МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ УКРАИНЫ

ХАРЬКОВСКИЙ НАЦИОНАЛЬНЫЙ
УНИВЕРСИТЕТ РАДИОЭЛЕКТРОНИКИ

ISSN 0555-2656

БИОНИКА ИНТЕЛЛЕКТА

информация, язык, интеллект

научно-технический журнал

Основан в 1968 г.

№ 1 (61), декабрь 2004

© *Харьковский национальный университет радиоэлектроники, 2004*

БІБЛІОТЕКА
ХНУРЕ

Свидетельство о государственной регистрации КВ № 9148 от 10.09.2004

Журнал включен в список специальных изданий ВАК Украины по техническим наукам

СОДЕРЖАНИЕ

Бондаренко М.Ф., Шабанов-Кушнаренко Ю.П. О БИОНИКЕ ИНТЕЛЛЕКТА 3

Бондаренко М.Ф., Шабанов-Кушнаренко Ю.П. ОБ АЛГЕБРЕ ПРЕДИКАТОВ 15

Бондаренко М.Ф., Чикина В.А., Шабанов-Кушнаренко Ю.П. МОДЕЛИ ЯЗЫКА 27

Вечирская И.Д., Дударь З.В., Иванилов А.А., Лещинский В.А. ЛИНЕЙНЫЕ ЛОГИЧЕСКИЕ ОПЕРАТОРЫ В ВИДЕ СХЕМ И ГРАФОВ 38

Четвериков Г.Г. АЛГЕБРО-ЛОГІЧНІ ЗАСОБИ МОДЕЛЮВАННЯ ПРИРОДНОІ МОВИ (ПОВІДОМЛЕННЯ 1) 42

Руденко О.Г., Сотников П.О. ОБ ОДНОМ АЛГОРИТМЕ СИНТЕЗА ПЕРСПЕКТИВНЫХ ТРАНСФОРМАЦИЙ ТЕЛЕВИЗИОННЫХ ИЗОБРАЖЕНИЙ НА ОСНОВЕ АФИННЫХ ПРЕОБРАЗОВАНИЙ 46

Бессонов А.А., Бобух В.А., Руденко О.Г. ПРИМЕНЕНИЕ НЕЙРОННОЙ СЕТИ СМАС ПРИ РЕШЕНИИ ЗАДАЧ ОБРАБОТКИ ИЗОБРАЖЕНИЙ И ЕЕ АППАРАТНАЯ РЕАЛИЗАЦИЯ 49

Киричук Е.П., Сироджа И.Б. ИНТЕЛЛЕКТУАЛЬНАЯ СИСТЕМА ПОДДЕРЖКИ ПРИНЯТИЯ ТЕХНОЛОГИЧЕСКИХ РЕШЕНИЙ ПРИ ВЫБОРЕ ОБОРУДОВАНИЯ В ТЕХНОЛОГИЧЕСКОЙ ПОДГОТОВКЕ АВИАЦИОННОГО ПРОИЗВОДСТВА 56

Ваде Гриби, Елисеев В.В., Хассан Ктейман, Хаханов В.И., Хаханова И.В. АНАЛИЗ НЕИСПРАВНОСТЕЙ ПРИ ПРОЕКТИРОВАНИИ ДИСКРЕТНЫХ ЛОГИЧЕСКИХ СЕТЕЙ 61

Ерохин А.Л., Полунин Р.А. ОБ ИНТЕЛЛЕКТУАЛИЗАЦИИ ПРОЦЕССА ПОДДЕРЖКИ ЖИЗНЕННОГО ЦИКЛА ПРОГРАММНЫХ СИСТЕМ 68

Бурцев В.Н., Ерохин А.Л. ФОРМАЛИЗАЦИЯ СЛОЖНООРГАНИЗОВАННЫХ СИСТЕМ И РАСПОЗНАВАНИЕ АВАРИЙНЫХ СИТУАЦИЙ. СООБЩЕНИЕ 1 74

Горшков Е.В., Колодяжный В.В., Плисс И.П. АДАПТИВНЫЙ АЛГОРИТМ ОБУЧЕНИЯ НЕО-ФАЗЗИ НЕЙРОНА И НЕЙРО-ФАЗЗИ СЕТИ НА ОСНОВЕ ПОЛИНОМИАЛЬНЫХ ФУНКЦИЙ ПРИНАДЛЕЖНОСТИ 78

Безрук В.М., Коваленко Н.П., Лысенко В.А. ОБ ОДНОМ МЕТОДЕ АВТОМАТИЗИРОВАННОГО РАСПОЗНАВАНИЯ СТАДИЙ СНА ПО ЭЛЕКТРОЭНЦЕФАЛОГРАММАМ 82

Петров К.Э. ПРОБЛЕМЫ ФОРМАЛИЗАЦИИ ИНТЕЛЛЕКТУАЛЬНОЙ ДЕЯТЕЛЬНОСТИ ЧЕЛОВЕКА 86

Гребенник И.В. МОДЕЛИ ОПТИМИЗАЦИИ НА КОМПОЗИЦИОННЫХ ОБРАЗАХ КОМБИНАТОРНЫХ МНОЖЕСТВ В СИСТЕМАХ ПОДДЕРЖКИ ПРИНЯТИЯ РЕШЕНИЙ 90

Альхалайбех Айман, Михаль О.Ф. МОДЕЛИРОВАНИЕ НА СЕТЯХ ПЕТРИ ИНФОРМАЦИОННОГО ОБМЕНА МЕЖДУ СИСТЕМАМИ С ОГРАНИЧЕННЫМ РЕСУРСОМ ГОТОВНОСТИ 98

Тесленко Н.А. АЛГОРИТМ ОБУЧЕНИЯ АВТОАССОЦИАТИВНОЙ ИСКУССТВЕННОЙ МНОГОСЛОЙНОЙ НЕЙРОННОЙ СЕТИ 103

Ляшенко С.А. ОБ ОДНОМ АЛГОРИТМЕ ОБУЧЕНИЯ НЕЙРОСЕТЕВОЙ МОДЕЛИ НЕЛИНЕЙНОГО ДИНАМИЧЕСКОГО ОБЪЕКТА 107

Шекета В.І. ПОБУДОВА МОДЕЛЕЙ МОДИФІКАЦІЙНИХ ПРЕДИКАТНИХ ЗАПИТІВ НА ОСНОВІ СИНТАКСИЧНОЇ КАТЕГОРІЙНОЇ СТРАТЕГІЇ 111

УДК 519.7

М.Ф. Бондаренко, Ю.П. Шабанов-Кушнаренко

О БИОНИКЕ ИНТЕЛЛЕКТА

«Искусственный интеллект как область науки – это лишь малая часть грандиозной попытки постичь мышление. Мы считаем, что это основная цель данной области науки и здесь достигнуты немалые успехи. Программы, которые мы пишем, важны как эксперимент, а не как конечный результат. Главный интерес для нас составляет именно интеллект, а не его искусственное происхождение. Если мы достигнем успеха в этом направлении, то проложим путь для создания механических помощников человеку в его повседневных делах и заботах. Но не в этом главное. Самое важное, чего мы тогда добьемся, – более глубокого понимания самих себя, что, безусловно, гораздо ценнее, чем любая программа».

Роджер Шенк [1, с.26].

В статье дана общая характеристика бионики интеллекта как перспективного научного направления, ее предмета исследования, языка и методов формального описания, перспектив дальнейшего развития и области практического использования.

1. Анализ интеллекта

Изобретение в конце XVII столетия паровой машины дало начало *первой научно-технической революции,* в результате которой достигнуто многократное усиление физических возможностей человека за счет использования механизмов и машин. Примерами достижений первой научно-технической революции могут служить: паровая машина, паровоз, электрическая машина, атомная электростанция, космический корабль, самолет, автомобиль, телескоп, станки. В середине XX столетия были созданы универсальные цифровые вычислительные машины, которые стали двигателем *второй научно-технической революции.* Её результатом явилось усиление интеллектуальных возможностей людей за счет использования информационной техники. Технические устройства и системы, усиливающие интеллектуальные возможности людей, называются *искусственным интеллектом.* Интеллект человека и других созданных природой интеллектуальных систем называется *естественным интеллектом.* Примерами изделий второй научно-технической революции являются: телеграф, телефон, радио, телевизор, магнитофон, видеомагнитофон, ЭВМ, персональный компьютер, роботы, Интернет, автоматизированные системы управления, экспертные системы, машинная графика. Первая научно-техническая революция направлена на познание *внешнего* (*объективного*) *мира*, вторая – *внутреннего* (*субъективного*) *мира* человека.

Родоначальником современных представлений об интеллекте является Рене Декарт (1596-1650). Первая научно-техническая революция стимулировала быстрое развитие физики внешнего мира и классической математики (в основном – числовой), которые стали её научным фундаментом. Прогрессу второй научно-технической революции призваны содействовать порождаемые ею физика и математика информационных процессов. Физика информационных процессов называется *физикой интеллекта* или физикой внутреннего мира. Математика информационных процессов имеет преимущественно не числовую, а логическую природу, иначе она называется *математикой интеллекта.* Логическая часть математики интеллекта называется *теорией интеллекта.* Основателем теории интеллекта является Готфрид Вильгельм Лейбниц (1646-1716). Математика интеллекта представляет собой математическое учение, ориентированное на познание природы интеллекта и на решение задач компьютеризации и информатизации. На рис. 1 показано соотношение между теорией интеллекта, математикой интеллекта и числовой математикой.

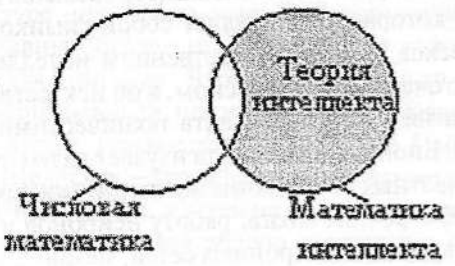


Рис. 1

Математика интеллекта используется как практический инструмент в физике и *технике интеллекта.* Задача физики интеллекта состоит в математическом описании естественных информационных процессов, наблюдаемых в природе, и искусственных информационных процессов, создаваемых техникой интеллекта. Математика интеллекта разрабатывает новый формальный язык и математические методы, необходимые для такого описания. Физика и техника интеллекта выдвигают перед математикой интеллекта новые задачи. Техника интеллекта ставит перед физикой интеллекта актуальные для нее задачи. Прямые и обратные связи (последние указаны жирными стрелками) между математикой, физикой и техникой интеллекта представлены на рис. 2.

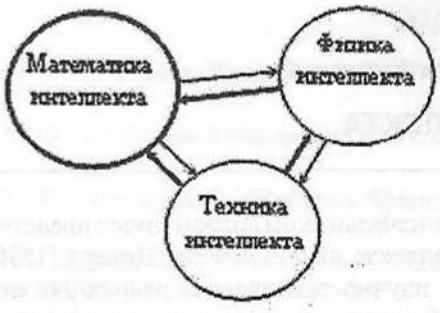


Рис. 2

Вместе взятые, физика и математика интеллекта образуют *анализ интеллекта*, который является научным фундаментом и инструментом в деле познания существующих и разработки новых интеллектуальных систем. Логическая часть анализа интеллекта называется *логическим анализом*. Основателем логического анализа является Бертран Рассел (1872–1970). Примерами объектов физики интеллекта могут служить: зрение, слух, восприятие, оценка ситуаций, принятие решений, язык, речь, мышление, внимание, узнавание, сознание, воля, творчество. Примеры объектов техники интеллекта: электронные схемы ЭВМ, программы для ЭВМ, базы данных и знаний, продукционные системы, интеллектуальные агенты, фреймы, семантические сети, иерархические и динамические информационные системы, средства защиты информации.

Анализ интеллекта вливается в более широкое научное направление, называемое *бионикой интеллекта*, которая представляет собой физико-математическое учение о естественном интеллекте, в первую очередь человеческом, и об искусственном воспроизведении интеллекта техническими средствами. Бионика интеллекта изучает разум человека и животных, механизмы эволюции живых организмов, строение мозга, работу нейронов и образованных из них нейронных сетей, механизмы зрения, слуха, восприятия, речи, понимания, узнавания, принятия решений, оценивания и т. п. Она входит особым разделом в *бионику*, изучающую строение и функционирование живых организмов, которые рассматриваются как изделия некой разумной материальной системы, называемой *генетическим интеллектом*. Изделия генетического интеллекта гораздо совершеннее всего того, что может создать человек со всем своим умом. Отсюда вытекает, что генетический интеллект умнее человеческого, и полезно людям у него подучиться. Бионика предлагает добытые ею знания инженерам различного профиля в качестве подсказки при создании ими технических устройств. Бионика имеет много разных направлений соответственно тем областям техники, которые она обслуживает. Имеются, к примеру, строительная бионика, изучающая конструкцию костей и скелета животных, строение пчелиных сот как инженерных сооружений; гидробионика (форма и движения рыб); аэробионика (механизм полета птиц, форма крыла, движения крыльев).

Для дела совершенствования искусственного интеллекта важна задача формального описания интеллекта человека, любых иных видов естественного интеллекта. Компьютер – это математическая машина, она может усвоить только те знания, которые предварительно были выражены людьми на точном математическом языке. Не менее важна и задача формального описания искусственного интеллекта, который тоже нуждается в физико-математическом анализе. Известны три основных подхода к формальному описанию интеллекта: функциональный, субъективный и материальный. *Функциональный подход* заключается в анализе наблюдаемого извне поведения интеллектуальной системы. *Субъективный подход* заключается в наблюдении интеллектуальной системы изнутри (рис. 3).

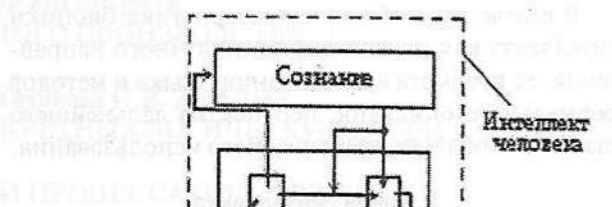


Рис. 3

Психические процессы представляют собой преобразования различных внутренних состояний интеллектуальной системы друг в друга. Психофизические процессы преобразуют состояния внешнего мира, информация о которых поступает в интеллектуальную систему, в ее внутренние состояния и обратно. *Материальный подход* состоит в том, что исследуется структура материального носителя интеллекта (например нейронных сетей мозга человека). Субъективный и функциональный подходы, соединенные вместе, образуют *идеальный подход*, который противостоит материальному. Субъективный и материальный подходы, вместе взятые, составляют *структурный подход*, который противостоит функциональному. Функциональный и материальный подходы, используемые совместно, образуют *объективный подход*, который противостоит субъективному. Связь между подходами выражается в виде треугольника анализа интеллекта (рис. 4).

Рис. 4

Всесторонний анализ интеллекта человека требует совместного использования всех подходов. Результаты такого анализа представляют интерес для разработчиков искусственного интеллекта. Анатомо-физиологические данные о мозге и нервной системе человека могут использоваться при разработке схем и отдельных элементов вычислительных машин. Сведения об интеллектуальных физических реакциях человека на внешние воздействия могут оказаться полезными при совершенствовании функций вычислительных устройств, выполняющих роль отдельных звеньев технических систем. Данные об интеллекте человека, полученные интроспективным путем, могут существенно помочь новым программным разработкам для диалоговых систем, а также любых других человеко-машинных систем, работающих в интерактивном режиме.

Исследование интеллекта человека – это трудоемкий, медленный и длительный процесс добывания информации. Нет ли способа сократить путь за счет получения готовой информации об интеллекте человека из какого-нибудь внешнего источника знаний, подобно тому как школьник получает знания из учебника? Один из таких способов – бионический – мы уже упомянули. Однако на бионическом пути тоже требуются исследования, быть может, не такие обширные, как если бы их пришлось выполнять без какой бы то ни было подсказки со стороны. Бионическая идея состоит в том, чтобы изучать изделие творца. Но можно обратиться непосредственно к замыслам творца человека, которым является генетический интеллект. Эти замыслы записаны в генетическом коде человека. Подсчитано, что запись этого кода на бумаге буквами латинского алфавита имела бы вид библиотеки, составленной из тысячи объемистых томов [2, с. 115]. Сегодня эта запись еще не прочитана полностью и не расшифрована, но в обозримом будущем она может стать источником ценной информации для разработчиков искусственного интеллекта. В этом направлении в настоящее время во всем мире ведутся интенсивные изыскания.

Представляется возможным и другой способ получения готовой информации об интеллекте человека. Этот способ еще более проблематичен, чем обращение к генетическому коду человека. У одного из авторов этой статьи обучался аспирант – профессиональный йог. Он утверждал, что с помощью медитации общается с высшим разумом. Для проверки справедливости этого утверждения аспиранту было предложено обратиться наверх для получения информации о механизме человеческого интеллекта. Аспирант приступил к делу и в результате за месяц написал текст объемом около 10 машинописных страниц. Текст имеет вид начальной части тщательно и квалифицированно написанного учебника о механизме человеческого интеллекта. В нем даются исчерпывающие (как утверждал аспирант – автор текста) перечень и характеристика подсистем (около десятка) интеллекта человека и связей между ними. Воспользоваться практически сообщаемыми в этом тексте сведениями разработчик искусственного интеллекта не имеет никакой возможности, так как текст обрывается «на самом интересном месте», и читатель остается в неведении, что же конкретно представляют собой эти подсистемы. У аспиранта было спрошено, когда же он доберется до самого низа и тем самым завершит свое описание механизма интеллекта человека. Он ответствовал: уровней в иерархической структуре интеллекта много; информация нарастает лавинообразно при переходе на каждый нижеследующий этаж. Чтобы все описать до конца и с исчерпывающей полнотой мне не хватит всей моей жизни. Тем все дело и кончилось. Исследование интеллекта – это движение снизу вверх: сначала изучаем детали, а затем на их основе постепенно воссоздаем все более общую картину его структуры. Обучение же механизму интеллекта – это движение сверху вниз. Пока не доберемся до самого низа, мы ничего конкретного не узнаем о механизме интеллекта. Чтобы изучить предмет, надо прочесть учебник до конца, а не довольствоваться его первыми страницами.

Различают частный и общий анализы интеллекта. *Частный анализ интеллекта* изучает конкретные интеллектуальные системы и их отдельные механизмы. Примерами разделов частного анализа интеллекта могут служить: анализ человеческого, машинного и генетического интеллекта; анализ механизмов зрения, слуха, узнавания, восприятия, понимания, оценивания и принятия решений. *Общий анализ интеллекта* изучает все то общее, что содержится во всевозможных интеллектуальных системах. С позиций общего анализа интеллекта любая интеллектуальная система представляет собой материальное воплощение некото-

рой части *универсальной логической алгебры*, разработка которой является одной из важнейших задач теории интеллекта. Разработка универсальной логической алгебры достигается главным образом за счет *алгебраизации логики*. Источниками идей для алгебраизации логики служат классическая математика, практика интеллектуальной деятельности человека, интроспективные сведения об интеллекте, физика и техника интеллекта, а также собственная логика развития теории интеллекта, логического анализа, бионики интеллекта. Приведем примеры разделов общего анализа интеллекта: алгебры булевых функций, отношений, множеств, предикатов, предикатных операций; теории линейных логических преобразований, моделей; учения о композиции и декомпозиции математических структур, об аксиоматических теориях, об алгебрах.

Анализ интеллекта входит составной частью в более широкую задачу *синтеза интеллекта*, которая схематически изображается в виде квадрата синтеза интеллекта. Проблема синтеза интеллекта разработана наукой и техникой еще очень слабо. Приведенная на рис. 5 схема синтеза интеллекта несовершенна, но лучшей схемы, насколько нам известно, пока нет. Вершины квадрата представляют собой учебу и исследование в области интеллектуальных систем, разработку и применение интеллектуальных систем. Противоположные стороны квадрата задают пары знание—изделие и производство—потребление, относящиеся к интеллектуальным системам. Каждая вершина квадрата подпитывается остальными. В процессе познания и совершенствования интеллектуальных систем квадрат синтеза интеллекта растет вверх и вправо в направлении всеведения и всемогущества. Интеллектуальная система обладает *всеведением*, если она все о себе знает, и *всемогуществом*, – если ей полностью подвластен собственный материальный носитель. Абсолютное всеведение и всемогущество для любой реально функционирующей интеллектуальной системы недостижимы, в этом – гарантия возможности не ограниченного во времени ее познания и совершенствования. Человеческий интеллект актуально не всеведущ и не всемогущ, но можно утверждать, что потенциально он этими свойствами все же обладает, поскольку человечество в целом может, развиваясь, достичь любого рубежа на пути самопознания и самосовершенствования. Можно также говорить и о потенциальном *бессмертии* биологического вида homo sapiens.

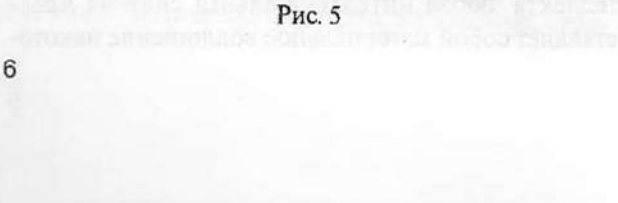


Рис. 5

2. Компараторная идентификация

Человек, интеллектуальная деятельность которого изучается, называется испытуемым. Математическое описание любого процесса называется его идентификацией. Важной задачей анализа интеллекта является идентификация интеллектуальной деятельности человека. Различают прямую и косвенную идентификацию процессов. При прямой идентификации на вход идентифицируемого процесса f подаются сигналы x, выбираемые из некоторого множества A, и регистрируются ответные сигналы y на выходе процесса. Всевозможные сигналы y образуют множество B. Цель идентификации заключается в получении математического описания функции $f(x)=y$, отображающей множество A на множество B, которая характеризует ход (поведение) изучаемого процесса. Функция f называется характеристической функцией процесса (рис. 6).

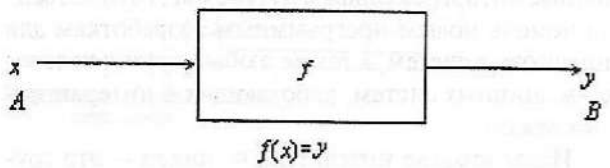


Рис. 6

Отличительная особенность прямой идентификации состоит в том, что входные и выходные сигналы изучаемого процесса доступны для прямого физического наблюдения и измерения. Методом прямой идентификации можно изучать только наблюдаемое извне поведение испытуемого, но не его внутренний субъективный мир. Приведем примеры прямой идентификации процесса: любой классический физический эксперимент, наблюдение за игрой в шахматы, наблюдение за работой ЭВМ, изучение поведения людей. Однако для информатизации представляют интерес не только разумное поведение людей, но и те внутренние субъективные состояния и информационные процессы, которые явились причиной этого поведения. Внутренние состояния субъективны, они недоступны для внешнего наблюдения. Для изучения внутренних состояний испытуемого метод прямой идентификации непригоден. Идентификация называется *косвенной*, если требуется найти характерис-

тическую функцию процесса в условиях, когда входной или выходной сигналы процесса недоступны для непосредственного физического наблюдения. При косвенной идентификации приходится, кроме самого процесса, математически описывать его неизвестные входные или выходные сигналы. Примерами косвенной идентификации процесса могут служить: определение с берега расстояния до плывущего в море корабля, измерение высоты пирамиды Хеопса, предсказание времени солнечного затмения. Найдя метод решения этих и подобных им задач, Фалес Милетский в 6 веке до н.э. положил начало косвенной идентификации.

В математике интеллекта разрабатывается и изучается, а в физике и технике интеллекта широко используется *компараторная идентификация*, являющаяся одним из важных видов косвенной идентификации. Компаратором K называется устройство с m входами $y_1, y_2, \ldots, y_m$ и одним выходом t (рис. 7).

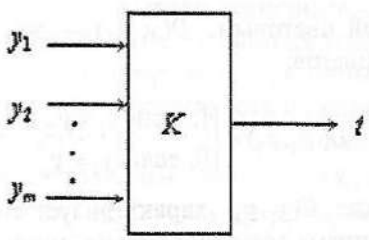


Рис. 7

Здесь t – двоичная реакция компаратора, $t \in \Sigma$, $\Sigma = \{0,1\}$; $y_1, y_2, \ldots, y_m$ – внутренние состояния информационной системы, $y_1 \in B_1, y_2 \in B_2, \ldots, y_m \in B_m$; $B_1, B_2, \ldots, B_m$ – вообще говоря, различные множества внутренних состояний. Приведем примеры внутренних состояний информационной системы: для человека – ощущение, присутствующее в поле зрения в данный момент времени; цвет и яркость зрительного ощущения в данной точке поля зрения; слуховое ощущение, его громкость и тембр; мысли, эмоции, намерения, оценки, представления, восприятия; для ЭВМ – текст или рисунок в файле, программа с некоторым именем, двоичный код в заданном регистре.

Компаратор устанавливает, находятся или нет его входные сигналы $y_1, y_2, \ldots, y_m$ в заданном отношении K. Если – да, то он реагирует сигналом $t = 1$, если – нет, то сигналом $t = 0$. Примеры компаратора: сознание человека, сравнивающее ощущения цветов или звуков на равенство, определяющее, является ли один заданный звук громче другого, устанавливающее предпочтение субъективной оценки одного места работы другому, сравнивающее оценки вариантов при выборе мужа, жены, друга, вуза, устанавливающее, соответствует ли смысл данного утверждения знаниям о заданной ситуации, следует ли заданное математическое утверждение B из математического утверждения A, присущ ли конкретной словоформе заданный падеж (например, «синей, родительный» – да; «столом, родительный» – нет), правилен ли ответ студента на экзамене, является ли данный цвет красным; программа ЭВМ, определяющая, имеются ли в тексте орфографические ошибки, русский ли это текст, содержит ли вводимый в машину текст вирусы, совпадают ли два заданные текста, выражают ли два предъявленные предложения одну и ту же мысль, соответствует ли данный рисунок словесному описанию. Своим поведением компаратор реализует предикат $K(y_1, y_2, \ldots, y_m) = t$, соответствующий отношению K, который называется предикатом компаратора. Ко входам компаратора K подключаются своими выходами идентифицируемые информационные процессы $f_1, f_2, \ldots, f_m$ (рис. 8). Здесь $x_1 \in A_1, x_2 \in A_2, \ldots$, $x_m \in A_m$ – физические сигналы; $A_1, A_2, \ldots, A_m$ – множества физических сигналов различного типа.

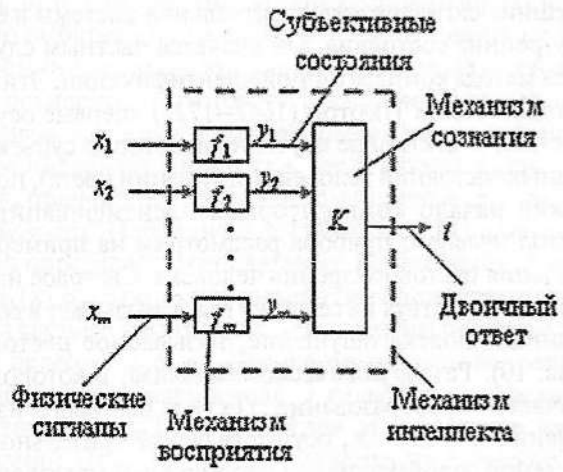


Рис. 8

Информационные процессы, вместе с подсоединенным к ним компаратором, называются идентифицируемым объектом (рис. 9). Символом P обозначен предикат объекта $P(x_1, x_2, \ldots, x_m) = t$. Он выражается, если верить субъективному свидетельству испытуемого, в следующем виде:

$$P(x_1, x_2, \ldots, x_m) = K(f_1(x_1), f_2(x_2), \ldots, f_m(x_m)).$$

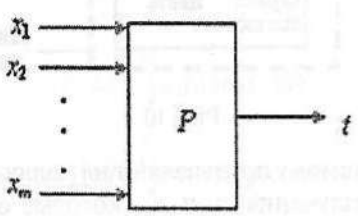


Рис. 9

Сигналы $y_1 = f_1(x_1)$, $y_2 = f_2(x_2), \ldots, y_m = f_m(x_m)$ являются внутренними состояниями объекта P,

недоступными для внешнего наблюдения. Примерами идентифицируемых объектов могут служить: человек, устанавливающий, равны или нет цвета двух предъявленных ему световых излучений; ЭВМ на атомной электростанции, блокирующая неправильные действия обслуживающего персонала. Простейшая задача компараторной идентификации заключается в том, чтобы по заданному компаратору и по известным свойствам объекта P математически описать выходные сигналы $y_1, y_2, \ldots, y_m$ процессов $f_1, f_2, \ldots, f_m$ и сами эти процессы. Более сложные задачи компараторной идентификации возникают, когда приходится одновременно рассматривать систему объектов $P_1, P_2, \ldots, P_n$, связанных друг с другом, где n — число объектов в системе.

3. Метод нулевого прибора

Метод нулевого прибора представляет собой наиболее употребительное средство объективного изучения внутренних состояний и преобразований внешних сигналов интеллектуальной системы в её внутренние состояния. Он является частным случаем метода компараторной идентификации. Этим методом Исаак Ньютон (1642–1727) впервые осуществил объективное изучение некоторых субъективных состояний человека (ощущений цвета), положив начало компараторной идентификации. Метод нулевого прибора рассмотрим на примере изучения цветового зрения человека. Световое излучение, действуя на сетчатку глаза, вызывает в сознании человека ощущение, называемое цветом (рис. 10). Раздел логического анализа, в котором изучается преобразование $f(x)=y$ светового излучения x в цвет y, осуществляемое зрительной системой человека, называется теорией цветового зрения.

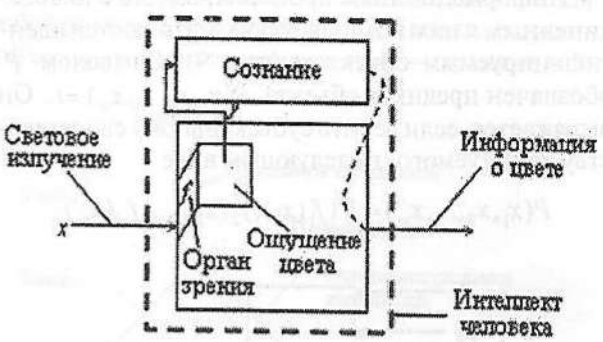


Рис. 10

Испытуемому предъявляют на полях сравнения световые излучения x_1 и x_2, которые он воспринимает в виде цветов y_1 и y_2. Если цвета совпадают друг с другом, то испытуемый должен отреагировать ответом $t=1$, если же не совпадают, то — ответом $t=0$ (рис. 11).

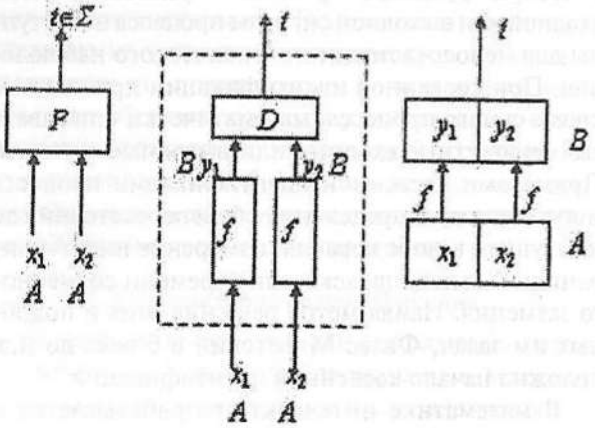


Рис. 11

Своим поведением испытуемый (если верить его субъективному свидетельству) реализует предикат

$$P(x_1,x_2)=D(f(x_1),f(x_2)),$$

называемый цветовым. $D(y_1,y_2)$ — это предикат равенства цветов:

$$D(y_1,y_2)=\begin{cases}1, & \text{если } y_1=y_2,\\ 0, & \text{если } y_1=y_2.\end{cases}$$

Предикат $D(y_1,y_2)$ характеризует собой действие механизма сознания испытуемого, анализирующего ощущения цвета y_1, y_2 и устанавливающего их равенство или неравенство. Функция $f(x)=y$ характеризует собой преобразование объективного светового излучения x в субъективный цвет y, осуществляемое зрительной системой человека. Компаратор, реализующий предикат равенства, называется нулевым прибором.

