настільки, щоб використовувати їх на рівні систем.

Список літератури: 1. *Бондаренко, М.Ф.* Основи теорії синтезу надшвидкодіючих структур мовних систем штучного інтелекту [Текст] / М.Ф. Бондаренко, З.Д. Коноплянко, Г.Г. Четвериков. – К.: ІЗМН, 1997. – 264 с. **2.** *Коноплянко, З.Д.* Концепціїї організації інформаціойно-інтелектуальних технологій [Текст] / З.Д. Коноплянко // Бионика интеллекта. – 2008. – №2(69). – С. 164–172. **3.** *Четвериков, Г.Г.* Концепціїя уніфікації методів та засобів побудови просторових багатозначних структур мовних систем [Текст] / Г.Г. Четвериков // Бионика интеллекта. – 2010. – №1(72). – С. 3–11. **4.** *Лачинов, В.М.* Информодинамика или Путь к Миру открытых систем [Текст] / В.М. Лачинов, А.О. Поляков. – Санкт-Петербург: Издательство СПбГТУ. – 1999. – 432 с. **5.** *Бондаренко, М.Ф.* Основи теорії багатозначних структур і кодування в системах штучного інтелекту [Текст] / М.Ф. Бондаренко, З.Д. Коноплянко, Г.Г. Четвериков. – Х.: Фактор-Друк, 2003. – 336 с. **6.** *Четвериков, Г.Г.* Формалізація принципів побудови універсальних *k*-значних структур мовних систем штучного інтелекту [Текст] / Г.Г. Четвериков // Доповіді НАН України. – 2001.– №1 (41). – С. 76–79. **7.** *Никитина, Ф.А.* Моделирование языковой действительности с помощью функционально-семантических полей [Текст] / Ф.А. Никитина // Проблемы бионики. – 1988. – Вып. 41. – С. 81–85. **8.** *Терзиян, В.Я.* Формализация процесса устранения многозначности синтаксического разбора естественоязыкового высказывания. Сообщение 1 и 2 [Текст] / В.Я. Терзиян, И.И. Попков // Проблемы бионики. – 1991. – Вып.47. – С. 23-36. **9.** *Осыка, А.Ф.* О формализации семантики словосочетания с инструментальным значением [Текст] / А.Ф. Осыка, О.А. Кравец //Проблемы бионики. – 1990. – Вып.45(90). – Харьков: Основа. – С. 3 – 10. **10.** *Кольцов, В.П.* О содержательной интерпретации алгебры идей [Текст] / В.П. Кольцов, Ю.П. Шабанов-Кушнаренко // Проблемы бионики. – 1990. – Вып.45(90). – Харьков: Основа. – С. 10–17. **11.** *Бондаренко, М.Ф.* О математическом моделировании семантики производных с модификационными значениями [Текст] / М.Ф. Бондаренко, А. С. Левицкий // Проблемы бионики. – 1990. – Вып.45(90). – Харьков: Основа. – С. 17–22. **12.** *Шабанов-Кушнаренко, Ю. П.* Математические модели межморфемных связей на множестве полисемантических производящих основ и словообразовательных суффиксов [Текст] / В.И. Булкин, Д.И. Ситников, Ю. П. Шабанов-Кушнаренко // Проблемы бионики. – 1991. – Вып. 47. – Харьков: Основа. – С. 3–22. **13.** *Осыка, А.Ф.* К вопросу о формировании семантических признаков слов. [Текст] / А.Ф. Осыка, О.А. Кравец // Проблемы бионики. – 1991. – Вып. 47. – Харьков: Основа. – С. 8–16. **14.** *Рябова, Н.В.* Моделирование семантики производных слов русского языка. [Текст] / Н.В. Рябова // Проблемы бионики. – 1990. Вып.45(90). – Харьков: Основа. – С. 17–23. **15.** *Четвериков, Г.Г.* Структура элементов компьютерной лингвистики и пути ее моделирования [Текст] / Г.Г. Четвериков, Т.Н. Федорова, И.Д. Вечирская // Прикладная лингвистика и лингвистические технологи: сборник научных трудов.К.: Довіра, 2010. – С. 206–214. **16.** *Широков, В.А.* Очерк основных принципов квантовой лингвистики [Текст] / В.А. Широков // Бионика интеллекта. – 2007. – №1(66). – С. 25–32. **17.** *Бондаренко, М.Ф.* Теория интеллекта [Текст]: учеб. / М.Ф. Бондаренко Ю.П. Шабанов-Кушнаренко – Харьков: Изд-во СМИТ, 2006. – 571 с. **18.** *Васильєв, Н.Р.* Математическая модель псевдоорганизма (КАУС) – путь к созданию искусственного интеллекта [Электронный ресурс] / Н.Р. Васильєв, Е.В. Задорин – 2008. – Режим доступа: http://arupa-manas.narod.ru/BASE/nauka/index.html

Надійшла до редколегії 12.09.2011

УДК 519.71

Концепция унификации информационно-интеллектуальных технологий в языковых системах / М.Ф. Бондаренко, З.Д. Коноплянко, Г.Г. Четвериков // Бионика интеллекта: научн.-техн. журнал. – 2011. – № 3 (77). – С. 150-156.

Проведен обзор состояния проблемы моделирования функций интеллекта человека на уровне его языкового поведения; дан сравнительный синтаксический и семантический анализ морфологии словоформ. Предложен аппаратный способ реализации моделей естественного языка в виде k-значных структур (АКП-структур). Изложена суть концепции в области дальнейшего развития систем искусственного интеллекта.

Ил. 1. Библиогр.: 18 назв.

Concept of unification information intellectual technologies in language systems / M.F. Bondarenko, Z.D. Konoplyanko, G.G. Chetverikov // Bionics of Intelligense: Sci. Mag. – 2011. – № 3 (77). –P. 150-156.

The review condition of modelling functions intelligence person problem at level of its language behaviour is spent. The comparative syntactic and semantic analysis of morphology word forms is given. The hardware way realization natural language models in the form of k-unit structures (AKP-STRUCTURES) is offered. The concept matter in the field of the further development artificial intelligence systems is stated.

Fig. 1. Ref.: 18 items.

УДК 004.652: 519.765:519.767:004.93

Б.М. Павлишенко

Львівський національний університет імені Івана Франка,
Україна, pavlsh@yahoo.com

КВАНТОВИЙ АЛГОРИТМ ПОШУКУ КЛЮЧОВИХ СЛІВ У МАСИВАХ ТЕКСТОВИХ ДАНИХ

Запропонована модель квантового запису масиву текстових даних. Проаналізовано пошук ключових слів у квантовій базі даних текстових масивів, використовуючи елементи алгоритму Гровера. Показана ефективність квантового алгоритму у випадку великих розмірів текстових масивів, які суттєво перевищують розмір словника, та у випадку багатократної появи ключових слів в аналізованому текстовому масиві.

КВАНТОВІ ОБЧИСЛЕННЯ, АНАЛІЗ ТЕКСТІВ, КВАНТОВИЙ АЛГОРИТМ ГРОВЕРА

Вступ

Квантові комп'ютери та алгоритми є новим перспективним напрямком сучасних інформаційних технологій. Вони дають можливість суттєво пришвидшити розв'язок деяких класів задач внаслідок реалізації квантового паралелізму та заплутаності квантових станів. Одним із відомих квантових алгоритмів є алгоритм Гровера для пошуку даних у неструктурованій базі даних [1,2,3]. Цей алгоритм дає можливість знайти дані, які відповідають певним критеріям за час $O(\sqrt{N})$, використовуючи кількість елементів пам'яті, пропорційну $O(\log N)$, де N – кількість елементів бази даних. В порівнянні із класичним алгоритмом, в якому пошук здійснюється за час $O(N)$, алгоритм Гровера дає квадратичне прискорення розв'язку, що є актуальним при великих значеннях N. Особливістю квантового алгоритму Гровера є те, що він є імовірнісним алгоритмом, тобто дає правильний розв'язок із заданою ймовірністю, яка може бути збільшена шляхом повторного використання алгоритму. Алгоритм Гровера може бути складовою частиною інших квантових алгоритмів, зокрема, в роботі [4] він використовується для еволюційного аналізу кліткових автоматів. Інтелектуальний аналіз даних, зокрема, текстового типу є також одним із поширених напрямків інформаційних технологій. Зростаючий об'єм інформації зумовлює пошук нових комплексних рішень для інтелектуального аналізу даних. З цих позицій є перспективним розгляд можливих квантових алгоритмів аналізу слабо структурованих даних, таких як текстові масиви.

На даний час розвитку квантових комп'ютерів розроблені лише елементарні базові елементи та пристрої з квантовим регістром, який містить декілька кубітів. Тому паралельно із розвитком елементної бази є необхідність розробки систем чисельного моделювання квантових алгоритмів на класичних комп'ютерах. В даній роботі розглянемо чисельне моделювання алгоритму Гровера для реалізації квантового пошуку ключових слів у масиві текстів.

1. Постановка задачі

Розглянемо базові елементи квантових обчислень, які можуть бути використані в алгоритмах аналізу текстів. Створимо модель квантового запису масиву текстових даних. Розглянемо пошук заданих ключових слів у квантовій базі даних текстових масивів, використовуючи елементи алгоритму Гровера. Проаналізуємо ефективність квантового пошуку ключових слів у порівнянні із класичними алгоритмами.

2. Базові елементи квантових обчислень

Розглянемо базові квантові принципи реалізації квантових алгоритмів [5,6]. Основною відмінністю квантового біта - кубіта від класичного біта є те, що кубіт крім станів $|0\rangle$ і $|1\rangle$ може також знаходитись в суперпозиції цих станів

$$|\psi\rangle = a|0\rangle + b|1\rangle .$$

Стани $|0\rangle$ і $|1\rangle$ є базисними векторами, які можна представити у матричному вигляді

$$|0\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \quad |1\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix}.$$

Регістр із n кубітів $|x_1, x_2, ...x_n\rangle$ утворює суперпозицію із $N = 2^n$ станів, яку можна записати так:

$$|\psi\rangle = \sum_{i=0}^{N-1} a_i |i\rangle .$$

Ортонормований базис $\{ |0\rangle, |1\rangle, ... |2^n - 1\rangle \}$ називають обчислювальним базисом.

Розглянемо однокубітні квантові вентилі. Розрахунки в квантових алгоритмах здійснюють за допомогою унітарних перетворень, які можна розглядати як повороти комплексного векторного простору. Розглянемо базові операції над кубітами. Оператор тотожного перетворення не змінює значення кубітів і має вигляд

$$I = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}.$$

Операцію заперечення використовують для реалізації інверсії значень кубітів і визначають так:

$$X = |0\rangle\langle 1| + |1\rangle\langle 0| .$$

У спінорному зображенні оператор заперечення має вигляд матриці

$$X = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}.$$

Одним із важливих елементів є «контрольоване НЕ», яке здійснюється над двома кубітами і змінює значення другого кубіта на протилежне, якщо значення першого кубіта рівне 1. Цей логічний елемент може бути визначений як

$$U_{CNOT} = |0\rangle\langle 0| \otimes I + |1\rangle\langle 1| \otimes X ,$$

а матриця оператора унітарного перетворення «контрольоване НЕ» має вигляд

$$U_{CNOT} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}.$$

Дію вентиля «контрольоване НЕ» можна зобразити так:

$$U_{CNOT} : |a,b\rangle \to |a, a \oplus b\rangle ,$$

де $\oplus$ означає сумування за модулем 2. Ще одним важливим логічним елементом є вентиль Тоффолі, який діє на три кубіти і змінює значення третього кубіта на протилежне, якщо значення першого та другого кубітів рівне 1. Від логічного елемента «контрольоване НЕ» вентиль Тоффолі відрізняється наявністю ще одного додаткового керуючого кубіта. Цей вентиль можна визначити так:

$$T = |0\rangle\langle 0| \otimes I \otimes I + |1\rangle\langle 1| \otimes U_{CNOT} .$$