Чисто объективные опыты на испытуемом с высокой точностью демонстрируют, что предикат P рефлексивен $\forall x\in A\ P(x,x)$, симметричен $\forall x_1,x_2\in A\ \ (P(x_1,x_2)\supset P(x_2,x_1))$ и транзитивен $\forall x_1,x_2,x_3\in A\ \ (P(x_1,x_2)\wedge P(x_2,x_3)\supset P(x_1,x_3))$. Любой двуместный предикат, обладающий свойствами рефлексивности, симметричности и транзитивности, называется предикатом эквивалентности. Рефлексивность цветового предиката означает, что одинаковые излучения порождают одинаковые цвета. Симметричность цветового предиката означает, что при перестановке излучений местами равенство цветов не нарушается. Транзитивность цветового предиката означает, что если цвета первого и второго излучений равны, а также равны цвета второго и третьего излучений, то цвета первого и третьего излучений также будут равны.

Теорема об общем виде предиката эквивалентности. Любой предикат эквивалентности и только предикаты эквивалентности можно представить в виде $P(x_1,x_2)=D(f(x_1),f(x_2))$ при подходящем выборе множества B и функции $f: A\to B\ \ (x_1,x_2\in A)$.

Функция f называется характеристической функцией эквивалентности. Характеристическая функция цветового предиката математически выражает процесс преобразования зрительной системой человека светового излучения в цвет. В теории цветового зрения существование (в логическом смысле этого слова) цвета y и функции $f(x)=y$, преобразующей световое излучение в цвет, выводится с помощью только что приведенной теоремы лишь из характера двоичных ответов испытуемого, иначе говоря, только из объективно наблюдаемых свойств цветового предиката. Таким образом, опираясь лишь на факты наблюдаемого извне поведения испытуемого и не прибегая к интроспективному свидетельству испытуемого, мы установили существование цвета y светового излучения x и наличие преобразования $f(x)=y$. Однако, что собой конкретно представляет цвет y и преобразование f, мы пока не знаем.

Для отыскания конкретного описания цвета y и функции f привлекаем дополнительные свойства предиката P: его аддитивность, однородность, трехмерность и непрерывность, которые проверяются в чисто объективных, хотя и непростых, экспериментах на испытуемом. Аддитивность формулируется следующим образом: $\forall x_1,x_2,x_1',x_2' \in A$ $(f(x_1)=f(x_2)\wedge f(x_1')=f(x_2')\supset f(x_1+x_1')=f(x_2+x_2'))$. Записывая аддитивность цветового предиката P, с целью придания ей более удобочитаемого вида, мы использовали выражение $f(x_1)=f(x_2)$ вместо равносильного ему выражения $P(x_1,x_2)=1$. Теперь это можно делать, поскольку существование функции f и ее значений уже доказано. В формулировке же свойств рефлексивности, симметричности и транзитивности такая замена была бы невозможна. Сложение x_1+x_2 световых излучений x_1,x_2 достигается их совмещением в пространстве (для этого используется специальная оптическая аппаратура). Аддитивность цветового предиката означает, что суммы равноцветных излучений равноцветны.

Однородность формально записывается в виде $\forall x_1,x_2 \in A$ $\forall \alpha \in R$ $(f(x_1)=f(x_2)\supset f(\alpha x_1)=f(\alpha x_2))$. Здесь α – любое действительное число, $\alpha \in R$, R – множесто всех действительных чисел. Умножение αx излучения $x=x(\lambda)$ на неотрицательное число α достигается изменением его мощности в α раз при сохранении его спектрального состава (это можно сделать, к примеру, с помощью диафрагмирования). Пример умножения спектра $x(\lambda)$ светового излучения x на число 2 показан на рис. 12.

Здесь λ – длина волны световых колебаний; $x(\lambda)$ – спектр светового излучения (зависимость его мощности x от длины волны λ); $[\lambda_1,\lambda_2]$ – диапазон длин волн световых колебаний; $\lambda_1=0{,}4$ мкм (фиолетовые световые колебания); $\lambda_2=0{,}8$ мкм (красные световые колебания). Умножение излучения на отрицательное число означает его перемещение на противоположное поле сравнения. Однородность цветового предиката означает, что равноцветные излучения после умножения их на одно и то же число снова будут равноцветными.

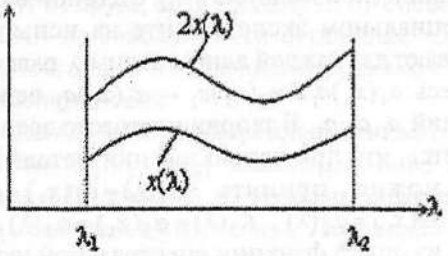


Рис. 12

Трехмерность формально описывается следующим образом: $\exists a_1,a_2,a_3 \in A$ $\forall x \in A$ $\exists!\alpha_1,\alpha_2,\alpha_3 \in R$ $(f(x)=f(\alpha_1 a_1+\alpha_2 a_2+\alpha_3 a_3))$ при единственном наборе чисел $\alpha_1,\alpha_2,\alpha_3 \in R$. Световые излучения a_1,a_2,a_3 и их цвета называются основными. Их можно выбирать по-разному. В роли основных цветов обычно используют красный, зеленый и синий. Трехмерность цветового предиката означает, что смесь $\alpha_1 a_1+\alpha_2 a_2+\alpha_3 a_3$ основных излучений можно подравнять по цвету к цвету любого излучения x, причем при единственном варианте пропорций $\alpha_1,\alpha_2,\alpha_3$. Набор чисел $\alpha_1(x),\alpha_2(x),\alpha_3(x)$ однозначно определяет цвет светового излучения x. Числа $\alpha_1(x),\alpha_2(x),\alpha_3(x)$ называются координатами цвета светового излучения x в цветовом пространстве с базисом a_1,a_2,a_3. Для формулировки непрерывности принимаем в роли A гильбертово пространство $L_2[\lambda_1,\lambda_2]$ всех спектров $x(\lambda)$ световых излучений. При непрерывном изменении светового излучения $x \in L_2[\lambda_1,\lambda_2]$ числа $\alpha_1(x),\alpha_2(x)$, $\alpha_3(x)$ изменяются непрерывно. Любой аддитивный, однородный и непрерывный предикат называется линейным предикатом.

Теорема об общем виде трехмерного линейного предиката эквивалентности. Характеристическая функция $f: L_2[\lambda_1,\lambda_2] \to R^3$ любого трехмерного линейного предиката эквивалентности может быть представлена в виде:

$$y=f(x)=(\alpha_1(x),\alpha_2(x),\alpha_3(x)),$$

где

$$\alpha_1(x)=\int_{\lambda_1}^{\lambda_2} x_1(\lambda)K_1(\lambda)d\lambda;$$

$$\alpha_2(x)=\int_{\lambda_1}^{\lambda_2} x_2(\lambda)K_2(\lambda)d\lambda;$$

$$\alpha_3(x)=\int_{\lambda_1}^{\lambda_2} x_3(\lambda)K_3(\lambda)d\lambda$$

при подходящем выборе функций $K_1(\lambda),K_2(\lambda),K_3(\lambda)$.

Интегралы приведенного вида называются цветовыми. Функции $K_1(\lambda), K_2(\lambda), K_3(\lambda)$ называются функциями спектральной чувствительности зрения в базисе a_1, a_2, a_3. Их определяют экспериментально. Один из возможных способов такого определения называется методом Максвелла. Он заключается в следующем. Берут монохроматическое излучение единичной мощности x_λ с длиной волны λ и в специальном эксперименте на испытуемом отыскивают для каждой длины волны λ равноцветную смесь $\alpha_1(x_\lambda)a_1 + \alpha_2(x_\lambda)a_2 + \alpha_3(x_\lambda)a_3$ основных излучений a_1, a_2, a_3. В теории цветового зрения доказывается, что при использовании метода Максвелла можно принять $K_1(\lambda) = \alpha_1(x_\lambda) = \alpha_1(\lambda)$, $K_2(\lambda) = \alpha_2(x_\lambda) = \alpha_2(\lambda)$, $K_3(\lambda) = \alpha_3(x_\lambda) = \alpha_3(\lambda)$. Найденные из опыта функции спектральной чувствительности зрения $K_1(\lambda), K_2(\lambda), K_3(\lambda)$ усредняют по многим испытуемым. На этой основе получают унифицированные функции спектральной чувствительности зрения, называемые кривыми сложения. Используют разные системы кривых сложения. Их вид зависит от выбора базиса a_1, a_2, a_3. Имеется международный стандарт на кривые сложения – система RGB. В нем в роли основных использованы монохроматические излучения красного, зеленого и синего цветов. Разные варианты кривых сложения получают линейным пересчетом кривых сложения стандартной системы RGB. В системе КЗС или основной физиологической системе кривые сложения имеют вид (рис. 13).

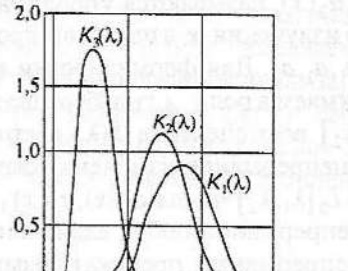


Рис. 13

Приведем примеры областей возможного применения теории цветового зрения: машинная графика, робототехника, телевидение, видеотехника, кино, архитектура, текстильная промышленность, светотехника, лакокрасочное производство, медицина, медицинское приборостроение, полиграфия, культура и искусство, узнающие устройства и системы. Приведем также примеры объектов, которые можно изучать по методу нулевого прибора: преобразование цвета в яркость, цветовой тон и насыщенность, звука – в громкость и тембр, текста – в его смысл, предмета – в зрительный образ, в набор признаков, ситуации – в её субъективную оценку, пятен краски на бумаге – в букву, звука – в фонему, звучащей речи – в текст.

4. Логическая идентификация

Компараторная идентификация используется для изучения периферийных механизмов интеллекта. К ним относятся восприятие, узнавание и понимание. А есть еще центр интеллекта – само сознание. Для идентификации центральных механизмов сознания может быть использована *логическая идентификация*. Для логической идентификации, так же как и для компараторной, характерно использование только двоичного ответа испытуемого со значениями «истина» и «ложь». Используя исключительно двоичный ответ, мы попадаем в царство логики. Логическая идентификация является обобщением метода компараторной идентификации. Компараторную идентификацию можно рассматривать как частный случай логической. К логической идентификации сводятся любые задачи косвенной идентификации. При логической идентификации, не сводящейся к компараторной, промежуточные субъективные состояния не появляются, поэтому исследователь не имеет возможности на них опереться. Сознание немое, оно не отдает самому себе отчета о разыгрывающихся в нем процессах. Механизмы сверхсознания психологическими методами не могут быть обнаружены.

Пусть $P_1, P_2, \ldots, P_n$ – идентифицируемые объекты, реализующие некоторые предикаты

$$P_1(x_{11}, x_{12}, \ldots, x_{1m_1}),$$
$$P_2(x_{21}, x_{22}, \ldots, x_{2m_2}),$$
$$\ldots\ \ldots\ \ldots\ \ldots\ \ldots$$
$$P_n(x_{n1}, x_{n2}, \ldots, x_{nm_n}),$$

n – число объектов; $m_1, m_2, \ldots, m_n$ – размерности предикатов; x_{ij} – аргументы предикатов $(i = \overline{1,n}, j_i = \overline{1,m_i})$. Из экспериментального изучения поведения объектов $P_1, P_2, \ldots, P_n$ выводятся их характеристические свойства, выражаемые системой логических уравнений:

$$F_1(P_1, P_2, \ldots, P_n) = 1,$$
$$F_2(P_1, P_2, \ldots, P_n) = 1,$$
$$\ldots\ \ldots\ \ldots\ \ldots\ \ldots \qquad (1)$$
$$F_r(P_1, P_2, \ldots, P_n) = 1.$$

Здесь r – число уравнений; $F_1, F_2, \ldots, F_r$ – предикаты от предикатов. Уравнения, связывающие неизвестные предикаты, называются логическими условиями. В логических условиях концовка "=1" для краткости обычно не пишется. Логические условия (1) называются аксиомами теории объектов $P_1, P_2, \ldots, P_n$.

Рассмотрим примеры аксиом теорий объектов. 1) Теория цветового зрения. В ней объектом идентификации служит единственный предикат

$P(x_1,x_2)$, называемый цветовым. Первый этаж рассмотренной нами ранее теории цветового зрения – теория эквивалентности. Она основывается на трех аксиомах: рефлексивности, симметричности и транзитивности. Вторым этажом теории цветового зрения служит линейная интегральная теория. Она базируется на аксиомах аддитивности, однородности, трехмерности и непрерывности. В конечномерном варианте цветовой теории достаточно иметь в ней аксиомы аддитивности, однородности и трехмерности, а в бесконечномерном – аддитивности, трехмерности и непрерывности. 2) Теория натурального ряда. В ней фигурируют два объекта – предикат $N(x)$, выражающий понятие множества всех натуральных чисел, и предикат $Q(x,y)$, представляющий собой понятие счета. Предикат $Q(x,y)$ выражает связь произвольно выбранного числа x в натуральном ряду с числом y, непосредственно следующим за ним. Теория натурального ряда основывается на аксиомах всюду определенности $\forall x\in N\ \exists y\in N\ Q(x,y)$, однозначности $\forall x,y,z\in N\ (Q(x,y)\wedge Q(x,z)\supset D(y,z))$, инъективности $\forall x,y,z\in N\ (Q(x,z)\wedge Q(y,z)\supset D(x,y))$, единицы $\exists! y\in N\ \forall x\in N\ \neg Q(x,y)$ (единица y, определяемая этой аксиомой обозначается символом 1) и математической индукции $\forall M\subseteq N(M(1)\wedge \wedge\forall x,y\in N\ (M(x)\wedge Q(x,y)\supset M(y))\supset\forall x\in N\,M(x))$. Аксиома всюду определенности означает, что для каждого натурального числа существует хотя бы одно непосредственно следующее за ним в натуральном ряду натуральное число. Смысл аксиомы однозначности состоит в том, что за каждым натуральным числом не может непосредственно следовать более одного натурального числа. Вместе взятые, эти две аксиомы постулируют существование функции счета $q(x)=y$. Аксиома единицы гласит, что в натуральном ряду имеется единственное число 1, которому непосредственно не предшествует ни одно натуральное число. Аксиома математической индукции выражает следующую мысль. Если некоторым свойством обладает число 1, и из предположения, что этим свойством обладает произвольно выбранное натуральное число, логически вытекает, что тем же свойством обладает и число, непосредственно следующее за ним, тогда таким свойством обладают все натуральные числа.

Совокупность истинных высказываний об объектах $P_1,P_2,\ldots,P_n$ называется теорией T этих объектов. Совокупность каких-либо высказываний теории T, из которых можно логически вывести любое истинное высказывание теории T (и только истинные высказывания), называется системой аксиом теории T. Для создания аксиоматических теорий нужно уметь выполнять многие виды работ. К примеру, необходимы опытная проверка аксиом, поиск и формулировка аксиом, разработка алгебро-логического языка для записи аксиом, разработка теории построения аксиоматических теорий, упрощение формулировки отдельных аксиом, разработка методов упрощения аксиом и их совокупности, проверка на полноту и несократимость системы аксиом, сокращение до минимума числа аксиом в системе. Аксиомы называются независимыми друг от друга, если ни одну из них невозможно вывести из совокупности остальных. Система аксиом, состоящая из независимых аксиом, называется несократимой.

Интеллект человека формально рассматривают как систему $\{P_1,P_2,\ldots,P_n\}$ взаимосвязанных объектов $P_1,P_2,\ldots,P_n$, каждый из которых характеризует один из механизмов интеллекта. *Теория интеллекта* представляет собой теорию системы объектов. В своем нынешнем состоянии теория интеллекта еще не может формально представить человеческий интеллект как единую, полную и завершенную систему взаимосвязанных механизмов. Дело осложняется еще и тем, что человеческий интеллект – система развивающаяся. Пока что осуществляется изолированное изучение отдельных, причем наиболее простых, механизмов интеллекта. Укажем примеры механизмов интеллекта: зрение, слух, осязание, узнавание, понимание, язык, мышление, сознание, эмоции, воля, критика, воображение, оценивание, творчество, интуиция, подсознание, умения, знания, потребности, совесть, привычки, внимание, идеалы, способности, обучение, развитие.

Набор конкретных предикатов $(P_1^*,P_2^*,\ldots,P_n^*)$, удовлетворяющий аксиомам теории T, называется моделью теории T. В системе (1) символы $P_1,P_2,\ldots,P_n$ обозначают переменные предикаты. Полагаем $P_1=P_1^*,P_2=P_2^*,\ldots,P_n=P_n^*$. Возможны случаи, когда аксиомам теории удовлетворяет единственная модель, много моделей или им не удовлетворяет никакая модель. Система аксиом, для которой не существует ни одной модели, называется противоречивой. Для практических применений противоречивые теории бесполезны. Непротиворечивая теория называется полной, если добавление к ее аксиомам любой независимой от них аксиомы делает ее противоречивой. Теория, основанная на аксиомах рефлексивности, симметричности и транзитивности, неполна. Это следует из того, что добавление к ним аксиом аддитивности, трехмерности, однородности и непрерывности не сделало эту теорию противоречивой.

Теория называется однозначной, если ей удовлетворяет единственная модель. Приведем примеры однозначных теорий: теория сложения и умножения натуральных чисел, теория равенства на произвольно выбранном множестве предметов, которая основывается на единственной аксиоме (законе Лейбница):

$$\forall x,y\in A\ (D(x,y)\sim\forall P\subseteq A(P(x)\sim P(y))).$$

Примеры неоднозначных теорий: теория эквивалентности, интегральная теория цвета. Если теория неоднозначна, то, вообще говоря, возможна её конкретизация, что мы и видим на примере теории цветового зрения. Отыскивая функции спектральной чувствительности зрения, мы тем самым осуществляем конкретизацию интегральной теории цвета. Теория называется категоричной, если все её модели изоморфны друг другу. В категоричной теории любые ее две модели могут различаться только терминологически, по существу же они идентичны. Приводим примеры категоричных теорий: евклидова геометрия, арифметика натуральных, рациональных и действительных чисел, теория арифметического пространства фиксированной размерности. Примеры некатегоричных теорий: теория эквивалентности, теории групп и полугрупп, теория множеств.

Запись семейства всех моделей теории T в виде некоторой формулы называется общим видом модели теории T. Примерами общего вида модели теорий могут служить: общий вид эквивалентности $P(x_1,x_2)=D(f(x_1),f(x_2))$, выражения (1) для цветовых интегралов в интегральной теории цвета. Общий вид модели получаем в результате решения системы логических уравнений. Структурной идентификацией объектов называется такое их экспериментальное и теоретическое изучение, которое завершается формулированием аксиом теории и отысканием общего вида моделей этих объектов. Параметрической идентификацией объектов называется такое их экспериментальное и теоретическое изучение, которое завершается отысканием конкретных значений параметров общего вида модели этих объектов. Аксиоматические теории классической математики и теории интеллекта имеют разную направленность: первые направлены на вывод теорем из аксиом; вторые – на отыскание общего вида и параметров модели объектов со свойствами, выраженными аксиомами теории. Приведем примеры возможной области практического применения моделей информационных объектов: эргономика, инженерная психология, автоматическая обработка текстов, робототехника, телевидение, поиск научной и деловой информации в Интернете, машинная графика, экспертные системы, базы знаний, объектное моделирование.

5. Мозгоподобные ЭВМ

Напрашивается вопрос: может ли бионика интеллекта указать магистральное направление для дальнейшего развития компьютеризации и информатизации? Представляется, что такое направление существует и оно состоит в создании мозгоподобных ЭВМ, основанных на принципе параллельного действия. Известный авторитет в области компьютеризации и информатизации, один из богатейших и влиятельнейших людей в мире Билл Гейтс называет систему машинной обработки информации, основанную на принципах работы мозга, *электронной нервной системой* [3, с. 16]. Он прочит ей большое будущее. Настоятельная потребность в ней возникла в связи с развитием Интернета – всемирной информационной сети, охватившей сегодня все сферы жизни. Гейтс даже переключил на ее создание и совершенствование все силы возглавляемой им корпорации.

Нынешние ЭВМ массового применения (в том числе и персональные компьютеры) имеют по сравнению с мозгом человека серьезный недостаток. Это – машины последовательного действия, тогда как мозг человека – система параллельного действия. Резервы увеличения быстродействия решающих элементов ЭВМ исчерпываются. Остается путь наращивания числа одновременно работающих элементов в процессоре компьютера. Уже сейчас имеется практическая возможность, опираясь на успехи микроминиатюризации и удешевления электронных элементов и на достижения в области автоматизации проектирования и изготовления ЭВМ, строить компьютеры с числом элементов до 10^{15}. Однако применительно к нынешним ЭВМ последовательного действия, работающим по принципу программного управления Джона фон Неймана, делать это не имеет смысла, поскольку в них в каждый момент времени одновременно находится в работе лишь небольшое число элементов. Попытки же перехода к машинам параллельного действия пока не дают ожидаемого роста их производительности. Так, например, производительность многопроцессорных ЭВМ растет не пропорционально числу имеющихся в ней процессоров, как, казалось бы, должно быть, а гораздо медленнее. Возникают существенные трудности также и при попытках создания высокопроизводительных нейрокомпьютеров, которые строятся в виде сетей из формальных нейронов. Между тем, существует «вычислительная машина», созданная Природой, а именно – мозг человека, для которой проблема полноценного распараллеливания обработки информации полностью решена.

Мозг человека по сравнению с современной ЭВМ – тихоход. О его «тактовой частоте» можно судить по пропускной способности нервных волокон. Известно, что каждое нервное волокно может пропускать не более 10^3 импульсов в секунду. По проводникам же нынешних ЭВМ передается порядка 10^9 импульсов в секунду. Следовательно, ЭВМ превосходит мозг человека в смысле скорости работы решающих элементов в $10^9 : 10^3 = 10^6$ раз. Тем не менее, по своей производительности мозг превосходит любую ЭВМ. Это обусловлено тем, что мозг человека имеет в своем составе около 10^{15} решающих элементов (в роли которых выступают си-

напсы – стыки между окончаниями нервных волокон), и все они, как свидетельствуют нейрофизиологические данные, работают одновременно. В ЭВМ же последовательного действия в любой момент времени одновременно работает в среднем не более 10^3 элементов. Таким образом, в смысле числа параллельно работающих элементов мозг превосходит машину в $10^{15} : 10^3 = 10^{12}$ раз. В итоге, по своей производительности мозг превосходит современную вычислительную машину последовательного действия в $10^{12} : 10^6 = 10^6$ раз.

Если бы удалось создать ЭВМ параллельного действия, работающую по принципам мозга, которая имеет 10^{15} элементов (последнее, как говорилось выше, возможно уже сейчас), то в результате была бы получена машина, превосходящая по производительности мозг человека в $10^9 : 10^3 = 10^6$ раз. Итак, ЭВМ параллельного действия, работающая по принципам мозга и построенная на современной элементной базе (она называется *мозгоподобной ЭВМ*, по-английски – *brainlike computer*), согласно вышеприведенным оценкам, в случае ее создания будет превосходить нынешние ЭВМ последовательного действия в 10^{12} раз, а мозг человека – в 10^6 раз. Если мозгоподобные ЭВМ удастся создать, то это приведет к значительному повышению темпов компьютеризации и информатизации. Пионеры искусственного интеллекта А. Ньюэлл, Дж.Шоу и Г.Саймон еще в конце 50-х годов XX столетия провозгласили, что глубинный смысл компьютеризации и информатизации заключается в том, чтобы побудить людей заняться познанием самих себя и снабдить их достаточными для этого средствами [4]. Представляется, что в результате создания мозгоподобных ЭВМ появятся небывалые возможности для самопознания и самоусовершенствования самого человека.

Почему же специалисты по нейрокомпьютерам до сих пор не смогли построить мозгоподобную ЭВМ с указанной выше производительностью, несмотря на то, что занимаются они этой проблемой уже около полувека? Попытку ответа на этот вопрос можно найти в книге Д. Хьюбела «Глаз, мозг, зрение» – Нобелевского лауреата, одного из крупнейших в мире специалистов в области анатомии и физиологии нейронных сетей головного мозга человека. Он пишет: «Иногда говорят, что нервная система содержит огромное число случайных межнейронных соединений. Хотя упорядоченность связей и в самом деле не всегда очевидна, я подозреваю, что те, кто говорит о случайных нейронных сетях, не утруждали себя ознакомлением с нейроанатомией. Даже беглый просмотр такой книги, как труд Кахала «Гистология нервной системы», достаточно убеждает кого угодно в том, что в чудовищной сложности нервной системы почти всегда можно усмотреть известную степень упорядоченности. Когда мы видим правильные ряды клеток в мозгу, впечатление создается такое же, как если бы мы рассматривали телефонную станцию, печатный станок или внутренность телевизора, – становится несомненным, что упорядоченность служит какой-то цели. Столкнувшись с тем или иным человеческим изобретением, мы едва ли усомнимся в том, что агрегат в целом, так же как и его отдельные части, обладает вполне постижимыми функциями. Чтобы понять их, нам нужно только прочесть ряд инструкций. В биологии появляется аналогичная вера в функциональную обоснованность и в конечном итоге даже в постижимость структур, которые не были изобретены кем-то, а совершенствовались на протяжении миллионов лет эволюции. Задача нейробиолога ... состоит в том, чтобы выяснить связь порядка и сложности с функцией» [5, с.14].

Высказывание Хьюбела можно понять в том смысле, что технические нейронные сети – это не совсем то, а вернее – совсем не то, чем являются их биологические прототипы. Можно не согласиться с Хьюбелом лишь в одном: к такому положению привело не нежелание инженеров знакомиться с биологическими нейронными сетями, а вынужденное незнание ими принципов их функционирования. В то время как анатомия (то есть строение) нейроструктур на микроскопическом уровне в настоящее время хорошо изучена (выявление и классификация их основных типов были в основном завершены в начале XX века), исследование физиологии (то есть функции) этих структур, несмотря на отдельные достижения, до сих пор буксует. Поэтому инженерам приходится на свой страх и риск самим строить произвольные гипотезы о принципах действия нейронных структур.

В нашем Университете на протяжении последних 40 лет разрабатывается упоминавшаяся ранее бионика интеллекта, в рамках которой предпринята попытка сдвинуть с мертвой точки решение обсуждаемой здесь проблемы. Суть подхода состоит в том, что интеллект человека рассматривается как логика в действии, как некоторое материальное воплощение механизма логики. Были выполнены работы по алгебраизации логики. В результате разработана специальная алгебра для формульного представления отношений и действий над ними. Отношения интерпретируются как мысли интеллекта, а действия над ними – как мышление. Схемная реализация формул этой алгебры приводит к характерным инженерным сетям, которые называются *логическими сетями*.

Каждый тип отношений и операций над ними (а таких типов сравнительно немного) приводит к своему типу логических сетей с легко узнаваемой на глаз архитектурой. При сопоставлении этих типов сетей с основными типами нейроструктур обнаруживается поразительное сходство архитекту-

ры технических и биологических конструкций. Опираясь на это сходство, можно определить функции различных типов нейронных структур и описать в точных математических и технических терминах принципы функционирования мозга. Главное в данном методе – это движение сверху вниз: от общих системных соображений к алгебро-логическим конструкциям, а от них – к логическим сетям, которые затем отождествляются с биологическими нейронными структурами. Специалисты же по нейрокомпьютерам сегодня пытаются идти иным путем: от архитектуры биологических нейронных сетей к угадыванию принципов их действия. По этим гипотетическим принципам строятся формальные нейроны и сети из них. Из-за серьезного дефицита знаний о функциях биологических нейронных структур трудно ожидать, что результат такого "угадывания" приведет к эффективно действующим схемам мозгоподобных ЭВМ. Двигаясь же по пути алгебраизации логики, мы принудительно приходим к естественным принципам построения мозгоподобных ЭВМ. Свернуть в сторону на этом пути нет никакой возможности. В результате вырисовываются простые и ясные способы построения мозгоподобных ЭВМ. Насколько нам известно, они существенно отличаются от всего того, что до сих пор предлагалось для параллельной обработки информации, в частности – при создании ЭВМ параллельного действия.

Важное направление исследований состоит в том, чтобы проводить детальное сравнение структуры логических сетей со структурой биологических нейронных сетей. Эта работа содействует познанию принципов функционирования мозга человека. Если в результате такой работы аналогия между логическими и естественными нейронными сетями будет все более подтверждаться и углубляться, это будет вселять все большую уверенность в том, что алгебраизация логики – правильный путь развития, и что она является главным методом познания интеллекта. Кроме этого обстоятельства, алгебраизация логики и создание прогрессивных ЭВМ не зависят от изучения естественных структур мозга. Впрочем, всегда полезно свериться с решениями природы, поскольку изучение нейронных структур мозга может навести на новые идеи и в области алгебраизации логики.

6. Заключение

Что же ожидает людей в свете перспективы создания мозгоподобных ЭВМ? Не загонит ли технический прогресс человечество в безысходный тупик? На этот вопрос лучше всех, на наш взгляд, ответил еще в 1968 году известный московский математик Г.Н. Поваров в предисловии к русскому изданию книги «Кибернетика» основателя кибернетики Норберта Винера: «Действительно, научно-технический прогресс ставит перед человечеством серьезные проблемы. Стремительное развитие науки и техники возлагает на нас колоссальную ответственность за разумное использование полученного нами могущества. «Кто живет в стеклянном доме, тот не должен бросать камней», – гласит старинная пословица. Человек стал настолько могущественным, что любое его нерассчитанное движение: с роботами, с атомной энергией, с химией – может иметь тяжелые непредвиденные последствия. Это парадокс могущества. Нельзя забывать, однако, что наука и техника не только возлагают новую ответственность на человека, но и доставляют ему новые средства справиться с нею. Это относится и к роботам. Альтернатива «человек или робот», «опасное развитие искусственного разума или своевременный отказ от него», чем ограничивается большинство авторов, имеет третье, более необычайное и, пожалуй, более вероятное решение, если только искусственный разум и искусственная жизнь вообще возможны. Человек, научившийся создавать искусственный разум и искусственную жизнь, не остановится перед коренной переделкой самого себя. Не роботы вместо людей, а новый человек вместо старого! Человек будущего вряд ли останется таким же «натуральным» существом, таким же теплокровным позвоночным, каким он вышел из горнила естественного отбора. Почти наверное, он будет искусственно развивать свой мозг и свое тело, будет по воле лепить и изменять свою физическую оболочку. Ему по силам быть впереди любого возможного робота. Это будет биологическая революция, и если смелые гипотезы оправдаются, она будет означать преобразование всего человеческого существования. Быть может, далекий смысл «безумной» винеровской идеи о передаче человека по телеграфу и есть достижение человеком перевоплощаемости? Позволим себе минуту фантазии: не станет ли тогда человек новым космическим существом, свободным от земных ограничений? Есть ли абсолютная граница могущества и сложности для человека и его творений, абсолютная граница могущества и сложности для саморазвивающихся систем вообще?... Впрочем, это вопросы для науки будущего, на которые она сумеет ответить лучше нас» [6, с. 26–27].