Перетворення Тоффолі можна зообразити в такому вигляді:

$$T : |a,b,c\rangle \to |a,b,c \oplus ab\rangle .$$

Вентиль Тоффолі є універсальним квантовим логічним елементом на основі якого можна побудувати оборотну квантову машину Тюрінга. В подальших дослідженнях будемо використовувати вентиль Тоффолі із *n* керуючими кубітами. В такому вентилі відбувається інверсія керованого кубіта при умові, що *n* керуючих кубітів мають значення $|1\rangle$.

3. Представлення текстових масивів у квантовій пам'яті

Нехай існує деякий текстовий масив, який можна представити у вигляді впорядкованої множини

$$T = \{ t_i | i = 1,2,...N_t \} ,$$

де N_t – кількість слів у текстовому масиві. Індекс кожного елемента цієї множини відповідає його положенню в текстовому масиві. Словник слів цього масиву розглянемо у вигляді такої впорядкованої множини:

$$W = \{ w_i | i = 1,2,...N_w \} ,$$

де N_w – кількість слів у словнику. Кожне слово w_i цього словника можна закодувати його номером *i* за допомогою відображення

$$U_w : \ w_i \to i .$$

Для простоти розгляду будемо вважати, що розмір множини текстового масиву рівний $N_t = 2^{(nt)}$, а розмір словника рівний $N_w = 2^{(nw)}$. Тоді для кодування положення слова в масиві потрібно *nt* двійкових елементів, а для кодування положення номера слова у словнику потрібно *nw* двійкових елементів. Нехай квантовий регістр складається із двох частин $|qt\rangle$ і $|qw\rangle$:

$$|qt\rangle \otimes |qw\rangle,$$
$$|qt\rangle = |q_1^t, q_1^t, ..., q_{nt}^t\rangle,$$
$$|qw\rangle = |q_1^w, q_1^w, ..., q_{nw}^w\rangle.$$

Регістр $|qt\rangle$ описує положення слова у текстовому масиві, а регістр $|qw\rangle$ описує положення слова у словнику. Введемо додатковий кубіт $|f\rangle$, який буде відображати наявність слова із заданим номером в заданій позиції текстового масиву. Тобто, якщо дане слово присутнє в заданій позиції, то значення цього кубіта рівне $|1\rangle$, в іншому випадку рівне $|0\rangle$. Тоді весь текстовий масив можна буде записати у вигляді такої системи кубітів

$$|qr\rangle = |q_1^t, q_1^t, ..., q_{nt}^t\rangle \otimes |q_1^w, q_1^w, ..., q_{nw}^w\rangle \otimes |f\rangle . \qquad (1)$$

Для запису в квантову пам'ять тексту довжиною N_t, який містить словник розміром N_w, достатньо $nt + nw + 1 = \log_2(2 \cdot N_t \cdot N_w)$ кубітів. Така експоненційна економія квантової пам'яті у порівнянні із класичною пам'яттю можлива внаслідок реалізації квантового паралелізму. Наприклад, якщо словник містить 2^{15} слів, а текстовий масив 10^{40} слів, що приблизно відповідає обсягу світової класичної літератури, то для запису такої інформації необхідно лише 15+40+1=56 кубітів. Для запису всієї відомої текстової інформації потрібно буде лише декілька сотень кубітів.

Запис тексту в квантову пам'ять розглянемо феноменологічно за допомогою квантового оракула. В теорії квантових обчислень показано, що на основі однокубітних та двокубітних квантових унітарних вентилів можна побудувати еквівалентні алгоритми класичної машини Тюрінга. Під оракулом будемо розуміти деяке формалізоване унітарне перетворення, за допомогою якого реалізуються наперед задані обчислення. Елементи текстового масиву визначаються квантовими станами складеного регістру кубітів (1). Суперпозиція цих ста-

нів утворює вектор в комплексному Гільбертовому просторі. Цей вектор є квантовим еквівалентим відображенням текстових даних. Розглянемо послідовність квантового запису тексту. Кубіт $|f\rangle$ візьмемо в початковому стані $|0\rangle$, а регістри $|qt\rangle$, $|qw\rangle$ в початкових станах $|0\rangle^{\otimes(nt)}$ та $|0\rangle^{\otimes(nw)}$ відповідно. Застосуємо однокубітні унітарні перетворення Адамара до регістрів $|qt\rangle\otimes|qw\rangle\otimes|f\rangle$:

$$|\psi\rangle = H^{\otimes(nt)} \otimes H^{\otimes(nw)} \otimes I\left(|qt\rangle\otimes|qw\rangle\otimes|f\rangle\right).$$

В результаті отримаємо

$$|\psi\rangle = \frac{1}{\sqrt{2^{nt+nw}}} \sum_{i=0,j=0}^{N_t,N_w} |i\rangle\otimes|j\rangle\otimes|0\rangle. \qquad (2)$$

Суперпозиція $|\psi\rangle$ містить базисні ортогональні стани, кожен з яких відповідає одному положенню визначеного слова у тексті. В процесі вимірювання відбувається редукція суперпозиції до одного базового стану, який відповідає одному слову в деякому положенні. Отже, та кількість памяті, яка в класичному випадку необхідна для запису одного слова, у квантовому випадку є достатньою для запису цілого текстового масиву. Нехай наявність заданого слова у певному місці тексту описується функцією $f_q(qt,qw)$, де індекс qt описує положення слова у тексті, а індекс qw описує положення цього слова у словнику. У випадку наявності заданого слова у заданому місці тексту функція набуває значення 1, інакше 0. Процес запису текста у квантову пам'ять опишемо унітарним перетворенням U_F, яке визначається квантовим оракулом

$$U_F : |qt\rangle\otimes|qw\rangle\otimes|f\rangle \rightarrow$$
$$\rightarrow |qt\rangle\otimes|qw\rangle\otimes\left|f \oplus f_q(qt,qw)\right\rangle,$$

де $\oplus$ означає сумування за модулем 2. Враховуючи, що кубіт $|f\rangle$ є в початковому стані $|0\rangle$, отримаємо

$$U_F : |qt\rangle\otimes|qw\rangle\otimes|0\rangle \rightarrow |qt\rangle\otimes|qw\rangle\otimes\left|f_q(qt,qw)\right\rangle. \quad (3)$$

4. Пошук заданого ключового слова у квантовій базі даних

Завдання полягає в пошуку деякого ключового слова w_k, яке може бути закодоване у вигляді квантового стану

$$|qk\rangle = \left|q_1^k, q_1^k, ..., q_{nw}^k\right\rangle.$$

Використаємо вентиль Тоффолі для знаходження квантових станів, в яких закодовані ключові теги. Введемо в систему кубітів (1) додатковий кубіт – анцилу $|z\rangle$, отримаємо

$$|qr\rangle = |qt\rangle\otimes|qw\rangle\otimes\left|f_q(qt,qw)\right\rangle\otimes|z\rangle. \qquad (4)$$

Подіємо на кубіт $|z\rangle$ в стані $|1\rangle$ оператором Адамара

$$|z\rangle = H|1\rangle = \frac{1}{\sqrt{2}}\left(|0\rangle - |1\rangle\right). \qquad (5)$$

Допоміжний кубіт $|z\rangle$ буде керуватись за допомогою $nw+1$-мірного елемента Тоффолі T_{nw+1}, в якому керуючими кубітами виступають nw кубітів регістру $|qw\rangle$ та кубіт $|f(qt,qw)\rangle$. Значення кубіта $|z\rangle$ в квантовому стані може змінитись під впливом вентиля Тоффолі у випадку, коли всі кубіти регістру $|qw\rangle$ та кубіт $|f(qt,qw)\rangle$ будуть рівні одиниці. Щоб перевести значення кубітів у значення $|1\rangle$ для квантових станів, які описують ключове слово w_i, розглянемо унітарний оператор, який є тензорним добутком однокубітних операторів

$$S_T = I^{\otimes(nt)} \otimes \left(\bigotimes_{i=1}^{nw} S_i\right) \otimes I \otimes I, \qquad (6)$$

де $S_i = \begin{cases} I, q_i^k = 1; \\ X, q_i^k = 0; \end{cases}$

Оператор S_T переводить квантові базисні стани регістру $|qr\rangle$ у стани, в яких значення регістру $|qw\rangle$ рівні 1, якщо співпадають кодування слова в тексті та ключового слова $|qw\rangle = |qk\rangle$, тобто

$$S_T|qr\rangle = |qt\rangle\otimes|1,1,...1\rangle_{nw}\otimes|f(qt,qw)\rangle\otimes|z\rangle,$$

якщо $|qw\rangle = |qk\rangle$.

Розглянемо оператор

$$U_T = (S_T)\cdot(I^{\otimes(nt)} \otimes T_{nw+1})\cdot(S_T). \qquad (7)$$

Подіємо цим оператором на систему регістрів кубітів $|qr\rangle$. Перша справа група операторів виділених дужками переводить регістр $|qw\rangle$ у значення $|1,1,...1\rangle_{nw}$ для станів, які відповідають шуканим ключовим словам, друга група операторів реалізує інверсію керованого кубіта $|z\rangle$ для шуканих станів, третя група повертає змінені першою групою стани у стан перед застосуванням оператора U_T. В результаті дії оператора U_T отримаємо

$$|\psi\rangle_T = U_T\left(\frac{1}{\sqrt{2^{nt+nw}}}\sum_{x=1}^{x=2^{nt+nw}} |x\rangle\otimes|f(x)\rangle\otimes|z\rangle\right) =$$
$$= \frac{1}{\sqrt{2^{nt+nw}}}\times\left(\sum_{x\notin X_k}|x\rangle\otimes|f(x)\rangle - \sum_{x\in X_k}|x\rangle\otimes|f(x)\rangle\right)\otimes|z\rangle, \quad (8)$$
$$|x\rangle = |qt\rangle\otimes|qw\rangle,$$

де X_k – множина станів суперпозиції, які відповідають кодуванню шуканих ключових слів. Допоміжний керований кубіт $|z\rangle$ не змінив свого значення під дією оператора U_T і знаходиться у стані (5), в якому він знаходився перед дією оператора U_T, тому його можна винести за дужки та вилучити із подальшого розгляду. Це зумовлено тим, що кубіт $|z\rangle$ був переведений в новий базисний стан (5) за допомогою оператора Адамара. Інверсія стану в цьому базисі є рівнозначна інверсії знаку амплітуди підсистеми квантових станів, в яких закодовані ключові слова. Роль цього кубіта полягала у тому, щоб під дією вентиля Тофолі він змінював свій знак на протилежний у квантових станах, які відповідають умові пошуку.