Список литературы: 1. *Шенк Р., Хантер Л.* Познать механизмы мышления//Реальность и прогнозы искусственного интеллекта. М.: Мир, 1987. 287 с. 2. *Кендрью Д.* Нить жизни. М.: Мир, 1968. 123 с. 3. *Гейтс Б.* Бизнес со скоростью мысли. 2-е изд. М.: ЭКСМО, 2004. 478 с. 4. *Newell A., Shaw I.C., Simon H.A.* Empirical explorations with the logic theory machine. Proceedings of the Western Joint Computer Conference. 1957. P. 218–239. 5. *Хьюбел Д.* Глаз, мозг, зрение. М.: Мир, 1990. 239 с. 6. *Винер Н.* Кибернетика. 2-е изд. М.: Сов. радио, 1968. 326 с.

Поступила в редколлегию 24. 11. 2004

УДК 519.7

М.Ф. Бондаренко, Ю.П. Шабанов-Кушнаренко

ОБ АЛГЕБРЕ ПРЕДИКАТОВ

В статье дана характеристика алгебры предикатов как эффективного средства формального описания объектов бионики интеллекта.

1. Отношения

Статья является продолжением работы «О бионике интеллекта» (публикуемой в этом же номере журнала), в которой понятие предиката существенно использовалось при общей характеристике бионики интеллекта. Предикаты – это основной математический инструмент, предназначенный для формального описания объектов бионики интеллекта. Алгебру предикатов можно рассматривать как аналог школьной алгебры в логической математике (школьную же алгебру можно отнести к числовой математике). Язык алгебры предикатов представляет собой универсальное средство формального описания любых механизмов интеллекта человека и машин. Алгебра предикатов – это первая ступень формального языка для разработчиков, проектирующих средства искусственного интеллекта. Второй ступенью является алгебра предикатных операций, которую можно рассматривать как логический аналог изучаемой в вузах высшей математики (последнюю тоже можно отнести к области числовой математики). На языке алгебры предикатов выражаются любые отношения, а на языке алгебры предикатных операций – любые действия над отношениями.

Отношения выражают свойства предметов и связи между ними. Они представляют собой универсальное средство формального описания любых объектов. За две с половиной тысячи лет науке не удалось обнаружить в мире ни одного объекта, о котором можно было бы с уверенностью сказать, что он в принципе не поддается формальному описанию с помощью отношений. Никакие другие известные средства формального описания объектов (например школьная алгебра, дифференциальное и интегральное исчисление) свойством универсальности не обладают. Естественный язык, представляющий собой универсальное средство духовного общения людей, можно рассматривать как механизм для выражения отношений. Обращаясь с речью к другим людям, мы передаем им вполне определенный смысл произносимого предложения, который есть ничто 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 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.

Научное определение понятия отношения сформировал в первой половине XVII века Декарт, опираясь на введенные им понятия переменной и координатной системы. С помощью последних можно ввести понятие предметного пространства. Отношения задаются над предметным пространством. Поэтому прежде мы рассмотрим понятие предметного пространства. Входные сигналы, предъявляемые испытуемому в экспериментах, относящихся к теории T, называются *предметами* теории T. К примеру, в теории цветового зрения человека (которая была рассмотрена в упомянутой выше работе «О бионике интеллекта») предметами служат световые излучения. Непустое множество U всевозможных предметов теории T называется *универсумом предметов* теории T.

Примерами универсума предметов могут служить множества {1, 2, 3}, {α, β, γ, δ}, множество натуральных чисел; множество всех действительных чисел, множество всех n-мерных числовых векторов (например совокупность всех линейчатых спектров световых излучений, каждый из которых состоит из n линий); множество всех действительных функций на отрезке $[\lambda_1,\lambda_2]$ числовой оси (например совокупность всех сплошных спектров световых излучений, каждый из которых имеет вид некоторой кривой). К примеру, в теории цветового зрения в роли универсума U выступает множество всех световых излучений. В принципе, и пустое множество можно взять в роли универсума предметов. Однако в этом случае получается *пустая теория* – в прямом и переносном смысле этого слова. Такая теория бесполезна. Это теория ни о чем. В пустой теории будет истинным любое утверждение.

Введем в рассмотрение *предметные переменные* $x_1, x_2, \ldots, x_m$ теории T. Здесь символом m обозначено число предметных переменных. Содержательно предметные переменные можно проинтерпретировать как *места* предметов, используемые в экспериментах теории T. Предметы, занимающие определенные места, называются еще иначе *состо-*

яниями этих мест. Если предмет a находится на месте $x_i (i=\overline{1,m})$, то говорят, что переменная x_i *принимает значение* a, и пишут $x_i = a$. Приведем примеры мест и их состояний: температура человека в разных точках его тела; цвет зрительного ощущения в разных точках поля зрения: скорость автомобиля в различные моменты времени; спектральная плотность лучистой яркости светового излучения в разных точках его спектра (то есть при разных длинах волн электромагнитных колебаний). Символ m интерпретируем как число всех мест, используемых в теории T. К примеру, в теории цветового зрения местами для световых излучений служат поля сравнения x_1, x_2. Их всего два, поэтому в данном случае $m=2$.

Непустое множество V всевозможных переменных теории T называется *универсумом переменных* теории T. Обычно универсум переменных берут конечным. Можно вводить также и бесконечные множества переменных, но в этом обычно не возникает необходимости, поскольку такие множества всегда можно выразить при помощи конечного числа переменных. Например континуальное множество переменных можно выразить одной функцией $y=f(x)$, для построения которой используются всего две числовые переменные x и y. В результате получаем бесконечное семейство $\{y_x\}$ переменных y_x с параметром $x \in R$ (R – множество действительных чисел). Поэтому мы ограничиваемся введением конечного универсума предметных переменных $V=\{x_1, x_2, \ldots, x_m\}$. К примеру, для теории цветового зрения универсумом переменных будет множество $V=\{x_1, x_2\}$, где предметные переменные x_1 и x_2 содержательно интерпретируются как поля сравнения в эксперименте, основанном на методе нулевого прибора.

Иногда вводят не одну, а сразу много теорий. В этом возникает необходимость, когда изучают связи между различными теориями. Бывает, что несколько теорий приходится соединять в единую теорию, в этом случае множества предметных переменных отдельных теорий объединяют в единое множество переменных новой теории. Если в теории в процессе ее развития вводятся новые предметные переменные (в дополнение к тем, которые в ней были введены ранее), то это, строго говоря, требует перехода к другой теории. Множество предметных переменных – это одно из характеристических свойств теории; если оно изменяется, то при этом изменяется и сама теория.

Если известно, что $x_1 = a_1, x_2 = a_2, \ldots, x_m = a_m$, то говорят о наличии *набора* $(a_1, a_2, \ldots, a_m)$ предметов $a_1, a_2, \ldots, a_m$, находящихся на местах $x_1, x_2, \ldots, x_m$. Примерами наборов могут служить: 1) линейчатый или сплошной спектры светового излучения (каждая линия спектра находится на своем месте; в случае линейчатого спектра место линии представлено ее номером; сплошной спектр содержит в своем составе бесконечное число линий, место каждой линии в нем представлено своим действительным числом); 2) пара цветов, сравниваемых испытуемым на полях сравнения по методу нулевого прибора; 3) цветовая картина, возникающая перед сознанием человека в его поле зрения, каждая точка которого характеризуется своим цветом; 4) напряженность электрического поля в различных точках пространства; 5) температура в определенной точке некоторого тела в различные моменты времени. Вместо того, чтобы говорить, что на месте x расположен предмет a, употребляют также выражение: «место x находится в состоянии a». Как внешний (объективный), так и внутренний (субъективный) миры человека устроены единообразно: всюду мы обнаруживаем места, находящиеся в определенных состояниях. С течением времени состояния мест могут меняться.

Места и их состояния обладают следующими характерными свойствами. Не может так случиться, чтобы некоторое место в какой-то момент времени не находилось ни в каком состоянии, иначе говоря, чтобы на заданном месте не присутствовал какой-либо предмет. Еще в V в. до н. э. древнегреческий философ Парменид обнаружил, что понятие места, не имеющего никакого состояния, логически противоречиво. Это положение он выразил следующими словами: «Вы говорите, что пустота *есть*. Следовательно, пустота – не ничто. Следовательно она – не пустота». Не бывает и такого, чтобы сразу несколько предметов одновременно занимали одно и то же место. Если какой-то предмет занимает некоторое место, то никакой другой предмет в данный момент времени там находиться уже не может. С первого взгляда такое утверждение кажется ложным. В самом деле, мы знаем, например, что одна и та же точка тела может одновременно испускать монохроматические световые лучи разной длины волны. Это наводит на мысль о том, что одно и то же место в какой-то момент времени может одновременно находиться во многих состояниях. Но, вникнув в суть сказанного, мы обнаруживаем, что у каждого такого светового луча имеется свое место на шкале длин волн.

Чтобы избежать противоречий, все состояния, обнаруженные якобы на одном и том же месте, приходится объединять в одно многокомпонентное состояние. Исходные же состояния теперь становятся компонентами некоторого векторного состояния, причем для каждого компонента такого вектора всегда найдется свое место. Ведь если предметы, которые кажутся расположенными на одном и том же месте, отличны друг от друга, то всегда найдется различающий их признак. Тогда ничто не сможет помешать нам принять значения этого признака в качестве мест для данных предметов. Так,

например, отдельные спектральные линии в световом излучении объединяют в одно сложное (векторное) состояние, называемое спектром. В роли мест для спектральных линий можно принять длины волн соответствующих монохроматических световых излучений. Термин «место», таким образом, иногда приходится понимать в обобщенном смысле. Не обязательно место понимать как точку в трехмерном физическом пространстве. Местом, к примеру, может быть некоторый момент времени, отметка на шкале частот, порядковый номер, одно из полей шахматной доски.

Понятие набора не совпадает с понятием множества. Например множество $\{a,b\}$ не изменится, если в его записи поменять местами предметы a и b; однако набор (a,b) после выполнения той же операции становится иным. Не совпадает понятие набора и с понятием упорядоченного множества, то есть такого множества, на котором введено какое-нибудь отношение порядка. К примеру, множество чисел «1,2,3», упорядоченное отношением $<$, после перестановки в его записи символов 1 и 2, уже не будет упорядоченным относительно отношения $<$. Набор же предметов (1,2,3), после выполнения той же операции, останется набором, хотя и иным. Ближе всего к понятию набора подходит математическое понятие последовательности, которое определяется как функция с произвольно выбираемым множеством значений, заданная на множестве натуральных чисел. Понятие набора обобщает понятие последовательности. Его можно определить как функцию, отображающую произвольно выбранное множество V элементов, называемых местами, в произвольно выбранное множество U элементов, называемых предметами (или состояниями). Можно сказать и иначе: набор есть индексированное множество предметов $\{y_x\}$ $(x \in V, y \in U)$.

Если $a_1,a_2,\ldots,a_m \in U$ и $x_1 = a_1, x_2 = a_2, \ldots, x_m = a_m$, то пишут $(a_1,a_2,\ldots,a_m) \in U^m$ и говорят, что *набор* (иначе– *вектор*, *кортеж*) $(a_1,a_2,\ldots,a_m)$ принадлежит *предметному пространству* U^m. Число m называется *длиной* набора (или *размерностью* вектора). Запись U^m читается «U в (декартовой) степени m». Если множество V выбрано конечным, то предметное пространство называется *конечномерным*, в противном случае – *бесконечномерным*. Часто сами переменные используют в роли их значений. В этом случае вместо $(a_1,a_2,\ldots,a_m)$ пишут $(x_1,x_2,\ldots,x_m)$. Что именно имеется в виду в каждом конкретном случае такой записи – переменные или их значения, определяется читателем по контексту. Пространство U^m теории T образовано из всевозможных наборов $(x_1,x_2,\ldots,x_m)$, каждый из которых составлен из предметов $x_1,x_2,\ldots,x_m$ теории T. Число m называется *размерностью пространства* U^m. Оно должно совпадать с числом всех мест, используемых в экспериментах теории T. Любое подмножество P пространства U^m называется m-местным *отношением*, заданным над U^m. Примером отношения в теории цветового зрения может служить множество пар равноцветных световых излучений.

Понятие отношения можно обобщить. Сделать это можно следующим образом. Произвольно выберем m каких-нибудь непустых необязательно различных подмножеств $A_1,A_2,\ldots,A_m$ универсума U. Образуем множество S всевозможных наборов $(x_1,x_2,\ldots,x_m)$, составленных из предметов $x_1 \in A_1, x_2 \in A_2, \ldots, x_m \in A_m$. Так определенное множество S называется *декартовым произведением* множеств $A_1,A_2,\ldots,A_m$. Оно записывается в виде $S = A_1 \times A_2 \times \ldots \times A_m$. Множество S называется также *подпространством* предметного пространства U^m. *Множества* $A_1,A_2,\ldots,A_m$ называются *координатными осями* подпространства S. Отношения можно задавать не только над всем пространством U^m, но также и над любым из его подпространств. В этом случае набор множеств $A_1,A_2,\ldots,A_m$ определяет *тип отношения*. Отношение типа $(A_1,A_2,\ldots,A_m)$ определяется как подмножество декартова произведения $A_1 \times A_2 \times \ldots \times A_m$. Отношения, заданные на одном и том же декартовом произведении, называются *однотипными*. Если тип отношения определен, то конкретный универсум U пространства можно не указывать. Достаточно довольствоваться возможностью его введения. В роли универсума можно использовать любое множество, охватывающее объединение множеств $A_1,A_2,\ldots,A_m$. В частном случае, принимая $A_1 = A_2 = \ldots = A_m = U$, приходим к первоначально введенному нами понятию предметного пространства.

Рассмотрим пример отношения. Пусть $U = \{1,2,3,4,5,6\}$, $V = \{x_1,x_2\}$, $m = 2$, $A_1 = \{1,2,3,4\}$, $A_2 = \{3,4,5,6\}$. Подпространство $S = A_1 \times A_2$ есть множество всех пар вида (x_1,x_2), удовлетворяющих условию $x_1 \in A_1, x_2 \in A_2$: $S = \{(1,3), (1,4), (1,5), (1,6), (2,3), (2,4), (2,5), (2,6), (3,3), (3,4), (3,5), (3,6), (4,3), (4,4), (4,5), (4,6)\}$. Декартово произведение S состоит из $4 \times 4 = 16$ двухкомпонентных векторов. В роли отношения P берем какое-нибудь одно из его подмножеств, например, $P = \{(1,6), (2,4), (3,3), (4,3), (4,4)\}$. Всего в подпространстве S можно образовать $2^{16} = 65536$ отношений. Отношение $\varnothing$, не содержащее ни одного вектора, называется *пустым*, отношение; образованное из всевозможных векторов подпространства S, – *полным*.

Остановимся особо на одноместных и двуместных отношениях. Если в записи одноместного отношения убрать круглые скобки, то получится запись соответствующего ему множества, например, из отношения $\{(1),(3),(4)\}$ получаем множество $\{1,3,4\}$. Такие отношения и множества взаимно однозначно связаны друг с другом, поэтому их можно не различать. В дальнейшем, для краткос-

БІБЛІОТЕКА
ХНУРЕ
ІНВ. №

ти, вместо одноместных отношений будем записывать соответствующие им множества. Двуместные отношения удобно наглядно представлять в виде двудольных графов. Ниже для примера на рис. 1 изображен двудольный граф отношения $P=\{(1,6), (2,4), (3,3), (4,3), (4,4)\}$. Каждой паре, входящей в состав отношения, соответствует свое ребро графа, которое соединяет его вершины, помеченные именами компонентов этой пары.

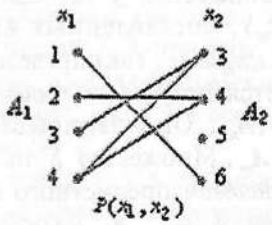


Рис. 1

Это же отношение можно представить в виде графиков (рис. 2, 3) и таблиц (табл. 1, 2) различного вида.

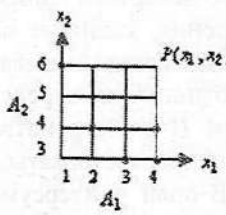


Рис. 2

Таблица 1

x_1	1	2	3	4	4
x_2	6	4	3	3	4

$P(x_1, x_2)$

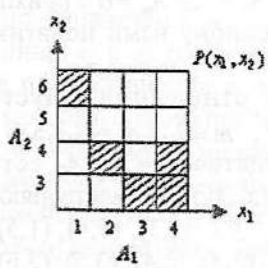


Рис. 3

Таблица 2

x_1	x_2
1	6
2	4
3	3
4	3
4	4

Для выражения m-местных отношений используются m-мерные графики и таблицы с m строками или m столбцами.

Любое отношение можно содержательно проинтерпретировать как знание о факте, выраженное некоторым высказыванием. *Факт* – это однозначная характеристика набора наблюдаемых состояний заданной совокупности мест. *Знание* же о факте не всегда обладает такой однозначностью, оно задает лишь некоторое множество возможных вариантов наборов состояний мест, то есть какое-то отношение. Факт однозначно характеризуется множеством, состоящим из одного набора состояний мест. Ничем иным факты и знания быть не могут. Проинтерпретируем, к примеру, этим способом записанное выше отношение P. Заданы два места x_1 и x_2. Изначально известно, что состояния места x_1 не могут выходить за пределы множества {1,2,3,4}, а состояния места x_2 – за пределы множества {3,4,5,6}. Отсюда следует, что с самого начала число возможных наборов состояний ограничено 16-ю парами. Предположим, что факт (то есть пара состояний мест x_1 и x_2, имеющих место в действительности) наблюдателю неизвестны. Объективно же факт всегда характеризуется какой-то вполне определенной парой состояний, например, парой (4,3). Это означает, что место x_1 находится в состоянии 4, а место x_2 – в состоянии 3. Но наблюдатель этого не знает, он располагает лишь знанием о факте, выраженным отношением $P=\{(1,6), (2,4), (3,3), (4,3), (4,4)\}$, которое сужает множество возможных пар состояний с 16 до 5. Опираясь на это знание, наблюдатель заключает, что пары состояний (1,3), (1,4), (1,5), (2,3), (2,5), (2,6), (3,4), (3,5), (3,6), (4,5), (4,6) не имеют места в действительности. Эту информацию он может использовать в своих рассуждениях при выработке каких-либо решений.

Высказывание о факте может быть истинным или ложным. Оно *истинно*, если характеризующее его отношение содержит в себе факт, то есть набор состояний мест, имеющий место в действительности, и *ложно* в противном случае. Например, если факт характеризуется парой (4,3), то высказывание, соответствующее отношению P, истинно, поскольку $(4,3)\in\{(1,6), (2,4), (3,3), (4,3), (4,4)\}$. Высказывание же, соответствующее отношению $\{(1,6), (2,4)\}$, ложно, поскольку $(4,3)\notin\{(1,6), (2,4)\}$. Высказывание, которому соответствует пустое отношение, *противоречиво*, поскольку невозможно, чтобы места $x_1, x_2,\ldots,x_m$ не находились ни в каких состояниях. Высказывание, отвечающее полному отношению S, образованному из всевозможных наборов состояний, *тавтологично* (бессодержательно), так как не несет никакой новой информации о факте: оно никак не ограничивает изначально заданное множество из 16-ти наборов состояний. Любое высказывание, которому соответствует непустое отношение, *выполнимо*, поскольку для него всегда найдется такой фактический набор состояний, для которого оно будет истинным.

Высказывание B *выводимо* (следует) из высказывания A, если отношение P_A, соответствующее высказыванию A, включено в отношение P_B, соответствующее высказыванию B. Пусть, к примеру, $P_A=\{(1,6), (2,4)\}$, $P_B=\{(1,6), (2,4), (3,3), (4,3), (4,4)\}$. Мы видим, что отношение P_A включено в отношение P_B: $\{(1,6), (2,4)\}\subseteq\{(1,6), (2,4), (3,3), (4,3), (4,4)\}$. Отсюда вытекает, что высказывание B следует из высказывания A. В этом случае пишут $A\Rightarrow B$. Если же $P_{A'}=\{(1,6), (4,5)\}$, то $P_{A'}$ не включено в P_B, а значит, из A' не следует B. Про-

тиворечивое высказывание нельзя вывести ни из одного выполнимого высказывания, тавтологичное высказывание следует из любого высказывания, из противоречивого высказывания выводимо любое высказывание.

Высказывания выражаются предложениями, которые бывают утвердительными, вопросительными и повелительными. *Утвердительное предложение* выражает высказывание, которое несет в себе знание о некотором факте. Это знание характеризуется вполне определенным отношением. *Вопросительные предложения* бывают двух родов. *Вопросительное предложение первого рода* требует двоичного ответа типа «да» или «нет», например, «Идет дождь? – Да». Чтобы дать на него правильный ответ, надо преобразовать вопросительное предложение в утвердительное (в нашем примере формируем утвердительное предложение «Идет дождь»), извлечь из него содержащееся в нем отношение и проверить, охватывает ли это отношение фактический набор состояний. К примеру, если задан вопрос, соответствует ли отношение P факту (4,3), то следует ответить утвердительно, поскольку $(4,3) \in \{(1,6), (2,4), (3,3), (4,3), (4,4)\}$. Отношение же {(1,6), (2,4)} факту (4,3) не соответствует, поскольку $(4,3) \notin \{(1,6), (2,4)\}$.

Вопросительное предложение второго рода требует развернутого ответа, например, «Который час? – Без четверти два». Чтобы дать на него правильный ответ, надо сформировать какое-нибудь отношение, охватывающее фактический набор состояний. Если, к примеру, факт характеризуется парой (4,3), то ответ, выражающий отношение {(1,6), (4,3)}, будет правильным, а ответ, выражающий отношение {(1,6), (2,4)}, – неправильным. Чем меньше в сформированном отношении содержится наборов состояний, тем более содержательным будет ответ. Ответ, который соответствует отношению, состоящему из единственного набора, будет полностью характеризовать фактические состояния всех мест. Такой ответ называется *однозначным* (исчерпывающим). В нашем примере отношение {(1,6), (4,3)} характеризует факт (4,3) более полно, чем отношение {(1,6), (2,4), (3,3), (4,3), (4,4)}. Отношение же {(4,3)} несет в себе исчерпывающую информацию о факте (4,3).

Повелительное предложение, в отличие от вопросительного, требует не слов, а действий, например «Решайте задачу!». Чтобы выполнить повеление, тот, к кому оно обращено, должен сформировать состояния указанных ему мест так, чтобы содержащееся в повелении отношение стало соответствовать действительному положению дела. Чтобы выполнить повеление «Решайте задачу!», надо найти такое решение, которое удовлетворяло бы условиям поставленной задачи. Если, к примеру, задано отношение {(1,6), (4,3)}, то действия, приводящие к фактическому положению, которое характеризуется парой (1,6) (или парой (4,3)), будут правильными, действия же, формирующие пару (2,4), будут неправильными.

Рассмотренное нами исходное понятие отношения может показаться случайно выбранной и ничем не примечательной математической конструкцией. Понятие отношения с первого взгляда воспринимается как рядовое, примитивное и неказистое, тем не менее, его роль в науке совершенно исключительна. Потребовался гений Декарта, чтобы понятие отношения выделить из числа многих ему подобных и увидеть в нем основание всей науки и даже всего мироздания. Все, что находится вне и внутри нас, может быть выражено отношениями. Это обусловлено тем, что мир основан на очень простых и единообразных принципах логики.

Отношения, являющиеся частями одного и того же пространства, однотипны. Над однотипными отношениями можно выполнять операции объединения, пересечения и дополнения. *Объединением отношений* P и Q называется отношение $P \cup Q$, образованное из всех наборов отношения P и всех наборов отношения Q. Например $\{(1,6), (2,4)\} \cup \{(1,6), (4,3)\} = \{(1,6), (2,4), (4,3)\}$. *Пересечением отношений* P и Q называется отношение $P \cap Q$, образованное из всех тех наборов, которые содержаться как в отношении P, так и в отношении Q. Например $\{(1,6), (2,4)\} \cap \{(1,6) (4,3)\} = \{(1,6)\}$. *Дополнением отношения* P называется отношение $\sim P = \tilde{P}$, образованное из всех наборов пространства , за исключением тех наборов, которые присутствуют в отношении . Например $\sim\{(1,6), (2,4), (3,3), (4,3), (4,4)\}\{(1,3), (1,4), (1,5), (2,3), (2,5), (2,6), (3,4), (3,5), (3,6), (4,5), (4,6)\}$.

Объединению $P_A \cup P_B$ отношений $\neg A = \bar{A}$ и P_B соответствует высказывание $A \vee B$, называемое *дизъюнкцией высказываний* A и B. Оно образуется соединением исходных высказываний союзом «или». Например дизъюнкцией высказываний «Идет дождь» и «Светит солнце» будет высказывание «Идет дождь или светит солнце». Пересечению $P_A \cap P_B$ отношений P_A и P_B соответствует высказывание $A \wedge B$, называемое *конъюнкцией высказываний* A и B. Оно образуется соединением исходных высказываний союзом «и». Например конъюнкцией тех же высказываний будет высказывание «Идет дождь и светит солнце». Дополнению $\sim P_A$ отношения P_A соответствует высказывание $\neg A = \bar{A}$, называемое *отрицанием* высказывания A. Оно образуется присоединением к исходному высказыванию частицы «не». Например отрицанием высказывания «Идет дождь» будет высказывание «Не идет дождь». Это же высказывание можно выразить предложением «Ложно, что идет дождь».

2. Предикаты

Важно располагать математическими средствами записи отношений с помощью формул. Как показывает опыт развития науки и техники, нет более удобного и практичного средства описания объектов, чем формулы. Формулы не только дают названия объектам, но и выражают их глубинные свойства и структуру. Вместо того, чтобы ставить опыты над реальными объектами, можно «поэкспериментировать» с формулами, описывающими эти объекты, и получить все интересующие нас сведения о них. Отношения, выраженные формулами, можно «оживить» при помощи ЭВМ, которая будет воспроизводить поведение описываемых ими объектов. При этом, как можно надеяться, у вычислительной машины появятся мысли, соответствующие этим отношениям, и она приобретет способность их обрабатывать, то есть мыслить. Обращаясь к опыту математики, можно заметить, что формулами всегда выражаются только функции. Видимо, никакие другие математические объекты не поддаются формульному описанию. Но отношение – это не функция, а нечто более общее. Известны различные способы выражения отношений: множествами наборов предметов, графами, графиками, таблицами. Но среди них нет ни одного формульного.

Известен такой метод: если не представляется возможным решить какую-то задачу, то ее заменяют другой взаимно однозначно связанной с нею задачей, которая поддается решению. Затем переводят полученное решение на язык первой задачи. В результате получают решение исходной задачи. Мы применим этот метод для отыскания способа формульной записи отношений. Называется он *методом перевода*. Возьмем какое-нибудь отношение P_A и перейдем к соответствующему ему высказыванию A. Это высказывание будет истинным относительно одних фактов и ложным по отношению к другим. Например высказывание, соответствующее отношению $P=\{(1,6), (2,4), (3,3), (4,3), (4,4)\}$, будет истинным для факта (4,3), поскольку $(4,3)\in P$, и ложным относительно факта (2,3), так как $(2,3)\notin P$. Факт истинности высказывания будем выражать символом 1, а факт его ложности – символом 0. Символ 1 называется *истиной*, символ 0 – *ложью*. Действуя так, мы приходим к функции $P(x_1,x_2)$ с двоичными значениями 0 и 1, определенной на декартовом произведении

$$A_1\times A_2=\{1,2,3,4\}\times\{3,4,5,6\}.$$

Ниже приводится табл. 3 этой функции.

Функции такого типа называются предикатами. Сформулируем общее определение понятия предиката. *Предикатом*, заданным на декартовом произведении $A_1\times A_2\times\ldots\times A_m$, называется любая функция $P(x_1,x_2,\ldots,x_m)=\xi$, отображающая декартово произведение $A_1\times A_2\times\ldots\times A_m$ множеств $A_1,A_2,\ldots,A_m$ во множество $\Sigma=\{0,1\}$. Символы 0 и 1 называются *булевыми элементами*, Σ – множество всех булевых элементов. Переменная $\xi\in\{0,1\}$, являющаяся значением предиката P, называется *булевой*. Предикат $P(x_1,x_2,\ldots,x_m)$, в отличие от соответствующего ему отношения P, есть функция, поэтому появляется надежда, что его удастся выразить в виде формулы. Между отношениями и предикатами существует взаимно однозначное соответствие. Поэтому, имея формулу какого-либо предиката, мы всегда сможем отнести ее и к соответствующему отношению. Предикат $P(x_1,x_2,\ldots,x_m)$ на $A_1\times A_2\times\ldots\times A_m$ называется *конечным*, если все множества $A_1,A_2,\ldots,A_m$ конечны, и *бесконечным* – в противном случае. Эта же терминология переносится и на соответствующие предикатам отношения. Переменные $x_1,x_2,\ldots,x_m$ называются *аргументами предиката P*.

Таблица 3

		x_2 / A_2			
		3	4	5	6
x_1 / A_1	1	0	0	0	1
	2	0	1	0	0
	3	1	0	0	0
	4	1	1	0	0

$P(x_1,x_2)$

Запишем правила, позволяющие переходить от отношений к соответствующим им предикатам и обратно. Пусть L – множество всех отношений над S, M – множество всех предикатов на S. Отношение P из L и предикат P из M называются *соответствующими* друг другу, если при любых $x_1\in A_1, x_2\in A_2,\ldots,x_m\in A_m$:

$$P(x_1,x_2,\ldots,x_m)=\begin{cases}1, \textit{если } (x_1,x_2,\ldots,x_m)\in P,\\ 0, \textit{если } (x_1,x_2,\ldots,x_m)\notin P.\end{cases} \quad (1)$$

Зная отношение P, можно всегда с помощью правила (1) отыскать все значения предиката P, то есть дать полную его характеристику (например, представить предикат P в виде таблицы). Обратный переход от предиката P к отношению P можно осуществить с помощью следующего правила:

если $P(x_1,x_2,\ldots,x_m)=1$, то $(x_1,x_2,\ldots,x_m)\in P$;

если $P(x_1,x_2,\ldots,x_m)=0$, то $(x_1,x_2,\ldots,x_m)\notin P$. (2)

Правило (2) позволяет по известным булевым (двоичным) значениям предиката P найти все наборы предметов, принадлежащие отношению P. Правила (1) и (2) устанавливают взаимно однозначное соответствие между всеми отношениями множества L и всеми предикатами множества M, заданными на S. Множество всех векторов $(x_1,x_2,\ldots,x_m)$, удовлетворяющих уравнению

$P(x_1,x_2,...,x_m)=1$, образует отношение P, которое называется *областью истинности предиката P*. Предикат $P\in M$, определяемый правилом (1), называется *характеристической функцией отношения* $P\in L$. В дальнейшем с целью упрощения символики предикат будем обозначать знаком точно так же, как и соответствующее ему отношение. Из контекста всегда будет легко установить, к чему относится символ – к предикату или отношению.

Любое уравнение, встречающееся в математике, есть отношение. Возьмем, к примеру, уравнение $x+y=3$, где x и y заданы на множестве $A=\{0,1,2,3\}$, и отыщем соответствующее ему отношение P. Имеем: $P=\{(0,3),(1,2),(2,1),(3,0)\}$. Переходим к табл. 4 соответствующего предиката P. Для этого во всех ячейках таблицы с координатами x, y проставляем 1, если $(x,y)\in P$, и 0, если $(x,y)\notin P$.

Таблица 4

x \ y	0	1	2	3
0	0	0	0	1
1	0	0	1	0
2	0	1	0	0
3	1	0	0	0

$P(x,y)$

От таблицы предиката P, если потребуется, можем перейти назад к отношению P, которое теперь для разнообразия представляем в нашем примере в виде двудольного графа. Делаем это, поочередно обходя все ячейки таблицы с единицами. По координатам x, y каждой ячейки проводим ребро графа, соединяющее его вершины, помеченные символами x и y. Выполнив это, получаем граф, изображающий отношение (рис. 4).

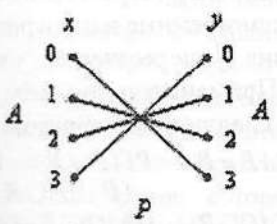


Рис. 4

Над булевыми элементами можно выполнять операции дизъюнкции, конъюнкции и отрицания. *Дизъюнкцией булевых элементов* ξ и η называется операция $\xi\vee\eta$, отображающая $\Sigma\times\Sigma$ в Σ, значения которой определяются равенствами $0\vee0=0$; $0\vee1=1\vee0=1\vee1=1$. *Конъюнкцией булевых элементов* ξ и η называется операция $\xi\wedge\eta=\xi\cdot\eta=\xi\eta$, отображающая $\Sigma\times\Sigma$ в Σ, значения которой определяются равенствами: $0\wedge0=0\wedge1=1\wedge0=0$; $1\wedge1=1$. *Отрицанием булева элемента* ξ называется операция $\neg\xi=\overline{\xi}$, отображающая Σ в Σ, значения которой определяются равенствами: $\neg0=1$, $\neg1=0$. Дизъюнкцию еще иначе называют *логическим сложением*, а конъюнкцию – *логическим умножением*.

Операции дизъюнкции, конъюнкции и отрицания подчиняются *законам идемпотентности* $\xi\vee\xi=\xi$, $\xi\xi=\xi$; *коммутативности* $\xi\vee\eta=\eta\vee\xi$, $\xi\eta=\eta\xi$; *ассоциативности* $(\xi\vee\eta)\vee\zeta=\xi\vee(\eta\vee\zeta)$, $(\xi\eta)\zeta=\xi(\eta\zeta)$; *дистрибутивности* $(\xi\vee\eta)\zeta=\xi\zeta\vee\eta\zeta$, $\xi\eta\vee\zeta=(\xi\vee\zeta)(\eta\vee\zeta)$; *элиминации (поглощения)* $\xi\vee\xi\eta=\xi$, $\xi(\xi\vee\eta)=\xi$; *свертывания* $\xi\vee\overline{\xi}\eta=\xi$, $\xi(\eta\vee\overline{\eta})=\xi$; *де Моргана* $\overline{\xi\vee\eta}=\overline{\xi}\,\overline{\eta}$, $\overline{\xi\eta}=\overline{\xi}\vee\overline{\eta}$; *двойного отрицания* $\overline{\overline{\xi}}=\xi$; *исключенного третьего* $\xi\vee\overline{\xi}=1$, *противоречия* $\xi\overline{\xi}=0$; *лжи* $\xi\vee0=\xi$, $\xi\cdot0=0$ и *истины* $\xi\cdot1=\xi$, $\xi\vee1=1$. Все законы справедливы при любых $\xi,\eta,\zeta\in\Sigma$.

Записанные равенства называются *законами алгебры булевых элементов*. Все законы, кроме закона двойного отрицания, парные. Говорят, что в каждой паре они *двойственны* друг другу. Закон же двойного отрицания *самодвойствен* (двойствен самому себе). Двойственные законы превращаются друг в друга заменой в них знака $\vee$ на знак $\wedge$, $\wedge$ на $\vee$, 0 на 1 и 1 на 0. Справедливость законов алгебры булевых элементов доказывается методом полного перебора. Например для доказательства первого закона де Моргана надо проверить выполнение следующих четырех равенств $\overline{0\vee0}=\overline{0}\cdot\overline{0}$, $\overline{0\vee1}=\overline{0}\cdot\overline{1}$, $\overline{1\vee0}=\overline{1}\cdot\overline{0}$, $\overline{1\vee1}=\overline{1}\cdot\overline{1}$.

Операции дизъюнкции, конъюнкции и отрицания можно распространить также и на предикаты. Определяются они следующими равенствами для любых $x_1\in A_1,x_2\in A_2,...,x_m\in A_m$:

$$(P\vee Q)(x_1,x_2,...,x_m)=$$

$$P(x_1,x_2,...,x_m)\vee Q(x_1,x_2,...,x_m); \qquad (3)$$

$$(P\wedge Q)(x_1,x_2,...,x_m)=$$

$$P(x_1,x_2,...,x_m)\wedge Q(x_1,x_2,...,x_m); \qquad (4)$$

$$(\neg P)(x_1,x_2,...,x_m)=\neg(P(x_1,x_2,...,x_m)). \qquad (5)$$

Символы $\vee$, $\wedge$, $\neg$, стоящие в (3)–(5) слева от знаков равенства, обозначают операции над предикатами; справа эти же символы обозначают операции над значениями предикатов, то есть над булевыми элементами. Согласно определению (3) предикат $P\vee Q$ можно получить, проставляя в каждой ячейке его таблицы дизъюнкцию значений предикатов P и Q при тех же наборах значений аргументов, то есть образуя дизъюнкцию логических элементов, содержащихся в одинаково расположенных ячейках таблиц предикатов P и Q. Аналогично заполняются таблицы конъюнкции $P\wedge Q$ предикатов P и Q и отрицания $\overline{P}$ предиката P. Ниже даны примеры построения таблиц

дизъюнкции, конъюнкции и отрицания предикатов (табл. 5–9).

Таблица 5

x \ y	2	3	4	5
0	1	0	0	1
1	1	1	0	0
2	0	1	0	0
3	1	0	0	1

$P(x,y)$

Таблица 6

x \ y	2	3	4	5
0	0	0	0	1
1	0	1	1	0
2	0	1	0	1
3	0	0	0	1

$Q(x,y)$

Таблица 7

x \ y	2	3	4	5
0	1	0	0	1
1	1	1	1	0
2	0	1	0	1
3	1	0	0	1

$(P \vee Q)(x,y)$

Таблица 8

x \ y	2	3	4	5
0	0	0	0	1
1	0	1	0	0
2	0	1	0	0
3	0	0	0	1

$(P \wedge Q)(x,y)$

Таблица 9

x \ y	2	3	4	5
0	0	1	1	0
1	0	0	1	1
2	1	0	1	1
3	0	1	1	0

$\bar{P}(x,y)$

Предикат, во всех ячейках таблицы которого стоят нули, называется *тождественно ложным* и обозначается символом 0. Если же во всех ячейках таблицы предиката стоят единицы, то такой предикат называется *тождественно истинным*. Он обозначается символом 1. Тождественно ложному предикату соответствует пустое отношение, тождественно истинному – полное.

Операции $\cup$, $\cap$, $\sim$ над отношениями из L и операции $\vee$, $\wedge$, $\neg$ над предикатами из M называются *соответствующими* друг другу. Если ими действовать на отношения и предикаты, соответствующие друг другу, то в результате будут получаться отношения и предикаты, также соответствующие друг другу. Образуем, к примеру, по предикатам P и Q, заданным вышеприведенными таблицами, соответствующие им отношения $P=\{(0,2),(0,5),(1,2),(1,3),(2,3),(3,2),(3,5)\}$, $Q=\{(0,5),(1,3),(1,4),(2,3),(2,5),(3,5)\}$ и найдем их объединение и пересечение, а также дополнение отношения P: $P \cup Q=\{(0,2),(0,5),(1,2),(1,3),(1,4),(2,3),(2,5),(3,2),(3,5)\}$, $P \cap Q=\{(0,5),(1,3),(2,3),(3,5)\}$, $\sim P=\{(0,3),(0,4),(1,4),(1,5),(2,2),(2,4),(2,5),(3,3),(3,4)\}$. Переходя от полученных отношений к соответствующим им предикатам, получаем совпадение с предикатами, указанными в таблицах.

С помощью равенств (3)–(5) законы для операций дизъюнкции, конъюнкции и отрицания, введенные нами ранее для булевых элементов, легко переносятся на одноименные операции над предикатами. Например

$$\overline{(P \vee Q)}(x_1,x_2,\ldots,x_m)=\neg\ (P \vee Q)(x_1,x_2,\ldots,x_m)=$$

$$=\neg(P(x_1,x_2,\ldots,x_m) \vee Q(x_1,x_2,\ldots,x_m))=$$

$$=\neg P(x_1,x_2,\ldots,x_m) \wedge \neg Q(x_1,x_2,\ldots,x_m)=$$

$$=(\bar{P})(x_1,x_2,\ldots,x_m) \wedge (\bar{Q})(x_1,x_2,\ldots,x_m)=$$

$$=(\bar{P} \wedge \bar{Q})(x_1,x_2,\ldots,x_m)$$

для любых $x_1 \in A_1, x_2 \in A_2, \ldots, x_m \in A_m$. Следовательно, $\overline{P \vee Q}=\bar{P} \wedge \bar{Q}$. Итак, для любых $P,Q,R \in M$ $P \vee P=P$, $PP=P$; $P \vee Q=Q \vee P$, $PQ=QP$; $(P \vee Q) \vee R=P \vee (Q \vee R)$, $(PQ)R=P(QR)$; $(P \vee Q)R=PR \vee QR$, $PQ \vee R=(P \vee R)(Q \vee R)$; $P \vee PQ=Q$, $P(P \vee Q)=P$; $P \vee Q\bar{Q}=P$, $P(Q \vee \bar{Q})=P$; $\overline{P \vee Q}=\bar{P}\bar{Q}$, $\overline{PQ}=\bar{P} \vee \bar{Q}$; $\bar{\bar{P}}=P$; $P \vee \bar{P}=1$, $P\bar{P}=0$; $P \vee 0=P$, $P \cdot 0=0$; $P \cdot 1=P$, $P \vee 1=1$. Теперь символы 0 и 1 означают не булевы элементы, а тождественно ложный и тождественно истинный предикаты. Записанные равенства называются *законами алгебры предикатов*.

Мы видим, что при переходе от булевых элементов к предикатам форма законов для операций дизъюнкции, конъюнкции и отрицания остается неизменной, хотя сами операции теперь воздействуют на объекты другой природы. Известно много разных троек операций, которые подчиняются тем же законам. Еще одним примером таких операций могут служить рассмотренные нами в разделе 1 операции объединения $\cup$, пересечения $\cap$ и дополнения $\sim$ отношений. Применительно к ним те же законы перепишутся следующим образом: для любых $P,Q,R \in L$ $P \cup P=P$, $P \cap P=P$; $P \cup Q=Q \cup P$, $P \cap Q=Q \cap P$; $(P \cup Q) \cup R=P \cup (Q \cup R)$, $(P \cap Q) \cap R=P \cap (Q \cap R)$; $(P \cup Q) \cap R=(P \cap R) \cup (Q \cap R)$, $(P \cap Q) \cup R=(P \cup R) \cap (Q \cup R)$; $P \cup (P \cap Q)=Q$, $P \cap (P \cup Q)=P$; $P \cup (Q \cup \sim Q)=P$, $P \cap (Q \cap \sim Q)=P$; $\sim(P \cup Q)=\sim P \cap \sim Q$; $\sim(P \cap Q)=\sim P \cup \sim Q$; $\sim\sim P=P$; $P \cup \sim P=S$, $P \cap \sim P=\varnothing$; $P \cup \varnothing=P$; $P \cap \varnothing=\varnothing$; $P \cap S=P$, $P \cup S=S$. Записанные равенства называются *законами алгебры отношений*.

Можно выделить все то общее, что содержится в этих трех формулировках законов. Это «общее» называется булевой алгеброй. Ниже дается ее определение. *Булевой алгеброй* называется любое фиксированное множество B хотя бы с двумя какими-нибудь элементами 0 и 1 в нем и с заданными на

этом множестве какими-либо операциями $x+y$, $x\ y$ и $-x$, называемыми *сложением*, *умножением* и *обращением элементов* $x,y\in B$, которые при любых $x,y,z\in B$ подчиняются следующим семи законам: $x+x=x$, $x+y=y+x$, $(x+y)+z=x+(y+z)$, $(x+y)z=xz+yz$, $x+y(-y)=x$, $-(x+y)=(-x)(-y)$, $-(-x)=x$. Эти законы называются *аксиомами булевой алгебры*. Остальные законы, а именно $xx=x$, $xy=yx$, $(xy)z=x(yz)$, $xy+z=(x+z)(y+z)$, $x+xy=x$, $x(x+y)=x$, $x(y+(-y))=x$, $-(xy)=(-x)+(-y)$, $x+(-x)=1$, $x(-x)=0$, $x+0=x$, $x\cdot 0=0$, $x\cdot 1=x$, $x+1=1$, в булевой алгебре также выполняются, однако в перечне аксиом их приводить нет необходимости, поскольку они логически вытекают из указанных аксиом. Определение булевой алгебры, в силу своей общности, охватывает целое семейство родственных конкретных алгебр. В роли множества B, его элементов 0 и 1, а также операций сложения, умножения и обращения элементов множества B можно брать любые объекты без каких бы то ни было ограничений. Каждый такой их выбор, который обеспечивает выполнение всех перечисленных аксиом, дает один из возможных вариантов булевой алгебры. Если в роли множества B мы возьмем множество всех n-местных булевых функций, в роли 0 и 1 – булевы элементы 0 и 1, а в роли операций +, × и – – дизъюнкцию, конъюнкцию и отрицание булевых функций, то получим *булеву алгебру булевых функций*. Если под множеством B будем понимать множество L всех отношений какого-нибудь типа, под знаками 0 и 1 – пустое и полное отношения, а под знаками +, × и – – объединение, пересечение и дополнение отношений, то приходим к *булевой алгебре отношений*. Понимая же под B множество M всех предикатов некоторого типа, под знаками 0 и 1 – тождественно ложный и тождественно истинный предикаты, а под знаками +, ×, – – дизъюнкцию, конъюнкцию и отрицание предикатов, приходим к *булевой алгебре предикатов*. При изучении интеллекта булева алгебра играет исключительно важную роль, она описывает самые глубинные механизмы интеллекта, его первооснову. Можно сказать и иначе: булева алгебра – это тот каркас, который скрепляет в единое целое здание интеллекта.

3. Алгебра предикатов

В разделе 2 мы ввели предикаты и булевы операции над ними. Теперь займемся разработкой способа формульной записи предикатов заданного типа. Тип предиката задаем, указывая множества $M=\{x_1,x_2,\ldots,x_m\}$ и $A_i=\{a_{i1},a_{i2},\ldots,a_{ik_i}\}$, $i=\overline{1,m}$, k_i – число элементов в множестве A_i. Чтобы не усложнять задачу, будем пока считать, что все множества A_i конечны. Мы хотим записать формулой произвольно выбранный конечный предикат $P(x_1,x_2,\ldots,x_m)$ заданного типа. Для построения формул нам понадобятся базисные предикаты, играющие роль исходных строительных блоков, и базисные операции, обеспечивающие соединение блоков в единую конструкцию, каковой является искомая формула. В роли базисных предикатов используем предикаты 0, 1, а также специальные предикаты, называемые *предикатами узнавания предмета a по переменной x_i* ($i=\overline{1,m}$, $a\in A_i$) и записываемые в виде x_i^a. Предикат x_i^a «узнаёт» произвольно выбранный из множества A_i предмет x_i, сравнивая его с предметом a. Если $x_i=a$, то он отождествляет x_i с предметом a, сигнализируя об этом положительным ответом $x_i^a=a^a=1$. Если же $x_i=b$ ($b\neq a$), то предикат x_i^a отличает предмет b от предмета a, сигнализируя об этом отрицательным ответом $x_i^a=b^a=0$. Мы приходим к следующему определению:

$$x_i^a=\begin{cases}1, \text{ если } x_i=a,\\ 0, \text{ если } x_i\neq a.\end{cases} \qquad (1)$$

Символ a, который стоит в показателе записи x_i^a, называется *характеристикой предиката узнавания предмета* x_i^a. Указание предмета a и значения индекса i полностью определяет предикат вида (1). В роли способов соединения выбранных нами строительных блоков (то есть предикатов 0, 1 и предикатов вида x_i^a) используем введенные нами ранее операции дизъюнкции, конъюнкции и отрицания предикатов. В результате получаем так называемую *алгебру предикатов*.

Для предикатов узнавания предмета справедливы:

закон отрицания:

$$\overline{x_i^a}=\bigvee_{\substack{b\in A_i\\ b\neq a}} x_i^b,$$

закон ложности:

$$\text{если } b\neq a, \text{ то } x_i^a x_i^b=0,$$

закон истинности:

$$\bigvee_{b\in A_i} x_i^b=1.$$

Все эти законы справедливы при любых $i=\overline{1,m}$; $x_i\in A_i$; $a,b\in A_i$.

Для примера запишем закон отрицания для предметной переменной x_3, ее области изменения $A_3=\{a_1,a_2,a_3\}$ и предметов a_1 и a_2. Имеем: $\overline{x_3^{a_1}}=x_3^{a_2}\vee x_3^{a_3}$; $\overline{x_3^{a_2}}=x_3^{a_1}\vee x_3^{a_3}$. Как видим, закон отрицания можно использовать для исключения отрицания из формул алгебры предикатов и как средство введения отрицания в формулу, когда это потребуется. Запишем, к примеру, закон ложности для предметной переменной y и предметов a и c. Имеем $y^b y^c=0$, однако по второму закону идем-

потентности получаем $y^b y^b = y^b$ и $y^c y^c = y^c$. Это означает, что при $b = c$ закон ложности не выполняется. Закон ложности выражает факт попарной различимости элементов в каждом из множеств $A_1, A_2, ..., A_m$. Наконец, запишем, к примеру, закон истинности для предметных переменных $x, y \in A$ ($A = \{a, b, c\}$). Имеем $x^a \vee x^b \vee x^c = 1$, $y^a \vee y^b \vee y^c = 1$. Закон истинности перечисляет все элементы, содержащиеся в каждом из множеств $A_1, A_2, ..., A_m$.

На языке формул алгебры предикатов можно очень просто записать любое отношение. Отдельный факт, представленный набором $(a_1, a_2, ..., a_m)$ состояний $a_1, a_2, ..., a_m$ мест $x_1, x_2, ..., x_m$, можно выразить высказыванием «$x_1 = a_1$ и $x_2 = a_2$ и ... и $x_m = a_m$», что на языке алгебры предикатов запишется формулой $x_1^{a_1}, x_2^{a_2}, ..., x_m^{a_m}$. Выражения такого вида называются *конституэнтами единицы предиката*. Произвольное знание о факте можно представить отношением $\{(a_{11}, a_{21}, ..., a_{m1}), (a_{12}, a_{22}, ..., a_{m2}), ..., (a_{1r}, a_{2r}, ..., a_{mr})\}$, где r — число наборов в отношении. Это знание выражается высказыванием «($x_1 = a_{11}$ и $x_2 = a_{21}$ и ... и $x_m = a_{m1}$) или ($x_1 = a_{12}$ и $x_2 = a_{22}$ и ... и $x_m = a_{m2}$) или ... или ($x_1 = a_{1r}$ и $x_2 = a_{2r}$ и ... и $x_m = a_{mr}$)». На языке алгебры предикатов это же знание запишется формулой

$$x_1^{a_{11}} x_2^{a_{21}} ... x_m^{a_{m1}} \vee x_1^{a_{12}} x_2^{a_{22}} ... x_m^{a_{m2}} \vee ... \vee x_1^{a_{1r}} x_2^{a_{2r}} ... x_m^{a_{mr}},$$

называемой *совершенной дизъюнктивной нормальной формой* (сокращенно *СДНФ*) *предиката*. Пустое отношение записывается формулой 0, выражающей тождественно ложный предикат. СДНФ, состоящая из всех конституэнт единицы, выражает полное отношение, которое еще иначе можно записать в виде формулы 1, обозначающей тождественно истинный предикат.

Как видим, операции $\vee, \wedge$ и $\neg$ в сочетании с предикатами 0, 1 и всевозможными предикатами узнавания предмета образуют достаточный ассортимент выразительных средств (даже с избытком, поскольку мы обошлись без операции отрицания и без предиката 1) для формульной записи любого предиката произвольного типа, а значит — и любого соответствующего ему отношения. Учитывая, что отношения — это универсальное средство описания любых объектов, мы можем заключить, что на язык формул алгебры предикатов можно перевести любые зависимости, известные в науке. Например на языке формул алгебры предикатов, в принципе, можно записать любое интегральное уравнение. Язык интегрального исчисления — это специализированный язык, язык же алгебры предикатов — универсальный. На специализированном языке формулы получаются всегда проще, чем на универсальном при выражении одного и того же содержания. В этом язык алгебры предикатов уступает конкурирующим с ним специализированным языкам. Но зато на универсальном языке (а именно таким языком является алгебра предикатов) можно записать любой закон природы, любую программу для ЭВМ, любое математическое соотношение, вообще — любую мысль. Очень важно то, что нашелся такой «всеядный» язык, на котором выражается абсолютно все, что только можно наблюдать вокруг и внутри нас. Важно и то, что сам механизм интеллекта невозможно полноценно описать, не прибегая к языку алгебры предикатов. Интеллект — средство универсальное, и он может быть охвачен только универсальным языком. Язык алгебры предикатов предельно прост и естествен, он известен уже более четверти века, но конкурентоспособный заменитель ему пока не найден. Вряд ли найдется когда-либо другой язык, равный ему по степени универсальности и превосходящий его по простоте.

Рассмотрим пример формульной записи предиката $P(x, y)$, заданного табл. 10.

Таблица 10

$Q(x, y)$

x \ y	0	1	2	3
0	0	0	0	1
1	0	0	1	0
2	0	1	1	0
3	1	0	0	0

Чтобы получить СДНФ предиката P, надо обойти все ячейки таблицы с единичными значениями и в соответствие каждой из них поставить конституэнту единицы с показателями при x и y, совпадающими с координатами этой ячейки, а затем все полученные таким путем конституэнты единицы соединить знаками дизъюнкции:

$$P(x, y) = x^0 y^3 \vee x^1 y^2 \vee x^2 y^1 \vee x^2 y^2 \vee x^3 y^0.$$

По полученной формуле можно вычислить значения предиката $P(x, y)$ при любых значениях его аргументов x и y. Например пусть $x = 0$, $y = 2$. Тогда

$$P(0,2) = 0^0 \cdot 2^3 \vee 0^1 \cdot 2^2 \vee 0^2 \cdot 2^1 \vee 0^2 \cdot 2^2 \vee 0^3 \cdot 2^0 =$$

$$= 1 \cdot 0 \vee 0 \cdot 1 \vee 0 \cdot 0 \vee 0 \cdot 1 \vee 0 \cdot 0 = 0 \vee 0 \vee 0 \vee 0 \vee 0 = 0.$$

Если же $x = 2$, $y = 1 = 1$, то

$$P(2,1) = 2^0 \cdot 1^3 \vee 2^1 \cdot 1^2 \vee 2^2 \cdot 1^1 \vee 2^2 \cdot 1^2 \vee 2^3 \cdot 1^0 =$$

$$= 0 \vee 0 \vee 1 \vee 0 \vee 0 = 1.$$

Можно подставить в формулу предиката значения не всех его аргументов. В этом случае получаем не значение предиката при заданных значениях его аргументов, а некий новый предикат с меньшим числом предметных переменных. Пусть, к примеру, $x = 2$, тогда

$$P(2,y) = 2^0 \cdot y^3 \vee 2^1 \cdot y^2 \vee 2^2 \cdot y^1 \vee 2^2 \cdot y^2 \vee 2^3 \cdot y^0 =$$

$$= 0 \cdot y^3 \vee 0 \cdot y^2 \vee 1 \cdot y^1 \vee 1 \cdot y^2 \vee 0 \cdot y^0 =$$

$$= 0 \vee 0 \vee y^1 \vee y^2 \vee 0 = y^1 \vee y^2 .$$

При производстве этих вычислений приходится столкнуться с «сюрпризом»: возникает необходимость выполнять дизъюнкцию и конъюнкцию булева элемента на предикат, например, $0 \cdot y^0$, $1 \cdot y^1$, $y^1 \vee 0$, $0 \vee y^2$. Однако такая операция нами не была предусмотрена. При вычислениях мы воспользовались «почти» очевидными правилами для такого рода операций: $0 \cdot P = P \cdot 0 = 0$, $1 \cdot P = P \cdot 1 = P$, $0 \vee P = P \vee 0 = P$, $1 \vee P = P \vee 1 = 1$. Эти правила нами не рассматривались. Здесь обнаруживается пробел в наших знаниях, а за ним – глубокая, обширная и важная новая проблема, разработка которой приводит к так называемой *логической алгебре*, напоминающей по своему строению классическую линейную алгебру. В ней булевы элементы 0 и 1 выступают в роли скаляров, а предикаты – в роли векторов. Разработка логической алгебры позволяет проникнуть в чрезвычайно важную иерархическую структуру механизма логики. Оказывается, логика имеет вид «слоеного пирога» с уходящим ввысь бесконечным числом слоев. Возьмем два соседних слоя в произвольном месте «логического пирога». Нижний из них всегда можно рассматривать как скалярную булеву алгебру, а верхний – как векторную. Связывает их друг с другом третья – «боковая», скалярно-векторная булева алгебра. Вводя ранее алгебру предикатов, мы подошли к ней упрощенно – как к однослойной. Однако под алгеброй предикатов четко просматривается еще и нижний слой – алгебра булевых элементов, без которой алгебра предикатов была бы неполноценной. Благодаря только что рассмотренному примеру, при алгебре предикатов обнаружились следы также и третьей (междуэтажной) алгебры, связывающей оба слоя. Разработка логической алгебры – большая и серьезная задача, требующая специального рассмотрения.

Тот же пример наводит и на иные размышления. Когда мы в формулу предиката $P(x,y) = x^0y^3 \vee x^1y^2 \vee x^2y^1 \vee x^2y^2 \vee x^3y^0$ подставляем $x = 2$, то в результате получаем другой предикат $y^1 \vee y^2$. Получается, что подстановка – это операция, преобразующая один предикат в другой. Действует данная операция (обозначим ее символом $x/2$) из множества M всех предикатов, заданных на двумерном пространстве $S = A_1 \times A_2$. Но в какое множество она действует? Естественно считать, что в то же самое множество M. В этом случае предикат-значение должен быть взят из того же пространства, что и предикат-аргумент. Обозначим этот предикат символом $Q(x,y) = y^1 \vee y^2$. В его формуле переменная x отсутствует. Как говорят математики, она стала фиктивной или несущественной. Ниже приведена табл. 11 предиката $Q(x,y)$.

Таблица 11

		y = 0	1	2	3
x	0	0	1	1	0
	1	0	1	1	0
	2	0	1	1	0
	3	0	1	1	0

$Q(x,y)$

Как видим, в таблице предиката Q переменная x сохранилась. Не менее естественно считать также, что полученный в результате выполнения операции подстановки предикат $y^1 \vee y^2$ имеет лишь один аргумент y. Обозначим его через $R(y) = y^1 \vee y^2$. При таком понимании операция подстановки действует из M уже в другое множество M', где M' – множество всех предикатов, заданных на одномерном пространстве $S' = A_2$. Ниже приведена табл. 12 предиката $R(y)$, получаемого в результате выполнения этой операции.

Таблица 12

y = 0	1	2	3
0	1	1	0

$R(y)$

Операция $x/2(P) = Q$, отображающая M в M, заменяет каждую строку исходной таблицы строкой из нее же, помеченной предметом $x = 2$; операция же $x/2(P) = R$, отображающая M в M', превращает строку $x = 2$ исходной таблицы в отдельную таблицу. Мы видим, что кроме предикатов существуют еще и операции над предикатами, а значит должна существовать и некая *алгебра предикатных операций*, на языке которой можно было бы такие операции записывать в виде формул.

Обратим внимание на то, что только что рассмотренная операция подстановки не может быть получена никакой суперпозицией операций $\vee, \wedge$ и $\neg$. Это означает, что на языке алгебры предикатов нельзя выразить произвольно выбранную операцию вида $F(P) = Q$, которая преобразует любой предикат P в соответствующий ему предикат Q. И, тем не менее, этот факт не вступает в противоречие с провозглашенным нами ранее тезисом о ничем не ограниченной описательной силе алгебры предикатов. Ведь никто не мешает нам посмотреть на запись $F(P) = Q$ не как на операцию (то есть реальное действие получения предиката Q по произвольно взятому предикату P), а просто как на уравнение (то есть как на отношение), связыва-

ющее две переменные P и Q. Кроме того, вполне допустимо рассматривать предикатные переменные как предметные. Верно, что значениями переменных P и Q служат предикаты, но также верно и то, что предикаты – это тоже предметы, только предметы специального вида. А раз это так, то для отношения, выраженного равенством $F(P)=Q$, всегда найдется соответствующий ему предикат $\Phi(P,Q)$, который можно будет записать в виде некоторой формулы.

Здесь обнаруживается еще один существенный недостаток алгебры предикатов (наряду с отмеченной ранее большей громоздкостью ее формул по сравнению с формулами специализированных алгебр, описывающих те же объекты): ее описания хоть и универсальны, но лишь связывают переменные, они не предназначены для явного выражения одних переменных через другие. Вся информация об операции $F(P)=Q$ в неявном виде в описании $\Phi(P,Q)$ содержится, но ее нельзя непосредственно использовать для практической реализации этой операции в материальной системе. Формулу же $F(P)=Q$ алгебры предикатных операций можно непосредственно использовать как алгоритм, по которому строится программа для ЭВМ. А формула $\Phi(P,Q)$ алгебры предикатов такой непосредственной возможности не дает. Операцию F из отношения Φ извлечь можно, но для этого придется выполнить дополнительную работу. Таким образом, роли, которые выполняют алгебра предикатов и алгебра предикатных операций, оказываются четко разделенными. Алгебра предикатов описывает знания, она лишь «созерцает» объекты. Она формирует только то, что называется «словом»: высказывания, суждения, предложения. Алгебра же предикатных операций призвана формировать реальные действия, она обслуживает «дело». Язык алгебры предикатов декларативен, язык же алгебры операций над предикатами процедурен. Итак, кроме алгебры предикатов, нужна еще и алгебра предикатных операций, если мы хотим не только познавать уже существующие интеллектуальные системы, но и создавать новые. Машина должна не только иметь свои мысли, но и обладать способностью оперировать с ними, приводить их в движение, получать новые мысли. Иначе говоря, она должна еще и мыслить.

Нам осталось рассмотреть задачу формульного описания бесконечных предикатов, заданных на декартовом произведении $A_1 \times A_2 \times \ldots \times A_m$ множеств $A_1, A_2, \ldots, A_m$, среди которых имеются и бесконечные. СДНФ некоторых из таких предикатов будут иметь бесконечную длину, такие формулы в развернутом виде записать невозможно. Но мы можем сделать так, как в подобных случаях поступают в классической математике: вводят знак многократной суммы, которая распространяется на бесконечную область. В нашем случае используем формулу

$$P(x_1,x_2,\ldots,x_m) = \bigvee_{(a_1,a_2,\ldots,a_m)\in P} x_1^{a_1} x_2^{a_2} \ldots x_m^{a_m},$$

означающую, что берется дизъюнкция всех конституэнт единицы с показателями $a_1,a_2,\ldots,a_m$, которые образуются из компонентов векторов. Сами же векторы $(a_1,a_2,\ldots,a_m)$ принадлежат бесконечному отношению $P \subseteq A_1 \times A_2 \times \ldots \times A_m$, соответствующему предикату $P(x_1,x_2,\ldots,x_m)$.

Данная статья является введением в математический аппарат алгебры предикатов, который представляет собой эффективное средство формального описания объектов бионики интеллекта. С помощью описанного математического аппарата получили развитие многие научные направления в области создания искусственного интеллекта.