Умова рівності регістрів ключового тега та регістру кодування слова враховується в операторі S_T.

Наступним кроком алгоритму є переведення додаткового кубіта $|f\rangle$ у початковий базисний стан $|0\rangle$ та вилучення його із подальшого розгляду. Це можна зробити за допомогою унітарного оператора U_F^{-1}, який є оберненим до оператора U_F (3). В результаті отримаємо

$$|\psi\rangle_{F^{-1}} = U_F^{-1}|\psi\rangle_T = \frac{1}{\sqrt{2^{nt+nw}}}\left(\sum_{x\notin X_k}|x\rangle - \sum_{x\in X_k}|x\rangle\right),$$
$$|x\rangle = |qt\rangle\otimes|qw\rangle.$$

Отже, в результаті дії складеного оператора $U_F^{-1}U_TU_F$ отримаємо суперпозицію базисних рівноймовірнісних станів (2), в якій стани, що кодують наявні ключові слова, мають від'ємну амплітуду.

5. Підсилення амплітуд заданих квантових станів

Для того щоб при вимірюванні виявити квантові стани, в яких закодовані ключові слова, необхідно підсилити амплітуди шуканих станів. Таке підсилення амплітуд станів, які мають задані властивості, можна здійснити за допомогою оператора інверсії відносно середнього, який використовується в алгоритмі Гровера для пошуку в неструктурованій квантовій базі даних [1,2,3]. Оператор інверсії розглянемо у вигляді

$$U_G = 2|\psi_c\rangle\langle\psi_c| - I\ , \qquad (9)$$

де $|\psi_c\rangle = H^{\otimes n}|0\rangle^{\otimes n} = \frac{1}{\sqrt{2^n}}\sum_{i=0}^{i=2^n-1}|i\rangle$.

В геометричній інтерпретації оператор U_G здійснює в Гільбертовому просторі дзеркальне відображення деякого вектора відносно осі, яка визначається вектором $|\psi_c\rangle$. Оператор інверсії можна представити сукупністю однокубітних операторів Адамара та операторів інверсії стану кубіта відносно базисного вектора $|0\rangle$

$$U_G = H^{\otimes n}(2|0\rangle\langle 0| - I)^{\otimes n}H^{\otimes n}.$$

Оператор U_G також називають оператором інверсії відносно середнього [1, 2, 3, 6], оскільки перетворення U_G можна зообразити

$$U_G : \sum_i a_i|i\rangle \to \sum_i (2A - a_i)|i\rangle\ ,$$

де A – середнє значення амплітуд a_i. Оператор U_G можна зообразити унітарною матрицею, елементи якої визначаються правилом

$$S_{ij} = \begin{cases} \frac{2}{N}, i \neq j \\ \frac{2}{N} - 1, i = j. \end{cases}$$

Підсилення амплітуд станів із інверсними знаками амплітуд відбувається внаслідок дії оператора інверсії U_G аналогічно до механізму, описаного в алгоритмі Гровера [1, 2, 3]. Враховуючи визначення операторів (3), (6), (7), (9), розглянемо ітерацію алгоритму Гровера у загальному вигляді складеного оператора

$$U_I = U_G U_F^{-1} U_T U_F\ . \qquad (10)$$

Можна показати, що внаслідок реалізації ітерації U_I можна підсилити амплітуди заданих станів в 3 рази. Якщо шукане ключове слово зустрічається в текстовому масиві лише один раз, то аналогічно до алгоритму Гровера [1,2,3] оптимальна кількість ітерацій U_I буде

$$k_U \approx \frac{\pi}{4}\sqrt{N}, N = 2^{nt+nw},$$

де N – кількість квантових станів рівна $N = N_t N_w$. Отже, складість алгоритму в даному випадку буде $O(\sqrt{N})$. Якщо кількість шуканих ключових тегів рівна l, тоді згідно із [1,2,3,6] кількість необхідних ітерацій буде рівна

$$k_U \approx \frac{\pi}{4}\sqrt{\frac{N}{l}}\ .$$

Однак наперед невідомо, яка кількість входжень ключового слова в текстовому масиві. В такому випадку можна провести серію реалізацій алгоритму Гровера із кількостями ітерацій U_I, які утворюють прогресію

$$k_U = 1,2,4,8,...l_U\ ,$$

де l_U – деяке максимальне значення кількості ітерацій U_I. Тобто спочатку реалізується алгоритм з однією ітерацією, потім із двома і т.д. Якщо в цій серії реалізацій алгоритму при вимірюванні виявлено шукані квантові стани, тоді можна прийняти рішення про наявність шуканого ключового слова в аналізованому текстовому масиві, а також оцінити кількість появи цього слова в текстовому масиві. Можна показати, що складність алгоритму в такому випадку є також $O(\sqrt{N})$. В класичному алгоритмі складність пошуку в неструктурованому текстовому масиві буде $O(N_t)$. Враховуючи те, що $N = N_t N_w$, можна побачити, що не для всіх випадків квантовий алгоритм дасть поліноміальне прискорення обчислень. Наприклад, коли $N_t \approx N_w$, складність пошуку класичного та квантового алгоритму буде співмірною. Поліноміальне прискорення алгоритму буде спостерігатись для випадку, коли розмір текстового масиву суттєво перевищує розмір словника, тобто $N_t >> N_w$.