Поступила в редколлегию 20.11.2004

УДК 519.7

М.Ф. Бондаренко, В.А. Чикина, Ю.П. Шабанов-Кушнаренко

МОДЕЛИ ЯЗЫКА

«Язык – это в сущности живая система, открытая непосредственному наблюдению. Чтобы наблюдать язык, не нужно ни электронного микроскопа, ни меченых атомов, ни других дорогостоящих приборов. (Эксперименты на ЭВМ при всей их важности имеют все-таки второстепенное значение для постижения самой природы языка). Нужны только наблюдения над текстами и над реакцией информантов – носителей языка и интенсивное размышление. А между тем изучение языка может оказать неоценимую пользу для понимания сущности живых систем».

Шрейдер Ю. А. [1, с. 29]

1. Склонение полных непритяжательных имен прилагательных

Совершенствование машинной обработки естественно-языковой информации является одной из важнейших задач компьютеризации и информатизации [2]. Особенно важно совершенствовать общение человека с ЭВМ на естественном языке, улучшать качество машинного перевода, поиска необходимой деловой и научной информации в Интернете. Решение этой задачи существенно сдерживается недостаточной глубиной формального описания механизмов языка. Компьютер – это математическая машина, в нем не удается привести в действие неформализованные знания о языке, которыми располагает лингвистика. До появления ЭВМ лингвистика обучала языку только людей, которые усваивают его гораздо легче, чем это могут делать машины, поскольку люди обладают неизмеримо более развитым интеллектом. До настоящего времени преобладают чрезвычайно трудоемкие и вместе с тем малоэффективные «силовые» методы решения задач машинной обработки информации, основанные на чисто эмпирических попытках искусственного воспроизведения на ЭВМ непосредственно наблюдаемых естественно-языковых процессов, то есть речевого поведения людей.

Между тем, многовековой опыт науки и техники учит, что гораздо более результативным и практичным является выявление и формальное описание глубинных законов, лежащих в основе изучаемых процессов. Известный московский лингвист, математик и философ Ю.А. Шрейдер пишет: «Рост науки в гораздо большей степени, чем добытыми фактами измеряется возникновением новых содержательных концепций. И очень часто существенно новые концептуальные достижения определяются не силовыми методами с помощью ультрасовременных «отмычек», а возникают в тиши кабинета с помощью наиболее совершенного инструмента – человеческого разума. В начале пятидесятых годов складывалось впечатление, что электронные вычислительные машины послужат той самой «отмычкой», которая даст принципиально новые сведения о языке и позволит быстро автоматизировать процессы языковой коммуникации (проблема машинного перевода и некоторые связанные с нею проблемы). Смысл неудачи грубых чисто эмпирических попыток машинного перевода как раз и состоял в том, что электронные вычислительные машины не сумели сыграть роль отмычки по отношению к языку. Волей – неволей пришлось использовать человеческий разум и его способности математического описания явлений. Только весьма развитое математическое описание языка делает осмысленным дальнейшее использование технико-кибернетических методов в лингвистике для всякого рода практических приложений» [1, с. 38–39].

Чтобы подойти к выявлению и формальному описанию законов естественного языка, полезно воспользоваться опытом классической физики, которая описывает законы, царящие во внешнем мире, посредством уравнений. Каждое уравнение выражает некоторую связь между вполне определенными переменными, которая с математической точки зрения представляет собой какое-то отношение. За многие века наука не обнаружила в мире ни одного объекта или процесса, которые невозможно было бы выразить на языке отношений. Фразы, которыми люди выражают свои мысли, – это отношения; более того, весь механизм языка представляет собой систему отношений и ничего более. Отсюда вытекает рекомендация: законы естественного языка следует описывать в виде уравнений на некотором математическом языке. Можно ли для этой цели воспользоваться тем языком математики, на котором описывает свои объекты классическая физика? Нет, нельзя. Дело в том, что во внешнем мире наблюдаются, в основном, непрерывные процессы, которые для своего формального описания нуждаются в языке числовой математики. Во внутреннем же мире человека преобладают процессы, имеющие логическую природу. Для формального описания законов естественного языка уместен язык логической математики, а именно, *алгебра предикатов*, на языке которой могут быть записаны в виде уравнений любые отноше-

ния. Аппарат алгебры предикатов развит настолько, что с его помощью можно уже в настоящее время приступить к формальному описанию механизма естественного языка.

Естественных языков много, мы останавливаемся на русском языке. Механизм естественного языка слишком обширен, чтобы его можно было охватить одним конкретным исследованием. Поневоле приходится выделить в нем для исследования небольшую посильную часть. Такое выделение мы осуществим, применяя методы алгебры предикатов к знаниям, накопленным классическим языкознанием, которые содержатся в грамматиках русского языка (см. напр. [3, 4]). Грамматика распадается на два больших раздела: морфологию – учение о слове и синтаксис - учение о предложении. Мы останавливаемся на морфологии, поскольку в грамматике она предшествует синтаксису. Морфология, в свою очередь, делится на учение о словоизменении и учение о словообразовании. Остановившись на словоизменении, мы далее расчленяем его на склонение имен и спряжение глаголов. Избрав склонение имен, далее останавливаемся на именах прилагательных. Наконец, последний выбор производим, ограничивая себя склонением полных непритяжательных имен прилагательных. Описывая формально различные механизмы естественного языка, можно будет затем воспроизводить их средствами вычислительной техники, постепенно повышая за счет этого уровень искусственного интеллекта.

В этой статье для примера рассматривается, формально описывается на языке алгебры предикатов и доводится до воспроизведения на *логической сети мозгоподобной ЭВМ* механизм склонения полных непритяжательных имен прилагательных. Грамматика знакомит школьников с процессом склонения полных непритяжательных имен прилагательных с помощью табл. 1, называемой *парадигматической.*

Таблица 1

Падеж, x_3	Число, x_2			
	Единственное			Множественное
	Род, x_1			
	мужской	женский	средний	
именительный	*сла'бый*	*сла'бая*	*сла'бое*	*сла'бые*
родительный	*сла'бого*	*сла'бой*	*сла'бого*	*сла'бых*
дательный	*сла'бому*	*сла'бой*	*сла'бому*	*сла'бым*
винительный	*сла'бый*	*сла'бую*	*сла'бое*	*сла'бые*
	сла'бого			*сла'бых*
творительный	*сла'бым*	*сла'бой*	*сла'бым*	*сла'быми*
		сла'бою		
предложный	*сла'бом*	*сла'бой*	*сла'бом*	*сла'бых*

В ячейках таблицы указаны *формы слова* в зависимости от их рода x_1, числа x_2 и падежа x_3. Таблица составлена для конкретного слова *сла'бый.* В каждом слове и во всех его формах будем отмечать ударение знаком ', который ставим после буквы, произносимой с ударением. Каждое *слово* будем задавать его *словарной формой*, приводимой в орфографических словарях, которая соответствует мужскому роду, единственному числу и именительному падежу. Формальное описание процесса склонения полных непритяжательных имен прилагательных нами выполняется для множества слов, указанных в «Грамматическом словаре русского языка» [5]. В ячейках таблицы для мужского рода, единственного числа и винительного падежа, а также для множественного числа и винительного падежа содержатся по две формы слова: одна – для неодушевленных предметов и другая – для одушевленных. Две формы слова – современная и архаичная содержатся также и в ячейке таблицы для единственного числа женского рода творительного падежа.

Любую форму каждого слова можно расчленить на ее *основу* и *окончание*, например, *сла'б-ого.* Основа и окончание словарной формы слова называются соответственно *основой* и *окончанием слова.* Ударение в формах слов может падать как на основу, так и на окончание, например, *сла'бый, седо'й.* Ударение в полных непритяжательных именах прилагательных *неподвижно.* Это значит, что при склонении слова ударение не меняет своего положения. При склонении других частей речи ударение может оказаться подвижным (например, *но'ж – ножа'*). Существуют полные непритяжательные имена прилагательные, которые склоняются поразному. Многие же слова склоняются однотипно. Все слова одного *типа склонения* имеют парадигматические таблицы с одними и теми же окончаниями. Слова различных типов склоняются по-разному, имеют различные таблицы окончаний. Типы склонения характеризуют *влияние слова* на его окончание. Всего имеется семь типов склонения полных непритяжательных имен прилагательных. Ниже приведены таблицы 2–8 окончаний для всех типов склонения с указанием их номеров, которые сопровождаются примерами слов, склоняемых по данному типу.

Таблица 2

1 – сла'бый			
ый	*ая*	*ое*	*ые*
ого	*ой*	*ого*	*ым*
ому	*ой*	*ому*	*ым*
ый	*ую*	*ое*	*ые*
ого			*ых*
ым	*ой*	*ым*	*ыми*
	ою		
ом	*ой*	*ом*	*ых*

Таблица 3

2 – ве'тхий			
ий	*ая*	*ие*	*ие*
ого	*ой*	*ого*	*их*
ому	*ой*	*ому*	*им*
ий	*ую*	*ое*	*ие*
ого			*их*
им	*ой*	*им*	*ими*
	ою		
ом	*ой*	*ом*	*их*

Таблица 4

3 – *ку'цый*			
ый	*ая*	*ее*	*ые*
его	*ей*	*его*	*ых*
ему	*ей*	*ему*	*ым*
ый	*ую*	*ее*	*ые*
его			*ых*
ым	*ей*	*ым*	*ыми*
	ею		
ем	*ей*	*ем*	*ых*

Таблица 5

4 – *ры'жий*			
ий	*ая*	*ее*	*ие*
его	*ей*	*его*	*их*
ему	*ей*	*ему*	*им*
ий	*ею*	*ее*	*ие*
его			*их*
им	*ей*	*им*	*ими*
	ею		
ем	*ей*	*ем*	*их*

Таблица 6

5 – *си'ний*			
ий	*яя*	*ее*	*ие*
его	*ей*	*его*	*их*
ему	*ей*	*ему*	*им*
ий	*юю*	*ее*	*ие*
его			*их*
им	*ей*	*им*	*ими*
	ею		
ем	*ей*	*ем*	*их*

Таблица 7

6 – *седо'й*			
о'й	*а'я*	*о'е*	*ы'е*
о'го	*о'й*	*о'го*	*ы'х*
о'му	*о'й*	*о'му*	*ы'м*
о'й	*у'ю*	*о'е*	*ы'е*
о'го			*ы'х*
ы'м	*о'й*	*ы'м*	*ы'ми*
	о'ю		
о'м	*о'й*	*о'м*	*ы'х*

Таблица 8

7 – *сухо'й*			
о'й	*а'я*	*о'е*	*и'е*
о'го	*о'й*	*о'го*	*и'х*
о'му	*о'й*	*о'му*	*и'м*
о'й	*у'ю*	*о'е*	*и'е*
о'го			*и'х*
и'м	*о'й*	*и'м*	*и'ми*
	о'ю		
о'м	*о'й*	*о'м*	*и'х*

2. Математическое описание склонения полных непритяжательных имен прилагательных

Приступаем к формальному описанию процесса склонения полных непритяжательных имен прилагательных, руководясь таблицами 2–8. С этой целью вводим необходимые предметные переменные: x_1 – *род* формы слова со значениями М – мужской, Ж – женский, С – средний; x_2 – *число* формы слова со значениями Е – единственное, М – множественное; x_3 – *падеж* формы слова со значениями И – именительный, Р – родительный, Д – дательный, В – винительный, Т – творительный, П – предложный; x_4 – *признак одушевленности* формы слова со значениями О – одушевленный, Н – неодушевленный; x_5 – *признак употребляемости* формы слова со значениями С – современная, А – архаичная; t – *тип склонения* слова со значениями 1–7; z – *окончание* формы слова со значениями АЯ, А'Я, ЯЯ, УЮ, У'Ю, ЮЮ, ОЕ, О'Е, ЕЕ, ОЙ, О'Й, ЕЙ, ОМ, О'М, ЕМ, ОГО, О'ГО, ЕГО, ОМУ, О'МУ, ЕМУ, ОЮ, О'Ю, ЕЮ, ЫЙ, ИЙ, ЫМ, Ы'М, ИМ, И'М, ЫЕ, Ы'Е, ИЕ, И'Е, ЫХ, Ы'Х, ИХ, И'Х, ЫМИ, Ы'МИ, ИМИ, И'МИ.

Контекст, в который включено полное непритяжательное имя прилагательное, влияет на форму слова *грамматическими признаками* $x_1 - x_5$. *Влияние контекста* на окончание слова определяется только значениями грамматических признаков. Влияний контекста не может быть больше, чем ячеек парадигматической таблицы. Производим нумерацию ячеек парадигматической таблицы. Полученные номера ячеек в дальнейшем будут использованы для обозначения влияний контекста. Некоторые из влияний контекста получат более одного имени, однако, от этого недостатка мы впоследствии постараемся избавиться. Для каждой ячейки парадигматической таблицы вводим свой номер q (табл. 9).

Таблица 9

x_2 / x_1	МЕ	ЖЕ	СЕ	-М		
И	1	8	15	21		q
Р	2	9	16	22		
Д	3	10	17	23		
В	4	11	18	24	Н	x_4:
	5			25	О	
Т	6	12	19	26	С	x_5:
		13			А	
П	7	14	20	27		

Выражаем номера q ячеек парадигматической таблицы через признаки $x_1 - x_5$:

$$x_1^{\text{М}} x_2^{\text{Е}} x_3^{\text{И}} = q^1;\ x_1^{\text{М}} x_2^{\text{Е}} x_3^{\text{Р}} = q^2;\ x_1^{\text{М}} x_2^{\text{Е}} x_3^{\text{Д}} = q^3;$$

$$x_1^{\text{М}} x_2^{\text{Е}} x_3^{\text{В}} x_4^{\text{Н}} = q^4;\ x_1^{\text{М}} x_2^{\text{Е}} x_3^{\text{В}} x_4^{\text{О}} = q^5;\ x_1^{\text{М}} x_2^{\text{Е}} x_3^{\text{Т}} = q^6;$$

$$x_1^{\text{М}} x_2^{\text{Е}} x_3^{\text{П}} = q^7;\ x_1^{\text{Ж}} x_2^{\text{Е}} x_3^{\text{И}} = q^8;\ x_1^{\text{Ж}} x_2^{\text{Е}} x_3^{\text{Р}} = q^9;$$

$$x_1^{\text{Ж}} x_2^{\text{Е}} x_3^{\text{Д}} = q^{10};\ x_1^{\text{Ж}} x_2^{\text{Е}} x_3^{\text{В}} = q^{11};\ x_1^{\text{Ж}} x_2^{\text{Е}} x_3^{\text{Т}} x_5^{\text{С}} = q^{12};$$

$$x_1^{\text{Ж}} x_2^{\text{Е}} x_3^{\text{Т}} x_5^{\text{А}} = q^{13};\ x_1^{\text{Ж}} x_2^{\text{Е}} x_3^{\text{П}} = q^{14};\ x_1^{\text{С}} x_2^{\text{Е}} x_3^{\text{И}} = q^{15};$$

$$x_1^{\text{С}} x_2^{\text{Е}} x_3^{\text{Р}} = q^{16};\ x_1^{\text{С}} x_2^{\text{Е}} x_3^{\text{Д}} = q^{17};\ x_1^{\text{С}} x_2^{\text{Е}} x_3^{\text{В}} = q^{18};\ x_1^{\text{С}} x_2^{\text{Е}} x_3^{\text{Т}} = q^{19};$$

$$x_1^{\text{С}} x_2^{\text{Е}} x_3^{\text{П}} = q^{20};\ x_2^{\text{М}} x_3^{\text{И}} = q^{21};\ x_2^{\text{М}} x_3^{\text{Р}} = q^{22};\ x_2^{\text{М}} x_3^{\text{Д}} = q^{23};$$

$$x_2^{\text{М}} x_3^{\text{В}} x_4^{\text{Н}} = q^{24};\ x_2^{\text{М}} x_3^{\text{В}} x_4^{\text{О}} = q^{25};\ x_2^{\text{М}} x_3^{\text{Т}} = q^{26};\ x_2^{\text{М}} x_3^{\text{П}} = q^{27}.$$

Выполняем *операцию почленной дизъюнкции* возможно большего числа родственных равенств:

$$x_1^{\text{М}} x_2^{\text{Е}} x_3^{\text{И}} = q^1;\ (x_1^{\text{М}} \vee x_1^{\text{С}}) x_2^{\text{Е}} x_3^{\text{Р}} = q^2 \vee q^{16};$$

$$(x_1^{\text{М}} \vee x_1^{\text{С}}) x_2^{\text{Е}} x_3^{\text{Д}} = q^3 \vee q^{17};\ x_1^{\text{М}} x_2^{\text{Е}} x_3^{\text{В}} x_4^{\text{Н}} = q^4;$$

$$x_1^{\text{М}} x_2^{\text{Е}} x_3^{\text{В}} x_4^{\text{О}} = q^5;\ (x_1^{\text{М}} \vee x_1^{\text{С}}) x_2^{\text{Е}} x_3^{\text{Т}} = q^6 \vee q^{19};$$

$$(x_1^{\text{М}} \vee x_1^{\text{С}}) x_2^{\text{Е}} x_3^{\text{П}} = q^7 \vee q^{20};\ x_1^{\text{Ж}} x_2^{\text{Е}} x_3^{\text{И}} = q^8;$$

$$x_1^{\text{Ж}} x_2^{\text{Е}} (x_3^{\text{Р}} \vee x_3^{\text{Д}} \vee x_3^{\text{П}}) = q^9 \vee q^{10} \vee q^{14};\ x_1^{\text{Ж}} x_2^{\text{Е}} x_3^{\text{В}} = q^{11};$$

$$x_1^{\text{Ж}} x_2^{\text{Е}} x_3^{\text{Т}} x_5^{\text{С}} = q^{12};\ x_1^{\text{Ж}} x_2^{\text{Е}} x_3^{\text{Т}} x_5^{\text{А}} = q^{13};$$

$$x_1^{\text{С}} x_2^{\text{Е}} (x_3^{\text{И}} \vee x_3^{\text{В}}) = q^{15} \vee q^{18};\ x_2^{\text{М}} x_3^{\text{И}} = q^{21};$$

$$x_2^{\text{М}} (x_3^{\text{Р}} \vee x_3^{\text{П}}) = q^{22} \vee q^{27};\ x_2^{\text{М}} x_3^{\text{Д}} = q^{23};$$

$$x_2^{\text{М}} x_3^{\text{В}} x_4^{\text{Н}} = q^{24};\ x_2^{\text{М}} x_3^{\text{В}} x_4^{\text{О}} = q^{25};\ x_2^{\text{М}} x_3^{\text{Т}} = q^{26}.$$

Родственными называются такие равенства, которые после выполнения над ними операции почленной дизъюнкции приводят к равенствам с левой частью в виде логического произведения, каж-

дый сомножитель которого зависит лишь от одной предметной переменной. Мотивом, побудившим выполнить операцию почленной дизъюнкции, является стремление получить экономное множество имен влияний контекстов, в котором каждому влиянию контекста соответствовало бы в точности одно имя.

Формируем функцию перехода от номеров ячеек парадигматической таблицы q к номерам влияний контекста r:

$$q^1 = r^1;\ q^2 \vee q^{16} = r^2;\ q^3 \vee q^{17} = r^3;\ q^4 = r^4;\ q^5 = r^5;$$
$$q^6 \vee q^{19} = r^6;\ q^7 \vee q^{20} = r^7;\ q^8 = r^8;\ q^9 \vee q^{10} \vee q^{14} = r^9;$$
$$q^{11} = r^{10};\ q^{12} = r^{11};\ q^{13} = r^{12};\ q^{15} \vee q^{18} = r^{13};\ q^{21} = r^{14};$$
$$q^{22} \vee q^{27} = r^{15};\ q^{23} = r^{16};\ q^{24} = r^{17};\ q^{25} = r^{18};\ q^{26} = r^{19}.$$

Вводим номера влияний контекста, перенумеровывая ячейки парадигматической табл. 9 (табл. 10):

Таблица 10

x_1 \ x_2	М Е	Ж Е	СЕ	-М
И	1	8	13	14
Р	2	9	2	15
Д	3	9	3	16
В	4	10	13	17
	5			18
Т	6	11	6	19
		12		
П	7	9	7	15

r

НОСА x_4:

x_5:

Формируем зависимость номера влияния контекста r от переменных x_1, x_2, x_3, x_4, x_5:

$$x_1^{\text{М}} x_2^{\text{Е}} x_3^{\text{И}} = r^1;\ (x_1^{\text{М}} \vee x_1^{\text{С}}) x_2^{\text{Е}} x_3^{\text{Р}} = r^2;$$
$$(x_1^{\text{М}} \vee x_1^{\text{С}}) x_2^{\text{Е}} x_3^{\text{Д}} = r^3;\ x_1^{\text{М}} x_2^{\text{Е}} x_3^{\text{В}} x_4^{\text{Н}} = r^4;$$
$$x_1^{\text{М}} x_2^{\text{Е}} x_3^{\text{В}} x_4^{\text{О}} = r^5;\ (x_1^{\text{М}} \vee x_1^{\text{С}}) x_2^{\text{Е}} x_3^{\text{Т}} = r^6;$$
$$(x_1^{\text{М}} \vee x_1^{\text{С}}) x_2^{\text{Е}} x_3^{\text{П}} = r^7;\ x_1^{\text{Ж}} x_2^{\text{Е}} x_3^{\text{И}} = r^8;$$
$$x_1^{\text{Ж}} x_2^{\text{Е}} (x_3^{\text{Р}} \vee x_3^{\text{Д}} \vee x_3^{\text{П}}) = r^9;\ x_1^{\text{Ж}} x_2^{\text{Е}} x_3^{\text{В}} = r^{10};$$
$$x_1^{\text{Ж}} x_2^{\text{Е}} x_3^{\text{Т}} x_5^{\text{С}} = r^{11};\ x_1^{\text{Ж}} x_2^{\text{Е}} x_3^{\text{Т}} x_5^{\text{А}} = r^{12};$$
$$x_1^{\text{С}} x_2^{\text{Е}} (x_3^{\text{И}} \vee x_3^{\text{В}}) = r^{13};\ x_2^{\text{М}} x_3^{\text{И}} = r^{14};$$
$$x_2^{\text{М}} (x_3^{\text{Р}} \vee x_3^{\text{П}}) = r^{15};\ x_2^{\text{М}} x_3^{\text{Д}} = r^{16};\ x_2^{\text{М}} x_3^{\text{В}} x_4^{\text{Н}} = r^{17};$$
$$x_2^{\text{М}} x_3^{\text{В}} x_4^{\text{О}} = r^{18};\ x_2^{\text{М}} x_3^{\text{Т}} = r^{19}.$$

Производим *бинаризацию* только что записанного отношения, связывающего переменную r с переменными x_1, x_2, x_3, x_4, x_5. Для этого отыскиваем отношение P_1, связывающее переменные x_1 и r:

$$P_1(x_1, r) = x_1^{\text{М}}(r^1 \vee r^4 \vee r^5) \vee (x_1^{\text{М}} \vee x_1^{\text{С}})(r^2 \vee r^3 \vee r^6 \vee r^7) \vee x_1^{\text{Ж}}(r^8 \vee r^9 \vee r^{10} \vee r^{11} \vee r^{12}) \vee x_1^{\text{С}} r^{13} \vee (x_1^{\text{М}} \vee x_1^{\text{Ж}} \vee x_1^{\text{С}})(r^{14} \vee r^{15} \vee r^{16} \vee r^{17} \vee r^{18} \vee r^{19}).$$

Далее получаемые отношения изображаются также в виде двудольных графов:

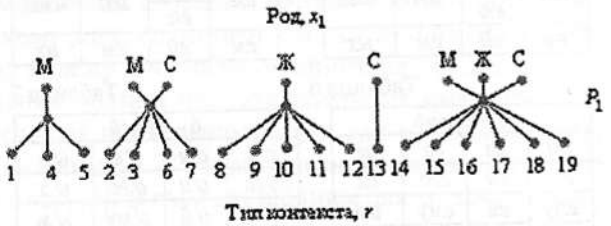


Рис. 1

Отыскиваем отношение P_2, связывающее переменные x_2 и r:

$$P_2(x_2, r) = x_2^{\text{Е}}(r^1 \vee r^2 \vee r^3 \vee r^4 \vee r^5 \vee r^6 \vee r^7 \vee r^8 \vee r^9 \vee r^{10} \vee r^{11} \vee r^{12} \vee r^{13}) \vee x_2^{\text{М}}(r^{14} \vee r^{15} \vee r^{16} \vee r^{17} \vee r^{18} \vee r^{19}).$$

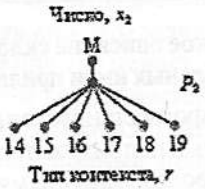
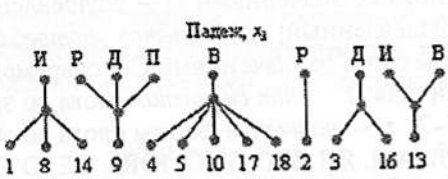


Рис. 2

Отыскиваем отношение P_3, связывающее переменные x_3 и r:

$$P_3(x_3, r) = x_3^{\text{И}}(r^1 \vee r^8 \vee r^{14}) \vee x_3^{\text{Р}} r^2 \vee x_3^{\text{Д}}(r^3 \vee r^{16}) \vee x_3^{\text{В}}(r^4 \vee r^5 \vee r^{10} \vee r^{17} \vee r^{18}) \vee x_3^{\text{Т}}(r^6 \vee r^7 \vee r^{11} \vee r^{12} \vee r^{19}) \vee (x_3^{\text{Р}} \vee x_3^{\text{Д}} \vee x_3^{\text{П}}) r^9 \vee (x_3^{\text{И}} \vee x_3^{\text{В}}) r^{13} \vee (x_3^{\text{Р}} \vee x_3^{\text{П}}) r^{15}.$$



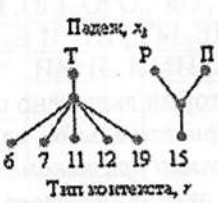


Рис. 3

Отыскиваем отношение P_4, связывающее переменные x_4 и r:

$$P_4(x_4,r)=x_4^{\text{Н}}(r^4\vee r^{17})\vee x_4^{\text{О}}(r^5\vee r^{18})\vee$$
$$\vee(x_4^{\text{Н}}\vee x_4^{\text{О}})(r^1\vee r^2\vee r^3\vee r^6\vee r^7\vee r^8\vee r^9\vee$$
$$\vee r^{10}\vee r^{11}\vee r^{12}\vee r^{13}\vee r^{14}\vee r^{15}\vee r^{16}\vee r^{19}).$$

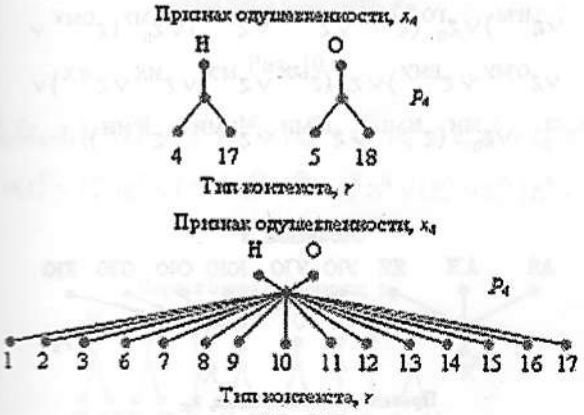


Рис. 4

Отыскиваем отношение P_5, связывающее переменные x_5 и r:

$$P_5(x_5,r)=x_5^{\text{С}}r^{11}\vee x_5^{\text{А}}r^{12}\vee(x_5^{\text{С}}\vee x_5^{\text{А}})(r^1\vee$$
$$\vee r^2\vee r^3\vee r^4\vee r^5\vee r^6\vee r^7\vee r^8\vee r^9\vee$$
$$\vee r^{10}\vee r^{13}\vee r^{14}\vee r^{15}\vee r^{16}\vee r^{17}\vee r^{18}\vee r^{19}).$$

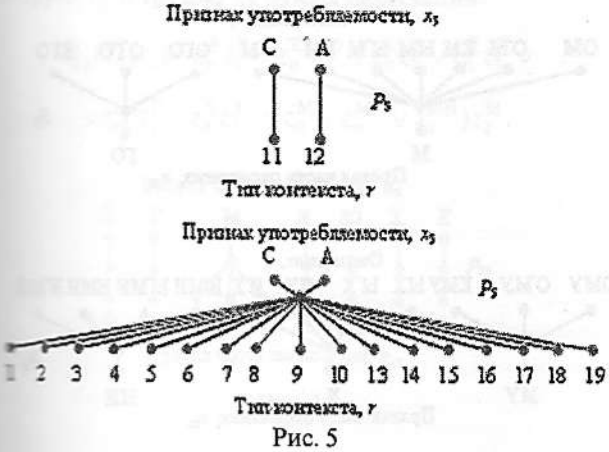


Рис. 5

Вводим левую $z_л$ и правую $z_п$ части окончания. Окончание z естественным образом расчленяется на левую и правую части, например, А′Я → (А, Я), А′Я → (А′, Я), ОГО → (О, ГО), О′ГО → (О′, ГО). Левая часть окончания состоит из двух компонентов: первой буквы окончания и значения признака ее ударности. Правая часть окончания состоит также из двух компонентов – второй и третьей букв окончания. Основанием для разделения окончания на левую и правую части является то, что правая часть окончания определяется исключительно влиянием контекста (признаки $x_1 - x_5$), тогда как левая часть окончания определяется, кроме того, еще и типом склонения слова s. Рассмотрим отношение P_6, связывающее номер влияния контекста r с левой частью окончания $z_л$ формы слова, присутствующей в этом контексте. Эта связь выражена табл. 11.

Таблица 11

Ы, И, О′	А, А′, Я	О, О′, Е	Ы, Ы′, И, И′
О, О′, Е	О, О′, Е	О, О′, Е	Ы, Ы′, И, И′
О, О′, Е	О, О′, Е	О, О′, Е	Ы, Ы′, И, И′
Ы, И, О′	У, У′, Ю	О, О′, Е	Ы, Ы′, И, И′
О,′О,Е			Ы, Ы′, И, И′
Ы, Ы′, И, И′	О, О′, Е	Ы, Ы′, И, И′	Ы, Ы′, И, И′
	О, О′, Е		
О, О′, Е	О, О′, Е	О, О′, Е	Ы, Ы′, И, И′

Записываем отношение P_6 в виде формулы:

$$P_6(z_л,r)=(z_л^{\text{Ы}}\vee z_л^{\text{И}}\vee z_л^{\text{О}})(r^1\vee r^4)\vee(z_л^{\text{О}}\vee$$
$$\vee z_л^{\text{О}'}\vee z_л^{\text{Е}})(r^2\vee r^3\vee r^5\vee r^7\vee r^9\vee r^{11}\vee$$
$$\vee r^{12}\vee r^{13})\vee(z_л^{\text{Ы}}\vee z_л^{\text{Ы}'}\vee z_л^{\text{И}}\vee z_л^{\text{И}'})(r^6\vee$$
$$\vee r^{14}\vee r^{15}\vee r^{16}\vee r^{17}\vee r^{18}\vee r^{19})\vee(z_л^{\text{А}}\vee$$
$$\vee z_л^{\text{А}'}\vee z_л^{\text{Я}})r^8\vee(z_л^{\text{У}}\vee z_л^{\text{У}'}\vee z_л^{\text{Я}})r^{10}.$$

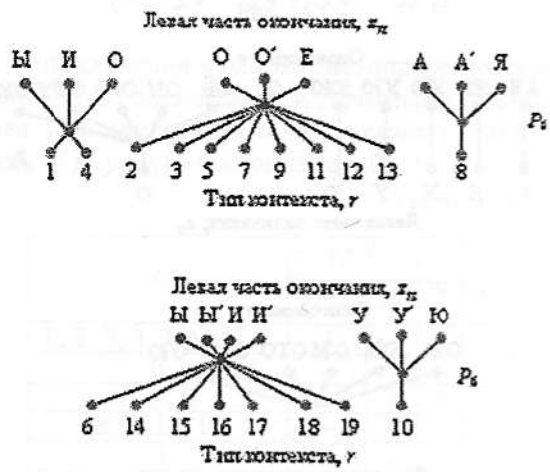


Рис. 6

Отношение P_7, связывающее номер влияния контекста r с правой частью окончания $z_п$, представлено в табл. 12.

Таблица 12

Й	Я	Е	Е
			Х
			М
Й	Ю	Е	Е
ГО			Х
М	Й	М	МИ
	Ю		
М	Й	М	Х

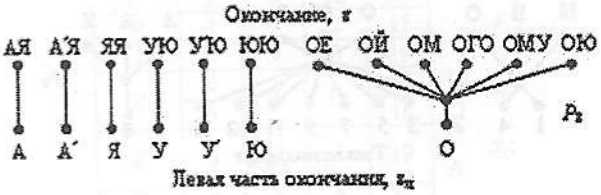
Записываем отношение P_7 в виде формулы:

$$P_7(z_п,r)=z_п^{\text{Й}}(r^1\vee r^4\vee r^9\vee r^{11})\vee z_п^{\text{ГО}}(r^2\vee$$
$$\vee r^5)\vee z_п^{\text{МУ}}r^3\vee z_п^{\text{М}}(r^6\vee r^7\vee r^{16})\vee z_п^{\text{Я}}r^8.$$

Рис. 7

Ниже приводятся формулы для отношений P_8 и P_9, связывающих окончание z с его левой $z_л$ и правой $z_п$ частями:

$$P_8(z_л,z)=z_л^{\wedge}z^{\wedge Я}\vee z_л^{\wedge'}z^{\wedge'Я}\vee z_л^{Я}z^{ЯЯ}\vee z_л^{У}z^{УЮ}\vee$$
$$\vee z_л^{У'}z^{У'Ю}\vee z_л^{Ю}z^{ЮЮ}\vee z_л^{О}(z^{ОЕ}\vee z^{ОЙ}\vee z^{ОМ}\vee z^{ОГО}\vee$$
$$\vee z^{ОМУ}\vee z^{ОЮ})\vee z_л^{О'}(z^{О'Е}\vee z^{О'Й}\vee z^{О'М}\vee z^{О'ГО}\vee$$
$$\vee z^{О'МУ}\vee z^{О'Ю})\vee z_л^{Е}(z^{ЕЕ}\vee z^{ЕЙ}\vee z^{ЕМ}\vee z^{ЕГО}\vee$$
$$\vee z^{ЕМУ}\vee z^{ЕЮ})\vee z_л^{Ы}(z^{ЫЙ}\vee z^{ЫМ}\vee z^{ЫЕ}\vee z^{ЫХ}\vee$$
$$\vee z^{ЫМИ})\vee z_л^{Ы'}(z^{Ы'М}\vee z^{Ы'Е}\vee z^{Ы'Х}\vee z^{Ы'МИ})\vee$$
$$\vee z_л^{И}(z^{ИЙ}\vee z^{ИМ}\vee z^{ИЕ}\vee z^{ИХ}\vee z^{ИМИ})\vee$$
$$\vee z_л^{И'}(z^{И'М}\vee z^{И'Е}\vee z^{И'Х}\vee z^{И'МИ}).$$



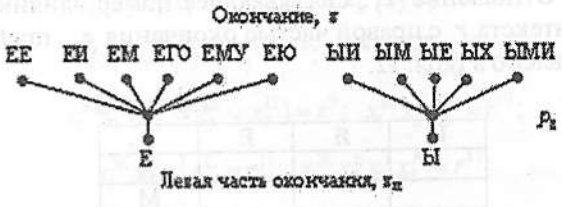
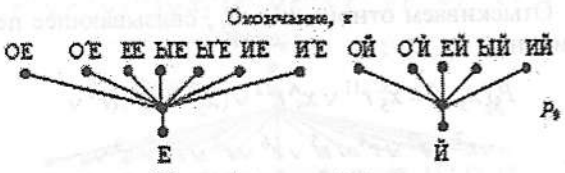


Рис. 8

$$P_9(z_п,z)=z_п^{Я}(z^{АЯ}\vee z^{А'Я}\vee z^{ЯЯ})\vee z_п^{Ю}(z^{УЮ}\vee z^{У'Ю}\vee$$
$$\vee z^{ЮЮ}\vee z^{ОЮ}\vee z^{О'Ю}\vee z^{ЕЮ})\vee z_п^{Е}(z^{ОЕ}\vee z^{О'Е}\vee z^{ЕЕ}\vee$$
$$\vee z^{ЫЕ}\vee z^{Ы'Е}\vee z^{ИЕ}\vee z^{И'Е})\vee z_п^{Й}(z^{ОЙ}\vee z^{О'Й}\vee z^{ЕЙ}\vee$$
$$\vee z^{ЫЙ}\vee z^{ИЙ})\vee z_п^{М}(z^{ОМ}\vee z^{О'М}\vee z^{ЕМ}\vee z^{ЫМ}\vee z^{ИМ}\vee$$
$$\vee z^{И'М})\vee z_п^{ГО}(z^{ОГО}\vee z^{О'ГО}\vee z^{ЕГО})\vee z_п^{МУ}(z^{ОМУ}\vee$$
$$\vee z^{О'МУ}\vee z^{ЕМУ})\vee z_п^{Х}(z^{ЫХ}\vee z^{Ы'Х}\vee z^{ИХ}\vee z^{И'Х})\vee$$
$$\vee z_п^{МИ}(z^{ЫМИ}\vee z^{Ы'МИ}\vee z^{ИМИ}\vee z^{И'МИ}).$$



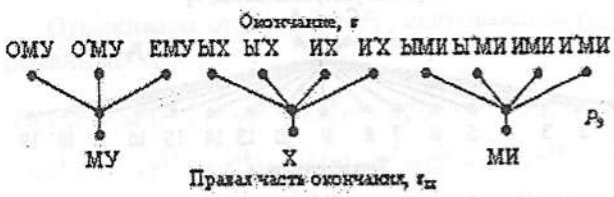


Рис. 9

Осталось развернуть окончание z по буквам. Разворачиваем левую часть окончания в знак ударения и первую букву окончания. Ниже приводится формульное и графовое описание отношений P_{10} и P_{11}, связывающих левую часть окончания $z_л$ со знаком ударения z_0 и с первой буквой окончания z_1.

$$P_{10}(z_л,z_0)=(z_л^{Ы}\vee z_л^{И}\vee z_л^{О}\vee z_л^{Е}\vee z_л^{А}\vee z_л^{Я}\vee z_л^{У}\vee$$
$$\vee z_л^{Ю})z_0^{*}\vee(z_л^{О'}\vee z_л^{Ы'}\vee z_л^{И'}\vee z_л^{А'}\vee z_л^{У'})z_0'.$$

Знак * при переменной z_0 означает отсутствие знака ударения над первой буквой окончания.

Рис. 10

$$P_{11}(z_л,z_1)=(z_л^{Ы}\vee z_л^{Ы'})z_1^{Ы}\vee(z_л^{И}\vee z_л^{И'})z_1^{И}\vee z_л^{Е}z_1^{Е}\vee z_л^{Ю}z_1^{Ю}\vee$$
$$\vee(z_л^{У}\vee z_л^{У'})z_1^{У}\vee(z_л^{О}\vee z_л^{О'})z_1^{О}\vee z_л^{Я}z_1^{Я}\vee(z_л^{А}\vee z_л^{А'})z_1^{А}.$$

Рис. 11

Разворачиваем правую часть окончания во вторую и третью буквы окончания. Ниже приводится формульное и графовое описание отношений P_{12} и P_{13}, связывающих правую часть окончания $z_п$ во вторую z_2 и третью z_3 буквы окончания.

$$P_{12}(z_п,z_2)=z_п^{Я}z_2^{Я}\vee z_п^{Ю}z_2^{Ю}\vee z_п^{Е}z_2^{Е}\vee z_п^{Й}z_2^{Й}\vee$$
$$\vee z_п^{ГО}z_2^{Г}\vee z_п^{Х}z_2^{Х}\vee(z_п^{М}\vee z_п^{МУ}\vee z_п^{МИ})z_2^{М}.$$

Рис. 12

$$P_{13}(z_п,z_3)=(z_п^{Я}\vee z_п^{Ю}\vee z_п^{Е}\vee z_п^{Й}\vee z_п^{Х}\vee$$
$$\vee z_п^{М})z_3^{*}\vee z_п^{ГО}z_3^{О}\vee z_п^{МИ}z_3^{И}\vee z_п^{МУ}z_3^{У}.$$

Знак * при переменной z_3 означает отсутствие третьей буквы в окончании.

Рис. 13

Тип склонения s связан с левой частью окончания $z_л$ отношением P_{14}, представленным табл. 13:

Таблица 13

S	1	2	3	4	5	6	7
$z_л$	А, У, О, Ы	А, У, О, И	А, У, Е, Ы	А, У, Е, И	Я, Ю, Е, И	'А, 'У, 'О, 'Ы	'А, 'У, 'О, 'И

Это же отношение, выраженное формулой алгебры предикатов, имеет вид:

$$P_{14}(s,z_л)=s^1(z_л^{А}\vee z_л^{У}\vee z_л^{О}\vee z_л^{Ы})\vee s^2(z_л^{А}\vee z_л^{У}\vee z_л^{О}\vee$$
$$\vee z_л^{И})\vee s^3(z_л^{А}\vee z_л^{У}\vee z_л^{Е}\vee z_л^{Ы})\vee s^4(z_л^{А}\vee z_л^{У}\vee z_л^{Е}\vee$$
$$\vee z_л^{И})\vee s^5(z_л^{Я}\vee z_л^{Ю}\vee z_л^{Е}\vee z_л^{И})\vee s^6(z_л^{А'}\vee z_л^{У'}\vee z_л^{О'}\vee$$
$$\vee z_л^{Ы'})\vee s^7(z_л^{А'}\vee z_л^{У'}\vee z_л^{О'}\vee z_л^{И'}).$$

Рис. 14

Тип склонения s однозначно определяется *последней буквой основы слова* y_1, *признаком ударности слова* y_2 и *признаком смягчения слова* y_3. Зависимость $g(y_1,y_2,y_3)=s$ выражена табл. 14.

Таблица 14

y_1	y_2=У		Б	
	y_3=Т	М	Т	М
Б, В, Д, З, Л, М, П, Т	1	–	6	–
Е	–	5	–	–
Н, Р, С	1	5	6	–
Ж, Ч, Ш, Щ	–	4	7	–
Ц	3	–	–	–
Г, К, Х	–	2	7	–

Зависимость g представляет собой частичную функцию, о чем свидетельствуют прочерки в таблице. В тех ячейках таблицы, где указаны прочерки, значение функции g не существует. Ниже функция $g(y_1,y_2,y_3)=s$ выражена на языке алгебры предикатов:

$$(y_1^{Б}\vee y_1^{В}\vee y_1^{Д}\vee y_1^{З}\vee y_1^{Л}\vee y_1^{М}\vee y_1^{П}\vee y_1^{Т}\vee y_1^{Н}\vee$$
$$\vee y_1^{Р}\vee\vee y_1^{С})y_2^{У}y_3^{Т}=s^1;\ (y_1^{Г}\vee y_1^{К}\vee y_1^{Х})y_2^{У}y_3^{М}=s^2;$$
$$y_1^{У}y_2^{У}y_3^{Т}=s^3;\ (y_1^{Ж}\vee y_1^{Ч}\vee y_1^{Ш}\vee y_1^{Щ})y_2^{У}y_3^{М}=s^4;$$
$$(y_1^{Е}\vee y_1^{Н}\vee y_1^{Р}\vee y_1^{С})y_2^{У}y_3^{М}=s^5;\ (y_1^{Б}\vee y_1^{В}\vee y_1^{Д}\vee$$
$$\vee y_1^{З}\vee y_1^{Л}\vee y_1^{М}\vee y_1^{П}\vee y_1^{Т}\vee y_1^{Н}\vee y_1^{Р}\vee y_1^{С})y_2^{Б}y_3^{Т}=s^6;$$
$$(y_1^{Ж}\vee y_1^{Ч}\vee y_1^{Ш}\vee y_1^{Щ}\vee y_1^{Г}\vee y_1^{К}\vee y_1^{Х})y_2^{Б}y_3^{Т}=s^7.$$

Производим бинаризацию функции $g(y_1, y_2, y_3) = s$. Записываем отношение P_{15}, связывающее предметные переменные y_1 и s:

$$P_{15}(y_1, s) = (y_1^{Б} \vee y_1^{В} \vee y_1^{Д} \vee y_1^{З} \vee y_1^{Л} \vee y_1^{М} \vee y_1^{П} \vee y_1^{Т}) \wedge$$
$$\wedge (s^1 \vee s^6) \vee y_1^{Е} s^5 \vee (y_1^{Н} \vee y_1^{Р} \vee y_1^{С})(s^1 \vee s^5 \vee s^6) \vee$$
$$\vee (y_1^{Ж} \vee y_1^{Ч} \vee y_1^{Ш} \vee y_1^{Щ})(s^4 \vee s^7) \vee y_1^{Ц} s^3 \vee$$
$$\vee (y_1^{Г} \vee y_1^{К} \vee y_1^{Х})(s^2 \vee s^7).$$

Рис. 15

Записываем отношение P_{16}, связывающее предметные переменные y_2 и s:

$$P_{16}(y_2, s) = y_2^{У}(s^1 \vee s^2 \vee s^3 \vee s^4 \vee s^5) \vee y_2^{Б}(s^6 \vee s^7).$$

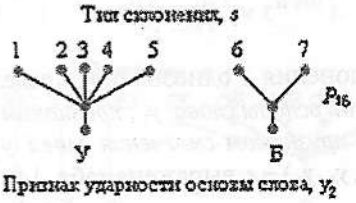


Рис. 16

Записываем отношение P_{17}, связывающее предметные переменные y_3 и s:

$$P_{17}(y_3, s) = y_3^{Т}(s^1 \vee s^3 \vee s^6 \vee s^7) \vee \vee y_3^{М}(s^2 \vee s^4 \vee s^5).$$

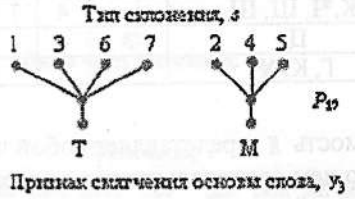


Рис. 17

Полноценная бинаризация функции g получилась даже без привлечения перенумерации ячеек табл. 14.

3. Логическая сеть

Мы построили *математическую модель склонения полных непритяжательных имен прилагательных*. Она характеризуется системой бинарных отношений $P_1 - P_{17}$, задаваемых двудольными графами и формулами соответствующих предикатов. Образуя конъюнкцию всех этих предикатов, получаем *предикат модели*:

$$P(x_1, x_2, x_3, x_4, x_5, r, s, y_1, y_2, y_3, z_л, z_п, z, z_0, z_1, z_2, z_3) =$$
$$= P_1(x_1, r) \wedge P_2(x_2, r) \wedge P_3(x_3, r) \wedge P_4(x_4, r) \wedge P_5(x_5, r) \wedge$$
$$\wedge P_6(z_л, r) \wedge P_7(z_п, r) \wedge P_8(z_л, z) \wedge P_9(z_п, z) \wedge P_{10}(z_л, z_0) \wedge$$
$$\wedge P_{11}(z_л, z_1) \wedge P_{12}(z_п, z_2) \wedge P_{13}(z_п, z_3) \wedge P_{14}(s, z_л) \wedge$$
$$\wedge P_{15}(y_1, s) \wedge P_{16}(y_2, s) \wedge P_{17}(y_3, s).$$

Предикату модели соответствует *отношение модели P*, связывающее между собой предметные переменные $x_1, x_2, x_3, x_4, x_5, r, s, y_1, y_2, y_3, z_л, z_п, z, z_0, z_1, z_2, z_3$. Отношение модели P можно наглядно изобразить в виде сети, называемой *логической*. Логическую сеть можно построить для любой модели. В нашем конкретном случае получаем *логическую сеть склонения полных непритяжательных имен прилагательных*. Она изображена на рис. 18.

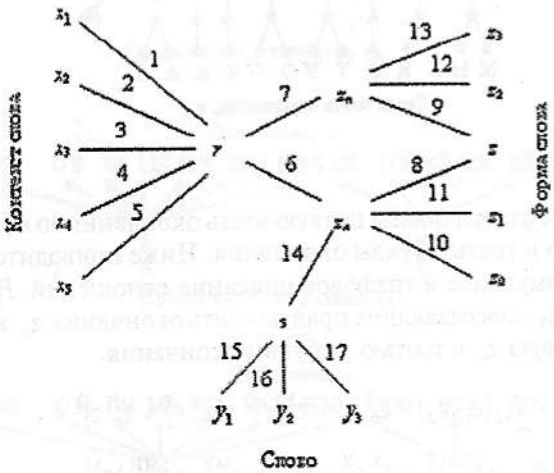


Рис. 18

Рассмотрим общие принципы построения логической сети. Каждой модели соответствует своя логическая сеть. Любая логическая сеть состоит из *полюсов* и *ветвей*. Каждому полюсу логической сети ставится в соответствие своя предметная переменная модели, которая называется *атрибутом* этого полюса. Каждый полюс обозначается своей предметной переменной. С каждым полюсом связываем его *домен*, то есть область изменения атрибута этого полюса. Любой полюс логической сети в каждый момент времени несет какое-то знание о значении своего атрибута. Это знание называется *состоянием полюса*. Оно представляет собой одно из подмножеств домена полюса. Указывая состояния всех полюсов сети в данный момент времени, получаем *состояние сети* в тот же момент времени.

Области изменения всех предметных переменных для сети склонения полных непритяжательных имен прилагательных были неявно охарактеризованы ранее во второй теме этого раздела. Теперь мы их все перечислим в явном виде. Область изменения переменной x_1 обозначаем символом A_1, имеем: $A_1 = \{М, Ж, С\}$; x_2 – $A_2 = \{Е, М\}$; x_3 –

A_3 = {И, Р, Д, В, Т, П}; $x_4 - A_4$ = {Н, О}; $x_5 - A_5$ = {С, А}; $r - A_6$ = {1, 2, ..., 19}; $s - A_7$ = {1, 2, ..., 7}; $y_1 - A_8$ = {Б, В, Г, Д, Е, Ж, З, К, Л, М, Н, П, Р, С, Т, Х, Ц, Ч, Ш, Щ}; $y_2 - A_9$ = {Б, У}; $y_3 - A_{10}$ = {Т, М}; $z_л - A_{11}$ = {А, А′, Я, У, У′, Ю, О, О′, Е, Ы, Ы′, И, И′}; $z_п - A_{12}$ = {Я, Ю, Е, Й, М, ГО, МУ, Х, МИ}; $z - A_{13}$ = {АЯ, А′Я, ЯЯ, УЮ, У′Ю, ЮЮ, ОЕ, О′Е, ЕЕ, ОЙ, О′Й, ЕЙ, ОМ, О′М, ЕМ, ОГО, О′ГО, ЕГО, ОМУ, О′МУ, ЕМУ, ОЮ, О′Ю, ЕЮ, ЫЙ, ИЙ, ЫМ, Ы′М, ИМ, И′М, ЫЕ, Ы′Е, ИЕ, И′Е, ЫХ, Ы′Х, ИХ, И′Х, ЫМИ, Ы′МИ, ИМИ, И′МИ}; $z_0 - A_{14}$ = {*, ′}; $z_1 - A_{15}$ = {А, Я, У, Ю, О, Е, Ы, И}; $z_2 - A_{16}$ = {Я, Ю, Е, Й, М, Г, Х}; $z_3 - A_{17}$ = {*, О, У, И}.

Каждой ветви логической сети ставится в соответствие свое бинарное отношение модели, которое называется *отношением* этой ветви. Каждая ветвь обозначается номером своего отношения. Она соединяет два полюса, отвечающие тем предметным переменным, которые связываются отношением, соответствующим данной ветви. Например ветвь 1 соответствует отношению $P_1(x_1, r)$, поэтому она соединяет полюсы x_1 и r. Всего в построенной нами сети имеется 17 полюсов и 17 ветвей. Случайно получилось так, что число полюсов сети совпало с числом ее ветвей, но такое совпадение не обязательно. Полюсы сети делятся на два класса – *внешние* и *внутренние*. Каждый внешний полюс соединен лишь с одной ветвью, каждый внутренний – более, чем с одной. В построенной нами сети к внешним относятся полюсы, помеченные предметными переменными $x_1, x_2, x_3, x_4, x_5, y_1, y_2, y_3, z_0, z_1, z_2, z_3$, к внутренним – $r, s, z_л, z_п, z$. Внешние полюсы сети естественно объединить в три группы: 1) x_1, x_2, x_3, x_4, x_5; 2) y_1, y_2, y_3; 3) z_0, z_1, z_2, z_3. В первую группу входят полюсы, помеченные теми предметными переменными, значения которых характеризуют *контекст*, окружающий склоняемое слово. Во вторую группу входят полюсы, характеризующие само склоняемое *слово*. В третью группу входят полюсы, которые характеризуют результат склонения слова, то есть *словоформу*, соответствующую данному контексту и данному слову.

Каждую логическую сеть можно превратить в электронную схему для автоматического решения некоторого класса задач, определяемого той моделью, для которой была построена данная сеть. Ее можно изготовить в виде сверхбыстродействующей карты, осуществляющей параллельную обработку информации. Эта карта устанавливается на материнской плате персонального компьютера. По мере надобности программа, управляющая работой компьютера, обращается к той или иной карте, которая за доли микросекунды формирует ответ на тот или иной ее запрос. Таким образом, персональный компьютер из машины последовательного действия превращается в машину последовательно-параллельного действия. Параллельно действующая часть компьютера будет представлена батареей карт, а последовательно действующая его часть – программой, управляющей работой этих карт. При этом достигается значительное повышение производительности компьютера.

Сравнивая действие такого компьютера с работой мозга человека, можно уподобить действие программы, управляющей работой компьютера, работе сознания человека, а действие батареи карт – работе подсознания. Механизм сознания человека представляет собой сравнительно медленно действующую систему обработки информации последовательного действия. Устная и письменная речь, чтение текста, слуховое восприятие речи, сознательные движения человека – всё это последовательные процессы. Механизмы же подсознания осуществляют параллельную обработку информации, именно они выполняют основной объем выполняемой мозгом работы. Можно сказать, что подсознание выполняет роль оркестра, а сознание – роль дирижера.

Электронную схему, соответствующую сети, изображенной на рис. 18, можно использовать для автоматического решения класса задач, относящихся к склонению полных непритяжательных имен прилагательных. Такие задачи ежеминутно решают люди в процессе анализа и синтеза сообщений на естественном языке. Примером задачи этого класса может служить *задача синтеза словоформы*. Она заключается в следующем: заданы слово и окружающий его контекст, требуется определить соответствующую им словоформу. Решение этой задачи человеком можно продемонстрировать с помощью упражнения, взятого из школьной грамматики русского языка [6, с. 131]. Задано предложение «Как-то давно, (*тёмный осенний*) вечером, случилось мне плыть по (*угрюмый сибирский*) реке» (Короленко). Требуется согласовать прилагательные, приведенные в скобках, с контекстом. В результате решения этой задачи формируем предложение: «Как-то давно, *тёмным осенним* вечером, случилось мне плыть по *угрюмой сибирской* реке».

Другим примером может служить *задача анализа словоформы*: заданы форма слова и окружающий ее контекст, требуется определить грамматические признаки, соответствующие этой словоформе. Из той же грамматики русского языка берем предложение «Над *алыми* цветами *пахучего* шиповника кружились *золотые* жуки» (А. Н. Толстой). Требуется указать род, число и падеж словоформ, выделенных курсивом, то есть определить значения их грамматических признаков x_1, x_2, x_3. В результате решения этой задачи находим: словоформа *алыми* имеет мужской род, множественное число, творительный падеж; словоформа *пахучего* имеет мужской род, единственное число, родительный падеж; словоформа *золотые* имеет мужской род, множественное число, именительный падеж.

Третьим примером может служить *задача нормализации словоформы*: заданы форма слова и окружающий ее контекст, требуется отыскать слово, соответствующее этой словоформе. Берем то же предложение. Требуется указать слова, соответствующие словоформам, выделенным курсивом, то есть отыскать для этих словоформ их словарные формы. В результате решения этой задачи находим: словоформе *алыми* соответствует слово «алый»; словоформе *пахучего* – «пахучий»; словоформе *золотые* – «золотой».

Каждый человек, владеющий русским языком, практически мгновенно без напряжения и без каких-либо затруднений решает подобные задачи на уровне подсознания. Чтобы прочесть заданное предложение и вникнуть в его содержание, человеку, владеющему русским языком, требуется не более трех секунд. Это значит, что на анализ и нормализацию каждой из трех словоформ (а выполнение этих операций, очевидно, необходимо для понимания смысла предложения) расходуется не более секунды. Если же учесть, что морфологической обработке подвергаются и другие слова предложения, и, кроме того, мозг выполняет множество других операций над элементами предложения в процессе его анализа, то мы придем к выводу, что на каждую элементарную операцию по обработке отдельной словоформы расходуется не более 0,1 сек. Тактовая частота, на которой работает мозг, не может превысить 10^3 тактов/сек, значит мозг затрачивает на выполнение каждой элементарной морфологической операции не более $10^3 \times 0{,}1 = 100$ тактов. Современный же компьютер работает на тактовой частоте порядка 10^9 тактов/сек, это значит, что карта, действующая по принципу работы мозга, будет выполнять элементарную морфологическую операцию не более, чем за десятую долю микросекунды ($100 : 10^9 = 10^{-7}$ сек).

4. Работа логической сети

Сеть предназначена для решения системы уравнений, задаваемых соответствующей моделью. В модели склонения полных непритяжательных имен прилагательных содержится 17 уравнений, каждое из которых связывает какие-то две переменные. Аналитически (то есть оперируя с формулами) можно решить любую систему уравнений алгебры предикатов, характеризующую данную модель. При этом можно задавать любое знание о значениях любых предметных переменных и получать знания о значениях любых других предметных переменных. Аналитический метод работает безотказно. Главная наша задача заключается в том, чтобы разработать такие методы построения логических сетей, чтобы эти сети автоматически решали любые системы уравнений алгебры предикатов так же безотказно, как это может делать человек, оперируя формулами алгебры предикатов.

Решая задачу, логическая сеть копирует действия человека, но с той лишь разницей, что человек при этом действует последовательно, а сеть – параллельно. Сеть работает по тактам. Каждый такт делится на два полутакта – первый и второй. В первом полутакте i-го такта сеть для каждого из своих уравнений вида $K(x,y)=1$ (K – это отношение, заданное таким уравнением) отыскивает: 1) по известному знанию $P_i(x)$ о значении переменной x в начале i-го такта знание $Q_i'(y)$ о значении переменной y в конце i-го такта; 2) по известному знанию $Q_i(y)$ о значении переменной y в начале i-го такта знание $P_i'(x)$ о значении переменной x в конце i-го такта. Математически эти две операции выражаются формулами:

$$\exists\ x \in A\ \ (K(x,y)P_i(x)) = Q_i'(y);$$

$$\exists\ x \in B\ \ (K(x,y)Q_i(y)) = P_i'(x).$$

Здесь A и B – области изменения переменных x и y. Как видим, каждая ветвь сети – это дорога с двусторонним движением.

Во втором полутакте каждого такта сеть отыскивает общую часть $P_{i+1}(x)$ всех знаний $P_{i1}'(x), P_{i2}'(x), \ldots, P_{il}'(x)$ о значении каждой из своих предметных переменных x, поступающих по ветвям сети со всех сторон к полюсу x. Выражается эта операция следующим образом:

$$P_{i1}'(x) \wedge P_{i2}'(x) \wedge \ldots \wedge P_{il}'(x) = P_{i+1}(x).$$

Полученное знание $P_{i+1}(x)$ затем используется в роли состояния полюса x в начальный момент $i+1$-го такта. Символ l обозначает число ветвей, подходящих к полюсу x. К началу $i+1$-го такта в каждом полюсе формируется знание-множество $P_{i+1}(x)$, которое всегда оказывается включенным в знание-множество $P_i(x)$, которое содержалось в том же полюсе в начале i-го такта. Так что единственным результатом работы логической сети является уточнение знаний, содержащихся во всех ее полюсах в соответствии с исходными данными.

Рассматриваем работу построенной нами конкретной логической сети по тактам.

Начальное состояние сети: $x_1 \in$ {М, Ж, С}; $x_2 \in$ {Е, М}; $x_3 \in$ {И, Р, Д, В, Т, П}; $x_4 \in$ {Н, О}; $x_5 \in$ {С, А}; $r \in$ {1, 2, ..., 19}; $s \in$ {1, 2, ..., 7}; $y_1 \in$ {Б, В, Г, Д, Е, Ж, З, К, Л, М, Н, П, Р, С, Т, Х, Ц, Ч, Ш, Щ}; $y_2 \in$ {Б, У}; $y_3 \in$ {Т, М}; $z_л \in$ {А, А′, Я, У, У′, Ю, О, О′, Е, Ы, Ы′, И, И′}; $z_п \in$ {Я, Ю, Е, Й, М, ГО, МУ, Х, МИ}; $z \in$ {АЯ, А′Я, ЯЯ, УЮ, У′Ю, ЮЮ, ОЕ, О′Е, ЕЕ, ОЙ, О′Й, ЕЙ, ОМ, О′М, ЕМ, ОГО, О′ГО, ЕГО, ОМУ, О′МУ, ЕМУ, ОЮ, О′Ю, ЕЮ, ЫЙ, ИЙ, ЫМ, Ы′М, ИМ, И′М, ЫЕ, Ы′Е, ИЕ, И′Е, ЫХ, Ы′Х, ИХ, И′Х, ЫМИ, Ы′МИ, ИМИ, И′МИ}; $z_0 \in$ {*, '}; $z_1 \in$ {А, Я, У, Ю, О, Е, Ы, И}; $z_2 \in$ {Я, Ю, Е, Й, М, Г, Х}; $z_3 \in$ {*, О, У, И}.

Первый такт.

Первый полутакт. Сужение множеств. Ввод в сеть исходных данных: $x_3 \in$ {В}, $y_1 \in$ {Ж}, $y_2 \in$ {Б}, $z_2 \in$ {Й}.

Второй полутакт. Пересечение множеств.

В результате получаем *состояние сети к началу второго такта:* $x_1 \in \{М, Ж, С\}$; $x_2 \in \{Е, М\}$; $x_3 \in \{В\}$; $x_4 \in \{Н, О\}$; $x_5 \in \{С, А\}$; $r \in \{1, 2, ..., 19\}$; $s \in \{1, 2, ..., 7\}$; $y_1 \in \{Ж\}$; $y_2 \in \{Б\}$; $y_3 \in \{Т, М\}$; $z_л \in \{А, А', Я, У, У', Ю, О, О', Е, Ы, Ы', И, И'\}$; $z_п \in \{Я, Ю, Е, Й, М, ГО, МУ, Х, МИ\}$; $z \in \{АЯ, А'Я, ЯЯ, УЮ, У'Ю, ЮЮ, ОЕ, О'Е, ЕЕ, ОЙ, О'Й, ЕЙ, ОМ, О'М, ЕМ, ОГО, О'ГО, ЕГО, ОМУ, О'МУ, ЕМУ, ОЮ, О'Ю, ЕЮ, ЫЙ, ИЙ, ЫМ, Ы'М, ИМ, И'М, ЫЕ, Ы'Е, ИЕ, И'Е, ЫХ, Ы'Х, ИХ, И'Х, ЫМИ, Ы'МИ, ИМИ, И'МИ\}$; $z_0 \in \{*, '\}$; $z_1 \in \{А, Я, У, Ю, О, Е, Ы, И\}$; $z_2 \in \{Й\}$; $z_3 \in \{*, О, У, И\}$.