Висновки

В роботі розглянута модель запису текстових масивів у квантову пам'ять. Запропонований квантовий алгоритм пошуку ключових слів у текстових масивах. Реалізація цього алгоритму здійснюється на основі квантових логічних елементів, зокрема,

з використанням вентиля Тоффолі. Ітерація Гровера використовується для підсилення амплітуд квантових станів із заданими властивостями. Показано, що реалізація квантового алгоритму пошуку ключових слів у текстових масивах за деяких умов дає можливість експоненційно скоротити об'єм необхідної пам'яті та поліноміально зменшити час виконання алгоритму у порівнянні із класичними алгоритмами внаслідок реалізації квантового паралелізму. Ефективність квантового алгоритму зростає у випадку великих розмірів текстових масивів, які суттєво перевищують розмір словника, та у випадках багатократної появи ключових слів в аналізованому текстовому масиві.

Список літератури. 1. *Grover, L.K.* Quantum Mechanics helps in searching for a needle in haystack/ L.K.Grover // Phys.Rev. Lett. 79(2):325–328, 1997. **2.** *Zalka, C.* Grover's quantum searching algorithm is optimal./ C. Zalka. // *Phys. Rev. A.*, 60(4):2746–2751, 1999. **3.** *Lavor, C., Manssur, L.R.U., Portugal, R.* Grover's Algorithm: Quantum Database Search [Електронний ресурс] // C.Lavor, L.R.U. Manssur, R. Portugal // arXiv:quant-ph/0301079v1 (2003). **4.** *Pavlyshenko, B.* Quantum Algorithm of Evolutionary Analysis of 1D Cellular Automata [Електронний ресурс] // B. Pavlyshenko // arXiv:1001.4870v1 (2010). **5.** *Крохмальський, Т.* Квантові комп'ютери: основи й алгоритми (короткий огляд) [Текст]/ Т.Крохмальський // Журнал фізичних досліджень. – 2004. – Т. 8. – № 4. – С. 1-15. **6.** *Китаев, А.* Классические и квантовые вычисления [Текст] / А. Китаев, А. Шень, М. Вялий – М.: МЦНМО, ЧеРо, 1999. – 192 с.

Надійшла до редколегії 26.09.2011

УДК 004.652: 519.765:519.767:004.93

Квантовый алгоритм поиска ключевых слов в текстовых массивах данных / Б.М. Павлышенко // // Бионика интеллекта: науч.-техн. журнал. – 2011. – № 3 (77). – С. 157-161.

Предложена модель квантовой записи массива текстовых данных. Проанализирован поиск ключевых слов в квантовой базе данных текстовых массивов используя елементы алгоритма Гровера. Показана эффективность квантового алгоритма в случаэ больших размеров текстовых массивов, которые существенно превышают размер словаря и в случае многократного появления ключевых слов в анализированном массиве.

Библиогр.: 6 назв.

УДК 004.652: 519.765:519.767:004.93

The Quantum Algorithm of Key Words Search in Text Arrays / B.M. Pavlyshenko // // Bionics of Intelligense: Sci. Mag. – 2011. – № 3 (77). –P. 157-161.

The model of text array quantum writing has been suggested. The search of key words in quantum database of text arrays with the use of Grover's algorithm elements has been analyzed. The effectiveness of quantum algorithm in case of large text arrays when their size exceeds essentially the dictionary size and in cases when key words appear repeateadly in the analyzed text array is shown.

Ref.: 6 items.

ОБ АВТОРАХ

Алексенко Виктория Сергеевна	74	студентка факультета компьютерных наук Харьковского национального университета радиоэлектроники
Аксак Наталия Георгиевна	94	канд. техн. наук, старший научный сотрудник, доцент кафедры ЭВМ Харьковского национального университета радиоэлектроники
Бабий Андрей Степанович	89	преподаватель кафедры информатики и информационных технологий в деяльности ОВС Харьковского национального университета внутренних дел
Барыло Екатерина Васильевна	107	аспирант секции информатики кафедры компьютерных наук Сумского государственного университета
Беспалов Юрий Гаврилович	54	старший научный сотрудник лаборатории моделирования адаптационных механизмов биологического факультета Харьковского национального университета им. В. Н. Каразина
Бенаддия Абдельлатиф	112	аспирант кафедры ЭВМ Харьковского национального университета радиоэлектроники
Бондаренко Михаил Федорович	3, 14, 30, 150	член-корреспондент НАН Украины, д-р техн. наук, профессор, ректор Харьковского национального университета радиоэлектроники
Бондаренко Татьяна Петровна	54	д-р биолог. наук, профессор, зав. отделом Института криобиологии и криомедицины Национальной академии наук Украины
Голян Наталья Викторовна	46	ассистент кафедры программного обеспечения ЭВМ Харьковского национального университета радиоэлектроники
Городнянский Илья Дмитриевич	54	студент биологического факультета Харьковского национального университета им. В. Н. Каразина
Гороховатский Владимир Алексеевич	85	д-р техн. наук, заведующий кафедрой информационных технологий Харьковского института банковского дела Университета банковского дела Национального Банка Украины
Гофман Евгений Александрович	126	аспирант кафедры программных средств Запорожского национального технического университета
Губаренко Евгений Витальевич	60	аспирант кафедры системотехники Харьковского национального университета радиоэлектроники
Гурина Анна Владимировна	70	студентка кафедры системотехники Харьковского национального университета радиоэлектроники
Джулгам Саад	98	аспирант кафедры компьютерных наук Сумского государственного университета
Ерохин Андрей Леонидович	89	д-р техн. наук, профессор, начальник факультета психологии, менеджмента, социальных и информационных технологий Харьковского национального университета внутренних дел
Жолткевич Григорий Николаевич	54	канд. физ.-мат. наук, д-р техн. наук, декан механико-математического факультета Харьковского национального университета им. В. Н. Каразина
Зайцев Сергей Алексеевич	131	аспирант кафедры программных средств Запорожского национального технического университета