Второй такт.

Первый полутакт. Сужение множеств. 1) Движемся от полюса x_3 к полюсу r ($x_3 \to r$). Из отношения P_3 получаем: $(P_3:) r \in \{4, 5, 10, 13, 17, 18\}$. 2) $y_1 \to s$, P_{15}: $s \in \{4, 7\}$. 3) $y_2 \to s$, P_{16}: $s \in \{6, 7\}$. 4) $y_3 \to s$, P_{17}: $s \in \{1, 3, 6, 7\}$. 5) $z \to z_п$, P_{12}: $z_п \in \{Й\}$.

Второй полутакт: Пересечение множеств.

Состояние сети к началу третьего такта: $x_1 \in \{М, Ж, С\}$; $x_2 \in \{Е, М\}$; $x_3 \in \{В\}$; $x_4 \in \{Н, О\}$; $x_5 \in \{С, А\}$; $r \in \{4, 5, 10, 13, 17, 18\}$; $s \in \{7\}$; $y_1 \in \{Ж\}$; $y_2 \in \{Б\}$; $y_3 \in \{Т, М\}$; $z_л \in \{А, А', Я, У, У', Ю, О, О', Е, Ы, Ы', И, И'\}$; $z_п \in \{Й\}$; $z \in \{АЯ, А'Я, ЯЯ, УЮ, У'Ю, ЮЮ, ОЕ, О'Е, ЕЕ, ОЙ, О'Й, ЕЙ, ОМ, О'М, ЕМ, ОГО, О'ГО, ЕГО, ОМУ, О'МУ, ЕМУ, ОЮ, О'Ю, ЕЮ, ЫЙ, ИЙ, ЫМ, Ы'М, ИМ, И'М, ЫЕ, Ы'Е, ИЕ, И'Е, ЫХ, Ы'Х, ИХ, И'Х, ЫМИ, Ы'МИ, ИМИ, И'МИ\}$; $z_0 \in \{*, '\}$; $z_1 \in \{А, Я, У, Ю, О, Е, Ы, И\}$; $z_2 \in \{Й\}$; $z_3 \in \{*, О, У, И\}$.

Третий такт.

Первый полутакт. Сужение множеств. 1) $z_п \to r$, P_7: $r \in \{1, 4, 9, 11\}$. 2) $s \to z_л$, P_{14}: $z_л \in \{А', У', О', И'\}$. 3) $z_п \to z$, P_{14}: $z \in \{ОЙ, О'Й, ЕЙ, ЫЙ, ИЙ\}$. 4) $z_п \to z_3$, P_{14}: $z_3 \in \{*\}$. 5) $s \to y_3$, P_{17}: $y_3 \in \{Т\}$.

Второй полутакт: Пересечение множеств.

Состояние сети к началу четвертого такта: $x_1 \in \{М, Ж, С\}$; $x_2 \in \{Е, М\}$; $x_3 \in \{В\}$; $x_4 \in \{Н, О\}$; $x_5 \in \{С, А\}$; $r \in \{4\}$; $s \in \{7\}$; $y_1 \in \{Ж\}$; $y_2 \in \{Б\}$; $y_3 \in \{Т\}$; $z_л \in \{А', У', О', И'\}$; $z_п \in \{Й\}$; $z \in \{ОЙ, О'Й, ЕЙ, ЫЙ, ИЙ\}$; $z_0 \in \{*, '\}$; $z_1 \in \{А, Я, У, Ю, О, Е, Ы, И\}$; $z_2 \in \{Й\}$; $z_3 \in \{*\}$.

Четвертый такт.

Первый полутакт. Сужение множеств. 1) $r \to x_1$, P_1: $x_1 \in \{М\}$. 2) $r \to x_2$, P_2: $x_2 \in \{Е\}$. 3) $r \to x_4$, P_4: $x_4 \in \{Н\}$. 4) $z_л \to z$, P_8: $z \in \{А'Я, У'Ю, О'Е, О'Й, О'М, О'ГО, О'МУ, О'Ю, И'М, И'Е, И'Х, И'МИ\}$. 5) $z \to z_л$, P_8: $z_л \in \{О, О', Е, Ы, И\}$.

Второй полутакт: Пересечение множеств.

Состояние сети к началу пятого такта: $x_1 \in \{М\}$; $x_2 \in \{Е\}$; $x_3 \in \{В\}$; $x_4 \in \{Н\}$; $x_5 \in \{С, А\}$; $r \in \{4\}$; $s \in \{7\}$; $y_1 \in \{Ж\}$; $y_2 \in \{Б\}$; $y_3 \in \{Т\}$; $z_л \in \{О'\}$; $z_п \in \{Й\}$; $z \in \{ОЙ\}$; $z_0 \in \{*, '\}$; $z_1 \in \{А, Я, У, Ю, О, Е, Ы, И\}$; $z_2 \in \{Й\}$; $z_3 \in \{*\}$.

Пятый такт.

Первый полутакт. Сужение множеств. 1) $z_л \to z_1$, P_{11}: $z_1 \in \{О\}$. 2) $z_п \to z_0$, P_{10}: $z_0 \in \{'\}$.

Второй полутакт: Пересечение множеств.

Состояние сети к началу шестого такта: $x_1 \in \{М\}$; $x_2 \in \{Е\}$; $x_3 \in \{В\}$; $x_4 \in \{Н\}$; $x_5 \in \{С, А\}$; $r \in \{4\}$; $s \in \{7\}$; $y_1 \in \{Ж\}$; $y_2 \in \{Б\}$; $y_3 \in \{Т\}$; $z_л \in \{О'\}$; $z_п \in \{Й\}$; $z \in \{ОЙ\}$; $z_0 \in \{'\}$; $z_1 \in \{О\}$; $z_2 \in \{Й\}$; $z_3 \in \{*\}$.

Сеть отыскала однозначные значения всех предметных переменных, кроме переменной x_5. В последующих тактах состояние сети уже не меняется. Именно этот факт служит сигналом к прекращению работы сети. После получения сетью результата можно осуществить съем информации с полюсов сети. Съем информации можно производить не со всех полюсов сети, а только с тех, которые интересуют пользователя.

Сеть выполняет функции своего рода базы данных. В роли атрибутов базы данных в сети выступают предметные переменные, представленные полюсами сети. В роли доменов атрибутов базы данных в сети выступают области изменения предметных переменных сети. В роли таблиц, хранящихся в базе данных выступают бинарные отношения, представленные ветвями сети. В роли исходных данных, поступающих в базу данных, для решаемой задачи, выступают подмножества некоторых из областей изменения предметных переменных сети. В роли запроса в базу данных выступают некоторые из предметных переменных сети. Пользователь может указать любое множество интересующих его предметных переменных. Этим предметным переменным соответствуют те полюсы, с которых производится съем информации, после того, как сеть решила поставленную перед нею задачу. Информация, выдаваемая сетью, имеет вид подмножеств областей изменения предметных переменных, указанных пользователем сети.

Изменения в полюсах распространяются по сети волнами. Они прекращаются после того, как в конце некоторого такта знания во всех полюсах оказываются такими же, как и в его начале. Рано или поздно такой момент с неизбежностью наступает, поскольку в каждом полюсе по прошествии очередного такта конечное множество-знание превращается в его подмножество. Стабилизация знаний в полюсах сети сигнализирует о том, что сеть завершила решение задачи. Теперь остается только произвести считывание полученных знаний с интересующих пользователя полюсов.

Таким образом, реализация данного подхода для различных механизмов естественного языка и их воспроизведение средствами вычислительной техники позволит перейти на новый уровень проектирования искусственного интеллекта.

Список литературы: 1. *Шрейдер Ю.А.* О понятии «математическая модель языка». М.: Знание, 1971. 63 с. 2. *Широков В.А.* Феноменологія лексикографічних систем. К.: Наукова думка, 2004. 327 с. 3. Грамматика русского языка. М.: АН СССР. 1960. Т.1. 719 с.; Т.2. Ч.1, 702 с.; Ч.2, 440 с. 4. Русская грамматика. М.: Наука, 1980, Т.1, 783 с.; Т.2, 709 с. 5. *Зализняк А. А.* Грамматический словарь русского языка. М.: Русский язык, 1977. 879 с. 6. *Бархударов С.Г., Крючков С.Е.* Учебник русского языка. М.: Просвещение. 1970. 296 с.

Поступила в редколлегию 23.11.2004

УДК 519.7

И.Д. Вечирская, З.В. Дударь, А.А. Иванилов, В.А. Лещинский

ЛИНЕЙНЫЕ ЛОГИЧЕСКИЕ ОПЕРАТОРЫ В ВИДЕ СХЕМ И ГРАФОВ

В [1] был обозначен подход к построению высокопроизводительных ЭВМ нового типа, основанных на параллельной архитектуре и ориентированных на обработку логических данных и естественно-языковой информации. Для ЭВМ такого типа введен термин «мозгоподобная». Основой мозгоподобных ЭВМ призвана стать аппаратная реализация новой алгебрологической конструкции, называемой логической сетью. В [1] приведен пример логической сети, которая описана системой бинарных предикатов. Однако, чтобы с помощью сети можно было извлекать знания из описываемой ею модели, необходимо решать логические уравнения, выраженные бинарными предикатами. Сами по себе предикаты не позволяют этого делать, они лишь описывают модель.

Важным логико-математическим инструментом, позволяющим решать логические уравнения, являются линейные логические операторы [2, 3]. Когда в логическом уравнении с двумя неизвестными задается множество значений одной предметной переменной, то линейный логический оператор находит множество значений второй переменной [2, 3]. Поэтому функционирующая логическая сеть представляет собой систему взаимодействующих операторов. Таким образом, одним из базовых элементов мозгоподобной ЭВМ призвана стать аппаратная реализация линейного логического оператора.

В свете вышесказанного актуальным становится вопрос о преобразовании математической формулы линейного логического оператора в техническую схему. После этого логическую сеть можно реализовать в виде устройства, состоящего из взаимодействующих линейных логических операторов.

К настоящему времени уже имеется проект конкретной логической сети в виде описания такого устройства на языке VHDL. На кафедре автоматизации проектирования вычислительной техники Харьковского национального университета радиоэлектроники разработана специализированная интегральная схема логической сети, описывающей связь грамматических признаков с окончаниями полных непритяжательных имен прилагательных русского языка [4]. Данная логическая сеть представлена на рис. 1 в виде неориентированного графа.

Каждая дуга на рис. 1 разворачивается в двудольный граф, подобный изображенному на рис. 3. Далее на основе двудольного графа можно построить пару линейных логических операторов (рис. 4, 5) и реализовать их аппаратно. Вершины выражают следующие признаки имени прилагательного: y_2 – род; y_1 –число; y_3 – падеж; t – архаичность; y_4 – одушевленность; v – ячейка парадигмы; y – класс окончания; x_1 – первая буква окончания; x_2 – вторая буква окончания; x_3 – третья буква окончания; x – окончание; u_1 – ударность; u_2 – смягчение; u_4 – последняя буква основы; u – тип основы.

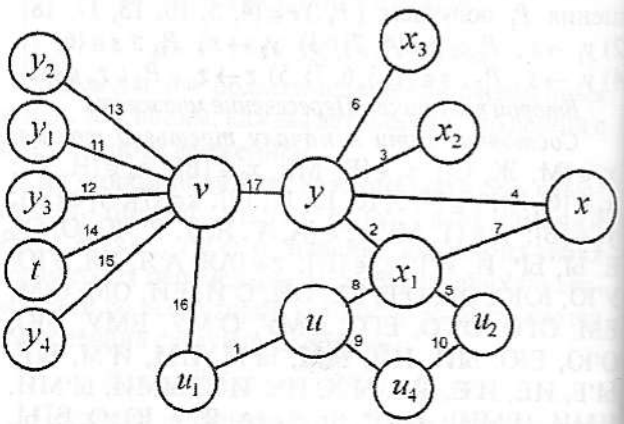


Рис. 1

Данная логическая сеть отвечает на различные запросы относительно прилагательных. Сеть может, например, решать задачу склонения: она установит окончание (или несколько окончаний) некоторого прилагательного, если указать все (или некоторые) его грамматические признаки. Сеть решает и обратную задачу, а также множество похожих задач, связанных с прилагательными.

Цель данной статьи – указать способ технической реализации линейного логического оператора. Средством математического описания при этом служит аппарат алгебры предикатов [5, 6, 7, 8], а средством технической реализации – комбинационные схемы, которые в [6] называются переключательными цепями.

Поскольку переключательная цепь линейного логического оператора служит удачным обобщением переключательной цепи многозначного отображения, сначала приведем математическое описание и реализацию для отображения, а затем – для оператора.

Пусть $M=\{a_1,a_2,...,a_k\}$ – фиксированное множество, состоящее из k элементов; A – какое-нибудь из его подмножеств $A \subseteq M$. Для множества A составляем набор $(\alpha_1,\alpha_2,...,\alpha_k)$ логических элементов $\alpha_1,\alpha_2,...,\alpha_k$ по следующему правилу: если

$\alpha_i \in A$, то $\alpha_i = 1$; если же $\alpha_i \notin A$, то $\alpha_i = 0$, $(i = \overline{1,k})$. Набор $(\alpha_1, \alpha_2, ..., \alpha_k)$ называется *характеристикой или координатным представлением* множества A, а логические элементы $\alpha_1, \alpha_2, ..., \alpha_k$ – *координатами* множества A. Пусть, например, $M = \{a,b,c,d\}$, $A = \{c,d\}$. Тогда характеристика множества A равна $(0,0,1,1)$. Можно по характеристике найти множество: если $M = \{a,b,c,d\}$ и характеристика A равна $(1,0,0,0)$, то $A = \{a\}$.

Произвольное отображение (в общем случае многозначное)

$$f(x_1, x_2, ..., x_n) = y, \ \left(y \in N, N = \{b_1, b_2, ..., b_k\}\right)$$
$$M_1 \times M_2 \times ... \times M_n \to N$$

ставит в соответствие набору значений аргументов $(x_1, x_2, ..., x_n)$ множество $B \subseteq N$, равное

$$B = \left\{ y \middle| F(x_1, x_2, ..., x_n, y) = 1 \right\},$$

где F – предикат, *соответствующий* отображению f или *характеристическая функция* отображения f (если n-местное отображение f рассматривать как $n+1$-местное отношение F). Пусть $\beta_1, \beta_2, ..., \beta_l$ – координаты множества B. Тогда

$$\beta_1 = F(x_1, x_2, ..., x_n, b_1),$$
$$\beta_2 = F(x_1, x_2, ..., x_n, b_2),$$
$$.....$$
$$\beta_l = F(x_1, x_2, ..., x_n, b_l).$$

Переключательную цепь удобно строить, когда отображение приведено к данному виду. Приведем пример построения схемы, реализующей какое-нибудь отображение. Берем предикат, связывающий род с первой буквой трехбуквенных окончаний полных непритяжательных имен прилагательных:

$$F(x,y) = x^а y^{ж} \vee x^{я} y^{ж} \vee x^{у} y^{ж} \vee x^{ю} y^{ж} \vee x^{о} y^{ж} \vee x^{о} y^{м} \vee x^{о} y^{с} \vee x^{е} y^{ж} \vee x^{е} y^{м} \vee x^{е} y^{с} \vee x^{ы} y^{ж} \vee x^{ы} y^{м} \vee x^{ы} y^{с} \vee x^{и} y^{ж} \vee x^{и} y^{м} \vee x^{и} y^{с}. \quad (1)$$

Строим соответствующее ему отображение $f(x) = y$. Принимаем $M = \{а,я,у,ю,о,е,ы,и\}$, $N = \{ж,м,с\}$. Имеем:

$$\beta_{ж} = x^{у} \vee x^{ю} \vee x^{а} \vee x^{я} \vee x^{о} \vee x^{е} \vee x^{ы} \vee x^{и},$$
$$\beta_{м} = x^{о} \vee x^{е} \vee x^{ы} \vee x^{и}, \quad (2)$$
$$\beta_{с} = x^{о} \vee x^{е} \vee x^{ы} \vee x^{и}.$$

Схема, реализующая отображение f, имеет вид, показанный на рис. 2.

Схема находит полный образ предмета. Пусть, например $x = e$. Тогда $\beta_{ж} = 1$, $\beta_{м} = 1$, $\beta_{с} = 1$; $y \in \{ж,м,с\}$. То есть полным образом предмета e является все множество N. Для нахождения прообраза нужно строить схему, действующую в обратную сторону. Исходную и обратную схемы легко строить по двудольному графу. Узлы на входе графа заменяются разветвлением проводов. При переходе от исходной схемы к схеме, действующей в обратную сторону, элементы разделения заменяются разветвлениями проводов, а разветвления проводов – элементами разделения. Узлы на выходе графа превращаются в узловые точки схемы.

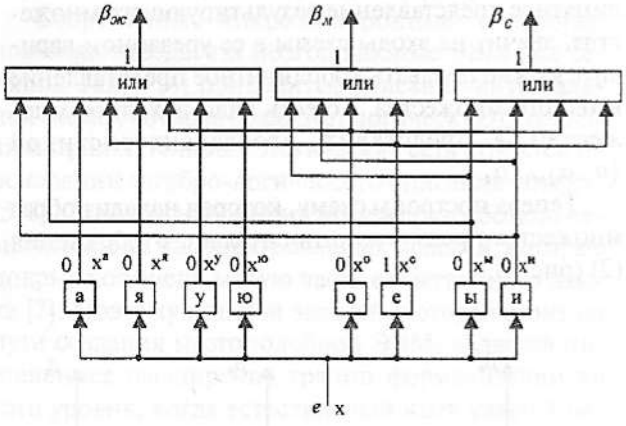
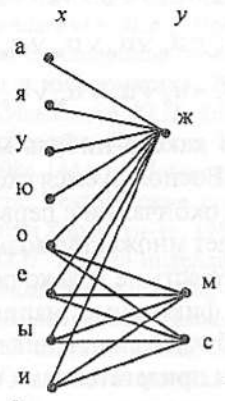


Рис. 2

Двудольный граф строится по бинарному отношению или предикату. Двудольный граф для бинарного предиката $F(x,y)$ изображен на рис. 3. По двудольному графу легко находить полный образ и прообраз предмета вручную. Для этого не нужно строить исходную и обратную переключательные цепи – одного графа достаточно. При этом можно заметить, что двудольный граф позволяет легко находить полный образ не только одного, но и нескольких предметов, то есть можно находить образ множества $A \subseteq M$. То же самое и в другую сторону: можно находить прообраз некоторого множества $B \subseteq N$.



F(x, y)

Рис. 3

Для того, чтобы находить образ множества, переключательная цепь, показанная на рис. 2, не годится. После прохождения входного сигнала через элементы узнавания предмета только по одному из проводов пойдет сигнал 1, по остальным – 0. Между тем, если «отрезать» всю нижнюю часть схемы с элементами узнавания предмета, то единичные сигналы можно будет пустить хоть по всем проводам. Полученная таким образом схема устроена так, что никакой коллизии не произойдет. В этом случае на входные двоичные коды схема будет реагировать выходными. Но на выходе мы регистрируем координатное представление результирующего множества, значит на входы схемы в ее урезанном варианте можно подавать координатное представление входного множества. То есть надо исходное множество A представить его характеристикой $(\alpha_1,\alpha_2,...,\alpha_k)$.

Теперь построим схему, которая находит образ множества предметов относительно отображения (2) (рис. 4).

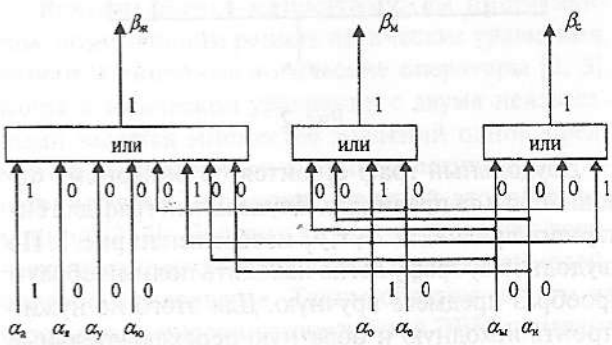


Рис. 4

Для того, чтобы описать эту переключательную цепь математически, достаточно в выражении (2) заменить $x^а$, $x^я$, $x^у$, $x^ю$, $x^о$, $x^е$, $x^ы$, $x^и$ на $\alpha_а$, $\alpha_я$, $\alpha_у$, $\alpha_ю$, $\alpha_о$, $\alpha_е$, $\alpha_ы$, $\alpha_и$ соответственно:

$$\begin{aligned}\beta_{ж} &= \alpha_а \vee \alpha_я \vee \alpha_у \vee \alpha_ю \vee \alpha_о \vee \alpha_е \vee \alpha_ы \vee \alpha_и,\\ \beta_{м} &= \alpha_о \vee \alpha_е \vee \alpha_ы \vee \alpha_и,\\ \beta_{с} &= \alpha_о \vee \alpha_е \vee \alpha_ы \vee \alpha_и.\end{aligned} \tag{3}$$

Найдем образ какого-нибудь множества для отображения (2). Воспользуемся схемой на рис. 4. Берем множество окончаний с первой буквой *а* и *о*. Схема порождает множество возможных родов: $\{а,о\}\to\{ж,м,с\}$, то есть не только реальные окончания рода, но и фиктивные, например, буква *а* оказалась первой буквой окончания полных непритяжательных имен прилагательных среднего рода, хотя таковой на самом деле не является.

Теперь обратимся к линейным логическим операторам [2, 3]. В рассматриваемой ситуации *линейным логическим оператором* $L(A)=B$ является преобразование вида

$$B(y)=\exists x\in M\big(F(x,y)\wedge A(x)\big), \tag{4}$$

которое преобразует одноместные предикаты $A(x)$, заданные на множестве M, в одноместные предикаты $B(y)$, заданные на множестве N. В выражении (4) бинарный предикат $F(x,y)$ фиксирован и задан на $M\times N$, он называется ядром оператора L; квантор существования означает логическую сумму.

Между одноместными предикатами и множествами существует взаимно однозначное соответствие. Предикат $A(x)$ на $M=\{a_1,a_2,...,a_k\}$ соответствующий множеству A с характеристикой $(\alpha_1,\alpha_2,...,\alpha_k)$, запишется формулой:

$$A(x)=\alpha_1x^{a_1}\vee\alpha_2x^{a_2}\vee...\vee\alpha_kx^{a_k}=\bigvee_{i=1}^{k}\alpha_ix^{a_i}.$$

Для предиката $A(x)$ набор $(\alpha_1,\alpha_2,...,\alpha_k)$ называется его *координатным представлением*. Таким образом, когда это удобно, под $A(x)$ можно понимать множество A.

Пользуясь координатным представлением предикатов, можно строить схемы линейных логических операторов. Пусть $M=\{a_1,a_2,...,a_k\}$. Тогда перепишем (4) в развернутом виде:

$$\begin{aligned}B(y)=F(a_1,y)A(a_1)\vee F(a_2,y)A(a_2)\vee...\vee\\ \vee F(a_k,y)A(a_k).\end{aligned}$$

Пусть $N=\{b_1,b_2,...,b_l\}$. Тогда (4) примет еще более развернутый вид:

$$\begin{aligned}B(b_1)&=F(a_1,b_1)A(a_1)\vee F(a_2,b_1)A(a_2)\vee...\\ &\vee F(a_k,b_1)A(a_k),\\ &...\\ B(b_l)&=F(a_1,b_l)A(a_1)\vee F(a_2,b_l)A(a_2)\vee...\\ &\vee F(a_k,b_l)A(a_k).\end{aligned} \tag{5}$$

Пусть $(\alpha_1,\alpha_2,...,\alpha_k)$ – характеристика множества A, $A\subseteq M$; $(\beta_1,\beta_2,...,\beta_l)$ – характеристика множества B, $B\subseteq N$. Тогда

$$\begin{aligned}\alpha_1=A(a_1),\alpha_2=A(a_2),...,\alpha_k=A(a_k),\\ \beta_1=B(b_1),\beta_2=B(b_2),...,\beta_l=B(b_l).\end{aligned} \tag{6}$$

С учетом (6) выражение (5) примет вид

$$\begin{aligned}\beta_1&=F(a_1,b_1)\alpha_1\vee F(a_2,b_1)\alpha_2\vee...\vee F(a_k,b_1)\alpha_k,\\ &...\\ \beta_l&=F(a_1,b_l)\alpha_1\vee F(a_2,b_l)\alpha_2\vee...\vee F(a_k,b_l)\alpha_k.\end{aligned} \tag{7}$$

По формуле (7) и строится схема линейного логического оператора. При этом множества A и B

ляются своими характеристиками. Кроме того, линейный логический оператор может быть задан с помощью двудольного графа, такого, как показан на рис. 3.

Выразим по формуле (7) линейный логический оператор, ядром которого является предикат $F(x,y)$ из (1). Имеем

$$\beta_ж = 1\alpha_а \vee 1\alpha_я \vee 1\alpha_у \vee 1\alpha_ю \vee 1\alpha_о \vee 1\alpha_е \vee 1\alpha_ы \vee 1\alpha_и,$$

$$\beta_м = 0\alpha_а \vee 0\alpha_я \vee 0\alpha_у \vee 0\alpha_ю \vee 1\alpha_о \vee 1\alpha_е \vee 1\alpha_ы \vee 1\alpha_и,$$

$$\beta_с = 0\alpha_а \vee 0\alpha_я \vee 0\alpha_у \vee 0\alpha_ю \vee 1\alpha_о \vee 1\alpha_е \vee 1\alpha_ы \vee 1\alpha_и.$$

Результат совпадает с выражением (3). То есть переключательная цепь (рис. 4), отыскивающая образ множества относительно отображения $f(x)=y$, реализует линейный логический оператор с ядром $F(x,y)$. Таким образом, упростив схему для реализации отображения так, чтобы можно было находить образ множества, мы естественно пришли на схему для линейного логического оператора. Затем данный результат подтвердился выкладками. Пользуясь взаимно однозначным соответствием между одноместными предикатами и множествами, можно считать, что линейный логический оператор (4) преобразует множества.

Для нахождения прообраза множества относительно заданного отображения $f(x)=y$ строится схема, действующая в обратную сторону. Это линейный логический оператор вида:

$$A'(x)=\exists y \in N\big(F(x,y)\wedge B(y)\big), \qquad (8)$$

где в общем случае $A'(x)\neq A(x)$.

Построим схему, реализующую отображение, обратное к отображению (2) (рис. 5):

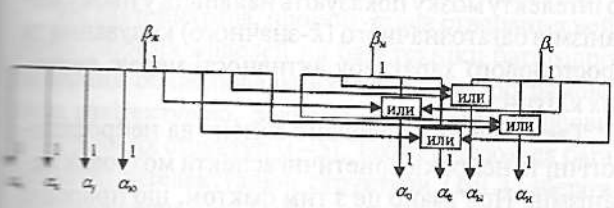


Рис. 5

При движении в обратную сторону элементы разделения превращаются в узлы и наоборот. Найдем образ какого-нибудь множества по данной схеме. По схеме на рис. 4 мы нашли: $\{a,o\}\to\{ж,м,с\}$. Теперь по данной схеме получаем:

$$\{ж,м,с\}\to\{a,я,у,ю,о,е,ы,и\}.$$

Убеждаемся в том, что $A'(x)\neq A(x)$.

Таким образом, при переходе на линейные логические операторы мы избавляемся от элементов описывания предметов и более рационально используем схемы, получая возможность находить не только полные образы предметов, но и образы множеств. Для линейного логического оператора был найден общий способ аппаратной реализации в виде переключательной цепи, которая строится по формуле (7). Полученные схемы позволят строить логические сети мозгоподобных ЭВМ, используя линейный логический оператор в качестве основного строительного элемента. Так, каждая дуга логической сети на рис. 1 представляет собой пару комбинационных схем изображенных на рис. 4, 5 и описывается парой операторов (4), (8).

Современные микроэлектронные технологии позволяют создавать мозгоподобные ЭВМ уже сегодня. Так, в [4] приводится описание интегральной схемы, которая реализует логическую сеть анализа прилагательных. Логическая сеть строится на основании алгебро-логического описания объекта [5, 6]. Однако в настоящий момент разработано незначительное число формальных моделей, которые покрывают очень малую часть естественного языка [7]. Поэтому главной задачей, которая стоит на пути создания мозгоподобной ЭВМ, является постепенное расширение границ формализации до того уровня, когда естественный язык удастся охватить целиком.

Главной отличительной чертой человека, как существа мыслящего, является владение естественным языком. Поэтому можно сказать, что «полноценная» мозгоподобная ЭВМ будет создана лишь тогда, когда будет построена логическая сеть, полностью реализующая механизм естественного языка.

Список литературы: 1. *Бондаренко М.Ф.* / М.Ф. Бондаренко, З.В. Дударь, И.А. Ефимова, В.А. Лещинский, С.Ю. Шабанов-Кушнаренко. О мозгоподобных ЭВМ // Радиоэлектроника и информатика. 2004. № 4. С. 83-99. 2. *Ротин И.М.* Линейные и билинейные логические операторы и их применение в автоматизированных информационных системах. Дисс...канд. техн наук. Харьков: ХТУРЭ. 1994. 163 с. 3. *Гвоздинская Н.А.* / Н.А. Гвоздинская, З.В. Дударь, С.А. Пославский, Ю.П. Шабанов-Кушнаренко. О действиях с линейными логическими операторами // Радиоэлектроника и информатика. 1998. № 3. С. 128–130. 4. *Bondarenko M.F., Hahanova I.V.* Logic networks application for computing process organization // Радиоэлектроника и информатика. 2003. № 3. С. 150-156. 5. *Шабанов-Кушнаренко Ю.П.* Теория интеллекта. Математические средства. Харьков: Вища шк. Изд-во при Харьк. ун-те, 1984. 144 с. 6. *Шабанов-Кушнаренко Ю.П.* Теория интеллекта. Технические средства. Харьков: Вища шк. Изд-во при Харьк. ун-те, 1986. 136 с. 7. *Шабанов-Кушнаренко Ю.П.* Теория интеллекта. Проблемы и перспективы. Харьков: Вища шк. Изд-во при Харьк. ун-те, 1987. 160 с. 8. *Баталин А.В., Тевяшев А.Д., Шабанов-Кушнаренко Ю.П.* О системном анализе информационных процессов // Радиоэлектроника и информатика. 1998. № 3. С. 102–110.

Поступила в редколлегию 14.10.2004

УДК 519.62

Г.Г. Четвериков

АЛГЕБРО-ЛОГІЧНІ ЗАСОБИ МОДЕЛЮВАННЯ ПРИРОДНОЇ МОВИ (ПОВІДОМЛЕННЯ 1)

1. Принципи роботи мозку як відправна точка побудови *k*-значних структур мовних систем штучного інтелекту

1.1. Вступ

Мислителі цивілізації чітко і ясно окреслили найголовнішу проблему і найкоротший шлях до створення систем штучного інтелекту (ШтІ) – моделювання морально-етичних норм і законів. Очевидно, що єдиним, відомим нам, об'єктивним носієм моралі і інтелекту є людина, а виразником, засобом до зовнішнього спілкування, є людська мова. Отже єдино можливим шляхом найоб'єктивнішого, найповнішого та найефективнішого створення систем ШтІ є шлях аналізу, моделювання та синтезу мовного інтелектуального інтерфейсу.