Зарецкая Ирина Тимофеевна	54	канд. физ.-мат. наук, доцент механико-математического факультета Харьковского национального университета им. В. Н. Каразина.
Зацеркляный Николай Милентиевич	89	д-р техн. наук, профессор кафедры информатики и информационных технологий в деятельности ОВС Харьковского национального университета внутренних дел
Калиновская Екатерина Михайловна	54	заместитель заведующего отделом рептилий Харьковского зоопарка
Калита Надежда Ивановна	70	канд. техн. наук, доцент кафедры системотехники Харьковского национального университета радиоэлектроники
Карреро Яфрейзи	54	бакалавр гуманитарных наук (BA), математика, Fine Arts minor, College of Mount Saint Vincent, Riverdale, New York; магистр ландшафтной архитектуры (MLA, Master of Landscape Architecture), The Bernard and Anne Spitzer School of Architecture, CUNY City College of New York, New York
Коноплянко Зеновий Дмитриевич	150	д-р техн. наук, профессор кафедры компьютерных технологий Львовского института банковского дела Университета банковского дела Национального Банка Украины
Кораблев Николай Михайлович	102	канд. техн. наук, доцент кафедры ЭВМ Харьковского национального университета радиоэлектроники
Кузьменко Сергей Викторович	65	канд. техн. наук, доцент Харьковского национального университета внутренних дел
Мальков Юрий Анатольевич	136	аспирант кафедры ЭВМ Харьковского национального университета радиоэлектроники
Мартыненко Сергей Сергеевич	98	ведущий специалист кафедры компьютерных наук Сумского государственного университета
Михаль Олег Филиппович	112, 119	д-р техн. наук, профессор кафедры ЭВМ Харьковского национального университета радиоэлектроники
Мохамад Али	119	аспирант кафедры ЭВМ Харьковского национального университета радиоэлектроники
Носов Константин Валентинович	54	канд. физ.-мат. наук, научный сотрудник лаборатории моделирования адаптационных механизмов биологического факультета Харьковского национального университета им. В. Н. Каразина
Олейник Андрей Александрович	126	канд. техн. наук, доцент кафедры программных средств Запорожского национального технического университета
Павлишенко Богдан Михайлович	157	канд. физ.-мат. наук, доцент факультета электроники Львовского национального университета им. Ивана Франка
Петров Эдуард Георгиевич	60	д-р техн. наук, профессор, заведующий кафедрой системотехники Харьковского национального университета радиоэлектроники
Писклакова Валентина Петровна	78	канд. техн. наук, старший научный сотрудник кафедры системотехники Харьковского национального университета радиоэлектроники
Писклакова Ольга Александровна	74, 78	канд. техн. наук, доцент кафедры системотехники Харьковского национального университета радиоэлектроники

Полякова Татьяна Викторовна	85	ассистент кафедры информатики Харьковского национального автодорожного университета
Почанский Олег Михайлович	143	аспирант кафедры искусственного интеллекта Харьковского национального университета радиоэлектроники
Пряничникова Алла Анатольевна	78	аспирант кафедры системотехники Харьковского национального университета радиоэлектроники
Русакова Наталья Евгеньевна	50	младший научный сотрудник кафедры программного обеспечения ЭВМ Харьковского национального университета радиоэлектроники
Субботин Сергей Александрович	126, 131	канд. техн. наук, доцент кафедры программных средств Запорожского национального технического университета
Танянский Сергей Станиславович	136	канд. техн. наук, доцент кафедры ЭВМ Харьковского национального университета радиоэлектроники
Турута Алексей Петрович	89	преподаватель кафедры информатики и информационных технологий в деяльности ОВС Харьковского национального университета внутренних дел
Фомичев Александр Александрович	102	аспирант кафедры ЭВМ Харьковского национального университета радиоэлектроники
Четвериков Григорий Григорьевич	150	д-р техн. наук, профессор кафедры программного обеспечения ЭВМ Харьковского национального университета радиоэлектроники
Шабанов-Кушнаренко Юрий Петрович	3, 14, 30, 46	д-р техн. наук, профессор кафедры программного обеспечения ЭВМ Харьковского национального университета радиоэлектроники
Шкловец Артем Вадимович	94	аспирант кафедры ЭВМ Харьковского национального университета радиоэлектроники
Шмидт Дмитрий Евгеньевич	74	студент факультета компьютерных наук Харьковского национального университета радиоэлектроники

ПРАВИЛА
оформлення рукописів для авторів науково-технічного журналу «БІОНІКА ІНТЕЛЕКТУ»

Науково-технічний журнал «Біоніка интелекту» приймає до друку написані спеціально для нього оригінальні рукописи, які раніше ніде не друкувались. Структура рукопису повинна бути такою: індекс УДК, заголовок, відомості про авторів, анотація, ключові слова, вступ, основний текст статті, висновки, список використаної літератури.

Відповідно до Постанови ВАК України від 15.01.2003 №7-05/1 (Бюлетень ВАК, №1, 2003, с. 2), стаття повинна мати такі необхідні елементи: постановка проблеми в загальному вигляді та її зв'язок із важливими науковими чи практичними завданнями; аналіз останніх досліджень і публікацій і виділення не вирішених раніше частин загальної проблеми в даній області; формулювання цілей та завдань дослідження; виклад основного матеріалу досліджень з повним обґрунтуванням отриманих наукових результатів; висновки з даного дослідження та перспективи подальших досліджень у даному напрямку.

Статті мають бути виконані в редакторі Microsoft Word. Формат сторінки – А4 (210х297 мм), поля: верхнє – 25 мм, нижнє – 20 мм, ліве, праве – 17 мм. Кількість колонок – 2, з інтервалом між ними 5 мм, основний шрифт Times New Roman, кегль основного тексту – 10 пунктів, міжрядковий інтервал – множник (1,1), абзацний відступ – 6 мм. Обсяг рукопису – від 4 до 12 сторінок (мови: російська, українська, англійська).

УДК друкується з першого рядка, без відступів, вирівнювання по лівому краю.

Назва статті друкується прописними літерами після відомостей про авторів; шрифт прямий, напівжирний, кегль 12. *Назви розділів* нумерують арабськими цифрами, виділяють жирним шрифтом. Відступи для назви статті, ініціалів та прізвищ авторів, відомостей про авторів, назв розділів, вступу та висновків, списку літератури: зверху – 6 пт, знизу – 3 пт.

Анотацію (мовою статті, абзац 4-10 рядків, кегль 9) розміщують на початку статті, в ній має бути розміщена інформація про результати описаних досліджень.

Ключові слова (4-10 слів з тексту статті, які з точки зору інформаційного пошуку несуть змістовне навантаження) наводять мовою рукопису, через кому в називному відмінку, кегль 9.

Малюнки та *таблиці* (чорно-білі, контрастні) розміщуються у тексті після першого посилання у вигляді окремих об'єктів і нумерують арабськими цифрами наскрізною нумерацією за наявності більше ніж одного об'єкта. Невеликі схеми, що складаються з 3-4 елементів виконують, використовуючи вставку об'єкта Рисунок Microsoft Word. Більш складні виконують у графічних редакторах у вигляді чорно-білих графічних файлів форматів .tiff, .jpg, .wmf, .cdr із розділенням 300 dpi. Рисунки мають міститися у текстовому файлі й обов'язково подаватися окремим файлом з відповідною назвою (наприклад, Рис.1.cdr).

Усі елементи малюнка, включаючи написи, повинні бути згруповані. Усі написи в малюнках і таблицях мають бути виконані шрифтом Times New Roman, кегль у малюнках – 10, у таблицях – 9.

Малюнок повинен мати центрований підпис (поза малюнком), шрифт 9, відступи зверху і знизу по 6 пт. Ширина малюнка має відповідати ширині колонки (або ширині сторінки).

Формули, символи, змінні, повинні бути набрані в редакторі формул MathType або Microsoft Equation. Формули розміщують посередині рядка й нумерують за наявності посилань на них у рукописі. Шрифт – Times New Roman. Висота змінної – 10 пунктів, великих і малих індексів – 8 пт, основний математичний символ – 12 (10) пт. Змінні, позначені латинськими літерами, набирають курсивом, грецькі літери, скорочення російських слів і цифри – прямим написанням. Змінні, які є в тексті, також набирають у редакторі формул.

Список літератури вміщує опубліковані джерела, на які є посилання в тексті, укладені у квадратні дужки, друкують без абзацного відступу, кегль 9 пт, відступ зверху – 6 пт.

Після списку літератури з відступом зверху 6 пт зазначають дату подання статті до редколегії. Число та місяць задають двозначними числами через крапку. Розмір шрифта – 9 пт, курсив, вирівнювання по правому краю.

Реферати (Times New Roman, кегль – 9 пунктів, 3-4 речення) подають російською та англійською мовами. Реферат не повинен дублювати текст анотації.

Разом із рукописом (на аркушах білого паперу формату А4 щільністю 80-90 г/м2, надрукований на лазерному принтері, у 2-х примірниках) необхідно подати такі документи:

1. Заяву, яку повинні підписати всі автори.
2. Акт експертизи про можливість опублікування матеріалів у відкритому друці.
3. Рецензію, підписану доктором наук.
4. Відомості про авторів.
5. Електронний варіант рукопису, реферату та відомостей про авторів.
6. Оплату за публікацію.

Необхідно також зазначити один з наступних тематичних розділів, якому відповідає рукопис:

1. Теоретичні основи інформатики та кібернетики. Теорія інтелекту
2. Математичне моделювання. Системний аналіз. Прийняття рішень
3. Інтелектуальна обробка інформації. Розпізнавання образів
4. Інформаційні технології та програмно-технічні комплекси
5. Структурна, прикладна та математична лінгвістика
6. Дискусійні повідомлення

БІОНІКА ІНТЕЛЕКТУ

інформація, мова, інтелект

Науково-технічний журнал

№ 3 (77)

2011

Рекомендовано Вченою радою
Харківського національного університету радіоелектроніки
(протокол № 5 від 28.10.2011)

Коректор — *Л. М. Денісова*

Комп'ютерне верстання — *О. Б. Ісаєва*

Свідоцтво про державну реєстрацію КВ № 12072-943 ПР від 07.12.2006

Підп. до друку 28.10.2011. Формат 60 x 84 $^{1}/_{8}$. Друк ризографічний.
Папір офсетний. Гарнітура Newton. Умов. друк. арк. 19,3. Обл.-вид. арк. 19,0.
Тираж 100 пр. Зам. № .

Надруковано в навчально-науковому видавничо-поліграфічному центрі ХНУРЕ

Свідоцтво суб'єкта видавничої справи ДК № 1409 від 26.06.2003

Адреса редакції та видавця:

просп. Леніна, 14, Харків-166, 61166, Україна,
Харківський національний університет радіоелектроніки, к. 127
тел. 702-14-77, факс 702-10-13,
e-mail: bionics@kture.kharkov.ua