Розвиток обчислювальної техніки є базою автоматизації розумової діяльності, і тому виникло нове поняття штучного інтелекту. Проте, успіхи в галузі інтелектуалізації обчислювальної техніки незначні, особливо, якщо порівнювати досягнуте з очікуваними результатами та прогнозами. Доводиться визнавати, що термін «штучний інтелект» відображає поки що тільки нездійсненні очікування. Слід відмітити, що основні проблеми, які зупинили вчених на перших етапах створення мовного інтелектуального інтерфейсу, до цього часу не вирішені.

На даний час проблеми у розвитку систем штучного інтелекту ставлять питання про застосування орієнтації на досягнення високорівневої технології обробки інформації (тобто отримання нової якості), що виявляється у намаганнях реалізувати на фоннеймановських комп'ютерах системи ШтІ. Отже сучасні, нові вимоги до технології обробки інформації можна звести до таких двох моментів: необхідність вирішення проблем, що погано формалізуються, та наявність кінцевого користувача, який за своєю професійною діяльністю не є програмістом.

Таким чином приходимо до розуміння одного з варіантів на шляху створення систем ШтІ – це шлях аналізу, моделювання та синтезу мовного інтелектуального інтерфейсу за допомогою алгебри скінченних предикатів (АСП), засобів *k*-значної логічної системи та *k*-значних структур і кодування.

Крім цього, ще однією проблемною задачею на сучасному етапі є вирішення завдання створення, розвитку й відродження української науково-технічної термінології, що через історичні обставини є у значній мірі зрусифікованою, а також спотвореною навіть на рівні елементарної фонетики. Мова ж є однією з головних прикмет нації, а культ нації – другим після релігії джерелом сили народу. Розвиток та широке застосування рідної мови є тим важелем, що формує й породжує національну культуру і науку, виробництво та економіку. Нав'язування і використання чужомовної термінології веде до несприйняття суспільством новітніх технологій, занепаду культури та розпаду державності.

Слід відмітити, що одним із найперспективніших напрямків застосування теорії багатозначних структур і кодування в системах штучного інтелекту на практиці є моделювання природної мови. Цей підхід дозволяє поряд з розвитком варіантів програмної реалізації отриманих моделей мови здійснити інший підхід – схемний. Мозок при цьому розглядається як відправна точка побудови *k*-значних просторових структур мовних систем штучного інтелекту. Основні результати, що відображають такі дослідження, наведені у [1, 2, 4, 5].

1.2. Постановка проблеми та аналіз останніх досліджень

Якщо машинам призначено на інтелект, то й «міркувати» вони будуть за тими ж законами, що й людина. Нейрофізіологічні дослідження природного інтелекту мозку показують наявність у ньому механізмів багатозначного (*k*-значного) кодування та просторового характеру активності мереж нервових клітин [1, 2].

Таким чином, зроблено акцент на нейрофізіологічні та нейрокібернетичні аспекти мозкових механізмів. Пов'язано це з тим фактом, що природні нейронні структури із нервових клітин (нейронів) по суті є високоефективними розпізнавальними системами. До того ж ці системи збуджують інтерес не тільки у медичних науковців та фізіологів, а й у фахівців, що займаються питаннями проектування систем штучного інтелекту. Але пряме перенесення результатів дослідження нейрофізіологів до практичної діяльності інженера у даний час неможливе через відсутність відповідної бази. Тому проблеми у розвитку засобів штучного інтелекту, з одного боку, ставлять питання про застосування багатозначного кодування та відповідних структур з просторовими принципами побудови, а з іншого – існують значні проблеми у створенні багатозначних елементів та структур. Відправною точкою у

цих дослідженнях є аналогія поміж багатозначними структурами із просторовими принципами побудови та мікроструктурою кори мозку. Тобто можна провести паралель (описати аналогію) поміж біологічними та технічними схемами.

Мозок – це моделюючий пристрій і більш нічого; це машина широко паралельної дії; це повільнохід, бо кожне нервове волокно може передавати тільки до 1000 імпульсів у секунду. Швидкість руху імпульсів по нервовому волокну незмірно менше, ніж у машини. Тобто усього десятки метрів за секунду, а по дротові машини імпульс рухається зі швидкістю світла – 300 тис. км/с. Дріт здатен передати 10^9 імпульсів за секунду (у сучасних ЕОМ), а може и значно більше (світлові ЕОМ). Мозок має понад 100 млрд. нервових клітин, а кожна з них має приблизно до 10000 нервових волокон [2, 3]. Щільність монтажу елементів в ЕОМ також значна, ступінь мініатюрності – теж. Немає проблеми у створенні ЕОМ з такою ж величезною кількістю елементів. Перевага мозку у його продуктивності. Незважаючи на те, що мозок – повільнохід у порівнянні з ЕОМ, він може дати фори будь-якій ЕОМ. Підраховано, що треба з'єднати 1000 найпотужніших обчислювальних сучасних ЕОМ, щоб одержати інформаційну систему, сумарну за продуктивністю мозку людини. У чому ж перевага мозку? Перевага полягає у просторовому (паралельному) характері його діяльності. А ЕОМ – це система послідовної дії. Отже окремі елементи ЕОМ здатні працювати паралельно й незалежно, як і нейрони мозку. Але цьому заважає принцип програмного управління Джона фон Неймана. Відомо, що ЕОМ здатні працювати за різними принципами: наприклад, машина Т'юрінга працює інакше, але ідея програмного управління у неї присутня.

Як альтернатива, пропонується створення нейроноподібних моделей на базі техніки багатопроцесорних обчислювальних систем із програмованою архітектурою, зокрема, на основі цифрових інтегруючих структур [1–5]. Тобто будуються багатозначні інтелектуальні структури (БІС), зберігаючі структуру фон Неймана, які є істотньо двозначними. Але за їх допомогою моделюються нейронні процеси просторового підсумовування різних за рівнем впливів нейронних мембран, що мають інерційні та порогові властивості, а також зміни частоти проходження повідомлень, що передаються. Але очевидно, що усі перелічені властивості та функції дискретні за часом та багатозначні за рівнем.

Неадекватність принципів кодування та елементної бази щодо моделюючих процесів, які застосовуються, призводить до надлишковості та ускладнення. Ускладнення, що виникають при створенні БІС, допомагають висунути концепцію адекватності багатозначної логіки, зокрема, у вигляді АКП, та структур створення БІС за бажаними властивостями та можливостями.

Відомо, що водночас в ЕОМ працює невелика кількість елементів. Паралельно діючі блоки є, але їх дуже мало. Це, наприклад, багаторозрядний підсумувач зі скрізним перенесенням. Але уся логіка працює послідовно, а мозок працює паралельно, тобто усіма нейронами та волокнами. У цьому є його велика перевага. Якби ЕОМ працювала за принципами мозку, то її продуктивність суттєво б збільшилась.

Як стверджується в [1], мозок є однією з відомих нам складних систем переробки інформації. Достатньо сказати, що у ньому знаходиться близько 100 мільярдів нейронів, кожен з яких має, у середньому, 10 000 зв'язків. При цьому мозок надзвичайно надійний: щоденно гине велика кількість нейронів, але він продовжує функціонувати. Обробка великої кількості інформації здійснюється дуже швидко, за частки секунд. Відомо, що нейрон повільнодіючий елемент. Час реагування – кілька мілісекунд.

Розглянемо перспективи побудови ЕОМ за принципами роботи мозку. Так, у ЕОМ водночас імпульси можуть передаватися по 10^2 дротах, а у мозку – по 10^{12} нервовим волокнам (10^{10} клітин і при кожній – по сотні волокон). Маємо виграш у 10^{10} разів – на розпаралелюванні. У даний час є багатопроцесорні ЕОМ (до тисячі процесорів у кращих машинах). Однак все одно залишається виграш у 10^7 разів за рахунок використання принципів роботи мозку.

Підрахуємо інакше: мозок еквівалентний за потужністю та продуктивністю тисячам найкращих ЕОМ. Одне волокно пропускає до тисячі імпульсів (10^3), а один дріт – 10^9 імпульсів за секунду. Виграш за рахунок машинізації мозку у 10^6 разів. Так що «машинізований» мозок рівнозначний за потужністю 10^9 ЕОМ. Порядок виграшу приблизно такий.

Водночас у мозку обсяг інформації, яка передається, дорівнює 10^{12}(волокон) × 10^3 (імпульсів за секунду)=10^{15} (імпульсів за секунду). Але в кращих ЕОМ – 10^3 (каналів) × 10^9 (імпульсів за секунду) = =10^{12} (імпульсів за секунду). Виходить, що треба об'єднати тисячу найкращих ЕОМ, щоб отримати ЕОМ, рівнозначну мозку за продуктивністю.

На сьогоднішній день маємо багатопроцесорні ЕОМ, але процесорів у таких ЕОМ не дуже багато (сотні, максимум – тисячі). Рекордна багатопроцесорна ЕОМ з рекордною швидкістю роботи була б рівнозначною за продуктивністю мозку. Коли б найкраща багатопроцесорна ЕОМ працювала не послідовно, а паралельно, то її продуктивність дорівнювала б 10^{10} мозкам людини, тобто, дорівнювала б об'єднаній інтелектуальній потужності усього людства.

2. Природна мова як об'єкт математичного опису (мовний інтерфейс – частка інтелектуального)

2.1. Аналіз останніх досліджень

Природна мова як явище інтелектуальної діяльності людини є дуже складним об'єктом. Але маючи її формальний опис, його можна реалізувати на ЕОМ і таким чином прищепити машині здібність володіти природною мовою. Треба щоб машина пристосовувалась до людини та розмовляла її мовою, а не навпаки – людина прилаштовувалась до машини. Природна мова для людини найзручніша. Людина ніколи не проміняє її ні на мову математики, ні на мову програмування. Уявляється, що немає іншої галузі знань, яка б у більшій мірі, ніж ця, сприяла підвищенню темпів комп'ютеризації та інформатизації суспільства. Тоді можна буде прищепити машині здібність до природної мови. Якщо можна буде добути з аналізу мови більш досконалу алгебраїчну мову, це дуже збільшить можливості розроблювачів інформаційних систем, нових інформаційних технологій. Алгебро-логічний апарат, який можна буде добувати з природної мови, дозволятиме розширювати можливості розроблювача, що займається створенням нових інформаційних технологій.

Відомо, що мова з'явилася як особливе відображення паралельної дійсності. Але комп'ютери виникли (історично) у зворотному напрямку: спочатку створювалися і засвоювалися послідовні ЕОМ, а потім почався перехід до паралельних багатопроцесорних обчислювальних систем. Внаслідок цього користувачі-програмісти спочатку оселилися у світі послідовних мов програмування, а зараз перейшли до паралельних мов програмування. Наступний крок на шляху спілкування людини-користувача та ЕОМ – це той ступінь, при якому ЕОМ стане справжнім інтелектуальним помічником, коли вона зможе «зрозуміти» природну мову, тобто момент створення інтелектуального інтерфейсу на підставі природно-мовного інтерфейсу.

Можливо запропонувати наступну узагальнену структуру інтелектуальної системи (ІС), яка задовольняє вимогам нової інформаційної технології рішення задач на ЕОМ. Остання складається з наступних компонентів. Перший компонент – це сукупність засобів інтелектуального інтерфейсу, які мають гнучку структуру, що забезпечує можливість адаптації у широкому спектрі інтересів кінцевих користувачів. Другий – являється сукупністю засобів виконавчої системи, тобто тих програмних та апаратних засобів, які забезпечують обчислювальні, логічні, пошукові та інші аналогічні функції. Третій компонент – це база знань, за допомогою якої здійснюється взаємодія перших двох компонентів (на підставі системи знань, що стосуються проблемного середовища).

Таким чином, основна функція, яку забезпечує інтелектуальний інтерфейс – це спілкування кінцевого користувача з ІС. При цьому використовуються природні для людини форми подання інформації: тексти, зображення, усна мова , а також інші їх комбінації. Однією з головних функцій інтелектуального інтерфейсу є функція підсистеми спілкування на природній мові. Процедури обробки інформації на природній мові утворюють комплекс, що дозволяє як аналізувати, так і синтезувати текст і складають *мовний інтерфейс*.

Відомо, що мова – це система знаків, а мовний знак – це матеріальний носій поняття про предмет. Інакше кажучи, під знаком розуміють матеріальний вираз концепту. Також мова розглядається як засіб програмування розумової діяльності людини. Для ІС зрозуміти значення тексту, який також є видом мовного знаку, означає удавати з себе у формальному вигляді зв'язок поміж знаком та концептом. З цього випливає наступне: для машинного розуміння тексту необхідно вміти формально уявляти поняття, яке виражене текстом, а також відомості (знання) щодо ситуації сповіщення.

Складність формалізації переходу від знака до його значення полягає у тому, що знаків (текстів) нескінченно багато. Перелічити усі зв'язки поміж текстами та їх значеннями неможливо. Вихід є – це виявити найпростіші знаки та їх концепти, а потім формалізувати способи отримання концептів складних знаків із концептів простих знаків.

Мова і мислення людини нерозривно пов'язані поміж собою. Це два взаємодоповнюючих процеси відображення речей (дійсності) у свідомості людини. Мова з'явилася як особливе відображення паралельної реальності і у даний час змогла розвиватися настільки, що сама проявляє тенденцію до незалежного руху та керує діяльністю розуму.

Пізнаючи механізми мови, шляхи її вірного вивчення, людина таким чином намагається розшифрувати та пізнати глибини свого розуму. Такий підхід привів до нового розуміння усієї проблеми. На мову можна дивитися як на деяку алгебраїчну систему. Тобто має право на життя така гіпотеза: природна мова – це деяка алгебра.

2.2. Постановка завдання та його рішення

Отже математичний аналіз механізмів мови дозволяє розглянути її як детермінований, дискретний та скінченний об'єкт. При цьому основна задача полягає у формалізації поняття тексту. Саме поняття тексту теж не просте. Означимо його на рівні інтуїції. Текст розуміємо як математичний опис відношення, яке дозволяє виділити підмножину усіх текстів із деякої множини усіх стрічок, довжина яких обмежена. Таке текстове відношення маємо у вигляді послідовного лінійного ланцюжка символів із букв скінченної абетки. Далі текст

сприймаємо як конструкцію з багатьма рівнями. Так, з букв утворюються морфи, з морфів – слова, із слів – словосполучення, з останніх – речення, з речень – абзаци тощо. І далі окремі частини цієї конструкції розглядатимемо як фрагменти тексту. При цьому кожен фрагмент сполучений з оточуючим його текстом та зазнає на собі його впливу. Дію відкинутої частини тексту замінюють уведенням деякого набору ознак. Останні інтерпретуємо як смисл фрагменту тексту. Відомо, що текст завжди повинен мати цілком визначену думку. Тоді думку, що містить текст, будемо називати смислом.

Таким чином, тексти описують властивості ситуації. Текст відповідає ситуації у тому випадку, якщо він виражає деяку властивість ситуації. При цьому будь-яка думка (це теж властивість ситуації, яких значно більше, ніж ситуацій) формально задається предикатом. Кожна людина являється носієм предикатів тих слів, словосполучень та речень, котрі вона розуміє. Таким чином, смисл речення є предикат, який задає визначений зв'язок $L(X, Y)$ поміж смислом та відповідним йому фрагментом тексту. Тут $X=(x_1,x_2...,x_m)$ змінний вектор смислу, m — число його компонентів, а $y_1,y_2,...,y_n$ — деякий фрагмент тексту, де $y_1,y_2,...,y_n$ — букви, які стоять на $1,2,...,n$ місцях фрагменту. На підставі наданої моделі базується процес формалізації (математичний опис) природної мови, який запропоновано у роботі.

У якості об'єкту дослідження та моделювання в даній роботі вибрана українська мова (точніше – її словоформи). Для дослідника слово – зовсім не такий простий об'єкт. Воно має складну синтаксичну структуру: з букв утворюються морфи (корені, префікси, суфікси, закінчення), а з них – слова. Сусідні морфи впливають один на одного, що веде до їх взаємної видозміни. Існують жорсткі обмеження на сполучення морфів. Розташування та вибір букв у кожному морфі регулюється досить таки складними законами у залежності від змісту слова. Встановлення послідовності слідування морфів у слові також визначається змістом слова за допомогою складних відношень. Слово зв'язане з оточуючим його текстом багатьма синтаксичними відношеннями, зокрема відношеннями до контексту й до всієї системи знань людини. Окрім літерної (абстрактної) форми слова існує ще й звукове (фізичне) його зображення. Звукова та літерна форми також зв'язані між собою достатньо складним чином. Усі ці зв'язки в українській мові ще не вивчені інженерною лінгвістикою до кінця.

3. Висновок

Таким чином, пропонується новий принцип побудови надшвидкодіючих структур мовних систем штучного інтелекту. В основу функціонування таких систем та структур покладено принцип роботи мозку, що при класичній елементній базі (ВІС, НВІС) найбільш пасує до машин Гільберта [2], ніж до існуючих машин Дж.фон Неймана, Е. Поста, А. Т'юринга, які мають принцип програмного управління. Останній реалізується достатньо повільно. Ось де колосальне підвищення продуктивності: до 10^9–10^{10} операцій за секунду. У статті запропонована узагальнена структура інтелектуальної системи, яка відповідає новій інформаційній технології рішення задач на ЕОМ, що орієнтовані на досягнення високорівневої технології обробки мовної інформації (отримання нової якості). Отже, нові вимоги до сучасної технології обробки інформації обумовлені: необхідністю рішення проблем, які погано формалізуються, та наявності користувача, який не є професійним програмістом. Така структура складається з трьох основних компонентів. Мовний інтерфейс розглядаємо як частку інтелектуального на підставі того, що однією з головних функцій інтелектуального інтерфейсу є функція спілкування на природній мові. Перспективи подальших розвідок у даному напрямку полягають у проведенні всебічного аналізу алгебро-логічної структури мови.

Список літератури: 1. *Бондаренко М.Ф., Коноплянко З.Д., Четвериков Г.Г.* Основи теорії синтезу надшвидкодіючих структур мовних систем штучного інтелекту. К.: ІЗМН, 1997. 264 с. 2. *Бондаренко М.Ф., Коноплянко З.Д., Четвериков Г.Г.* Основи теорії багатозначних структур та кодування в системах штучного інтелекту. Харків: Фактор-друк, 2003. 331 с. 3. *Бондаренко М.Ф., Осыка А.Ф.* Автоматическая обработка информации на естественном языке. К.: УМК ВО, 1991. 142 с. 4. Пат. 2147789 РФ, МКВ Н03К 19/02, Н03М 1/00. Функциональный преобразователь с могозначным кодированием / *Бондаренко М.Ф., Коноплянко З.Д., Четвериков Г.Г.* Опубл. 24.04.2000. Бюл.11. 4 с. іл. 5. *Четвериков Г.Г.* Формалізація принципів побудови універсальних k-значних структур мовних систем штучного інтелекту // Доповіді НАН України. 2001, №1 (41). С.76–79.

Надійшла до редколегії 15.11.2004

УДК 621.397.331

О.Г. Руденко, П.О. Сотников

ОБ ОДНОМ АЛГОРИТМЕ СИНТЕЗА ПЕРСПЕКТИВНЫХ ТРАНСФОРМАЦИЙ ТЕЛЕВИЗИОННЫХ ИЗОБРАЖЕНИЙ НА ОСНОВЕ АФФИННЫХ ПРЕОБРАЗОВАНИЙ

Введение

В системах имитации визуальных обстановок тренажеров, в системах анализа, нормализации, классификации и распознавания изображений, в системах машинного зрения, в устройствах телевизионных эффектов и т. д. основную роль играют видеопроцессоры, позволяющие работать с изображениями в реальном времени. В современных телевизионных (ТВ) системах цифровые видеопроцессоры должны позволять в реальном времени трансформировать изображение с неограниченной точностью без его компрессии и декомпрессии. Кроме того, независимость времени выполнения преобразований от степени детализации изображений позволяет использовать видеопроцессоры при обработке изображений в стандарте высокой четкости.

В [1, 2] описаны разработанные ранее видеопроцессоры, реализующие аффинные преобразования ТВ-изображений. Известно, что при аффинном отображении одной плоскости на другую параллельные прямые переходят в параллельные. Как показывает практика, в результате изображения, полученные при помощи разработанных видеопроцессоров, выглядят недостаточно реалистично, что в современных условиях ограничивает область применения устройства.

Для получения более реалистичных изображений трехмерных объектов необходимо учитывать «эффект перспективы». Целью данной статьи является разработка алгоритма синтеза перспективных проекций ТВ-изображений на основе аффинных преобразований. В статье предлагается метод преобразования общих выражений для перспективных преобразований к виду, в котором их можно реализовать на специализированном вычислителе [1, 2].

Постановка задачи

Аффинные преобразования плоскости описываются следующими соотношениями:

$$\begin{cases} x_э = A_x x + B_x y + C_x; \\ y_э = A_y y + B_y y + C_y, \end{cases} \quad (1)$$

где A_x, B_x, C_x, A_y, B_y, C_y – аффинные коэффициенты.

Для того чтобы осуществлять аффинные преобразования изображений в реальном времени, программируемые средства должны успевать осуществлять необходимые вычислительные операции за период развертки изображения на устройстве визуализации. Это могут быть незначительные предварительные вычисления, результат которых, воздействуя на аппаратные средства, будет настраивать их на выполнение преобразований координат всего массива элементов разложения, согласно заданному преобразованию.

В разработанном в [1, 2] видеопроцессоре такой подход к синтезу аффинных проекций телевизионных изображений реализован таким образом, что аффинные коэффициенты необходимых преобразований видеопроцессорам задаются от персонального компьютера (ПК). ПК используется как средство удобного интерфейса пользователя для задания и выдачи на видеопроцессор желаемых коэффициентов преобразования. ПК может выдавать последовательность динамически изменяемых коэффициентов преобразования, имитируя эволюции реальной обстановки, а видеопроцессор производит эти эволюции с входным изображением.

Для достижения реального времени выполнения преобразований независимо от сложности изображений был использован принцип функционального соответствия между координатами экрана и адресами элементов изображения в буфере регенерации. При выполнении аффинных преобразований над исходным изображением, хранящимся в буфере, для его элементов определяются новые координаты, после чего эти элементы последовательно выводятся на видовую поверхность. При этом достигается существенный выигрыш времени выполнения преобразования по сравнению со временем, необходимым для выполнения такого же преобразования традиционным методом определения координат всех элементов массива по формулам (1). Высокое быстродействие достигается благодаря переходу от вычисления координат каждого элемента дискретизации изображения по выражениям (1) к вычислению по этим выражениям координат лишь начальных элементов строк и определенному вычислению координат остальных элементов строк.

Таким образом, для выполнения преобразований аффинной группы изображений, представленных своим массивом элементов дискретизации, по каналам связи достаточно передавать для каждого нового кадра изображения новые координаты лишь трех точек плоскости изображения.

Воспользуемся таким же подходом для реализации аппаратно-ориентированного алгоритма синтеза перспективных проекций ТВ-изображений, включающих помимо двух координат x и y еще и третью координату z.

Алгоритм перспективных трансформаций

Рассмотрим плоскость изображения в ортогональных координатах (x, y, z) (рис. 1). Плоскость $z = 0$ в данном случае является плоскостью экрана, точка S – глаз наблюдателя, точка (0, 0, 0) – центр экрана. Таким образом, предполагается, что взгляд наблюдателя направлен по вектору нормали к экрану, имеющему начало в его центре.

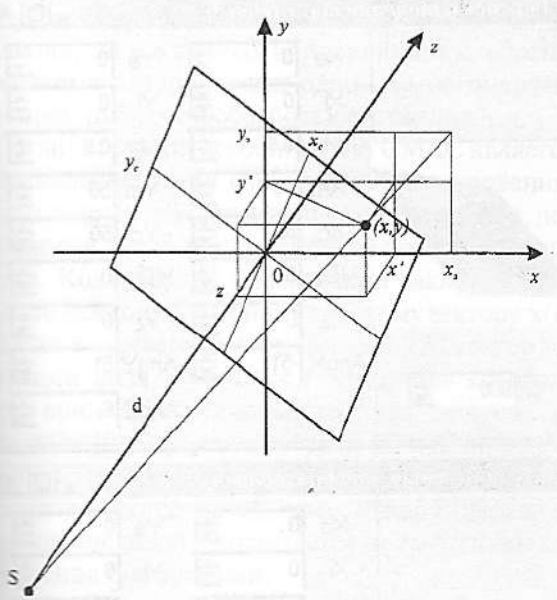


Рис. 1. Плоскость изображения

Введем следующие обозначения:

($x_э$, $y_э$) – координаты точки экрана;

(x, y, z) – координаты точки изображения;

x_e, y_e – смещения изображения относительно экрана соответственно по осям x и y экрана;

x_i, y_i – смещения изображения в плоскости изображения соответственно по осям x и y изображения;

(x_c, y_c) – координаты центра поворота изображения;

x, y, z – углы поворота изображения соответственно вокруг осей x, y, z экрана.

Спроецируем изображение на плоскость экрана, то есть отобразим точку (x, y, z) изображения в точку (x', y') экрана. Тогда отображение точки (x, y, z) может быть представлено следующими соотношениями:

$$\begin{cases} x' = (x-(x_c+x_i))\cos y\cos z + (y- \\ -(y_c+y_i))(\sin x\sin y\cos z + \cos x\sin z); \\ y' = -(x-(x_c+x_i))\cos y\sin z + (y- \\ -(y_c+y_i))(-\sin x\sin y\sin z + \cos x\cos z). \end{cases} \quad (2)$$

Пусть глаз наблюдателя удален от экрана на расстояние d. В этом случае с учетом «эффекта перспективы» точка (x, y, z) изображения будет отображаться в точку ($x_э$, $y_э$) экрана следующим образом:

$$\begin{cases} x_э = x'(1+\dfrac{z}{d-z}); \\ y_э = y'(1+\dfrac{z}{d-z}). \end{cases} \quad (3)$$

Предположим, что выполняется условие $z << d$, которое, как показало компьютерное моделирование, является вполне допустимым (слишком малые значения d давали чрезмерный «эффект перспективы», на практике не применимый, а при тех значениях d, которые давали наилучший результат, изображение было без видимых искажений).

Тогда выражения (3) принимают вид:

$$\begin{cases} x_э = x' + x'\dfrac{z}{d}; \\ y_э = y' + y'\dfrac{z}{d}. \end{cases} \quad (4)$$

Найдем координату z точки изображения

$$z = (x-(x_c+x_i))\sin y - (y-(y_c+y_i))\sin x\cos y. \quad (5)$$

Подстановка (2) и (5) в (4) приводит к формулам, которые ввиду их громоздкости здесь не приводятся, однако, как нетрудно убедиться, имеют вид

$$\begin{cases} x_э = a_1x^2 + a_2xy + a_3y^2 + a_4x + a_5y + a_6; \\ y_э = b_1x^2 + b_2xy + b_3y^2 + b_4x + b_5y + b_6. \end{cases} \quad (6)$$

Соотношения (6), в свою очередь, могут быть представлены следующим образом:

$$\begin{cases} x_э = (A_{xx}x + B_{xx}y + C_{xx})x + (A_{xy}x + B_{xy}y + C_{xy})y + C_x; \\ y_э = (A_{yx}x + B_{yx}y + C_{yx})x + (A_{yy}x + B_{yy}y + C_{yy})y + C_y. \end{cases} \quad (7)$$

Из (7) видно, что как сами эти преобразования, так и коэффициенты при x и y представляют собой аффинные преобразования.

Сравнивая формулы, получаемые в результате подстановки (2) и (5) в (4), с соотношениями (7), после несложных преобразований получаем выражения для коэффициентов A_{xx}, B_{xx}, C_{xx}, A_{xy}, B_{xy}, C_{xy}, C_x и A_{yx}, B_{yx}, C_{yx}, A_{yy}, B_{yy}, C_{yy}, C_y:

$$A_{xx} = d^{-1}\cos y\cos z\sin y;$$

$$B_{xx} = -d^{-1}\sin x\cos z\cos 2y;$$

$$C_{xx} = \cos y\cos z - d^{-1}(2(x_c+x_i)\cos y\cos z\sin y - \\ -(y_c+y_i)(\cos z\sin x\cos 2y - \cos x\sin y\sin z));$$

$$A_{xy} = d^{-1}\cos x\sin y\sin z;$$

$$B_{xy} = -d^{-1}\sin x\cos y(\sin x\sin y\cos z + \cos x\sin z);$$

$$C_{xy} = \sin x \sin y \cos z + \cos x \sin z +$$
$$+ d^{-1}((x_c + x_i)(\sin x \cos z \cos 2y - \cos x \sin y \sin z) +$$
$$+ 2(y_c + y_i)\sin x \cos y(\sin x \sin y \cos z + \cos x \sin z));$$
$$C_x = -(x_c + x_i)\cos y \cos z - (y_c + y_i)(\sin x \sin y \cos z -$$
$$- \cos x \sin z) + d^{-1}((x_c + x_i)^2 \cos y \sin y \cos z +$$
$$+ (x_c + x_i)(y_c + y_i)(-\sin x \cos z \cos 2y + \cos x \sin y \sin z) -$$
$$- (y_c + y_i)^2 \sin x \cos y(\sin x \sin y \cos z + \cos x \sin z)) +$$
$$+ x_c + x_e;$$

$$A_{yx} = -d^{-1} \cos y \sin y \sin z; \qquad (8)$$
$$B_{yx} = d^{-1} \sin x \sin z \cos 2y;$$

$$C_{yx} = -\cos y \sin z + d^{-1}(2(x_c + x_i)\sin y \cos y \sin z -$$
$$- (y_c + y_i)(\sin x \sin z \cos 2y + \cos x \sin y \cos z));$$

$$A_{yy} = d^{-1} \cos x \sin y \cos z;$$

$$B_{yy} = d^{-1} \sin x \cos y(\sin x \sin y \sin z - \cos x \cos z);$$

$$C_{yy} = \cos x \cos z - \sin x \sin y \sin z -$$
$$-d^{-1}((x_c + x_i)(\sin x \sin z + \cos x \sin y \cos z) +$$
$$+2(y_c + y_i)\sin x \cos y(\sin x \sin y \sin z - \cos x \cos z));$$

$$C_y = (x_c + x_i)\cos y \sin z + (y_c + y_i)(\sin x \sin y \sin z -$$
$$- \cos x \cos z) - d^{-1}((x_c + x_i)^2 \sin y \cos y \sin z +$$
$$+(x_c + x_i)(y_c + y_i)(\sin x \sin z \cos 2y + \cos x \sin y \cos z) +$$
$$+(y_c + y_i)^2 \sin x \cos y(\sin x \sin y \sin z - \cos x \cos z)) +$$
$$+ y_c + y_e.$$

Эффективность предлагаемого алгоритма проверялась путем компьютерного моделирования при помощи программы, разработанной в среде Delphi. Программа осуществляет преобразование изображений в соответствии с формулами (7), (8). Некоторые результаты трансформации произвольного изображения размером 120 × 100 точек при помощи разработанной программы приведены на рис. 2.

Как видно из фрагментов, представленных на рис. 2, расстояние d от глаза наблюдателя до экрана для достижения наилучшего результата было выбрано равным 600 точек. Для наглядности исходное изображение на каждом из фрагментов расположено под разными углами по отношению к осям координат (x, y, z) (соответственно ang$X$, ang$Y$, ang$Z$). При этом центр поворота изображения совпадает с его центром ($x_c = 60$; $y_c = 50$).

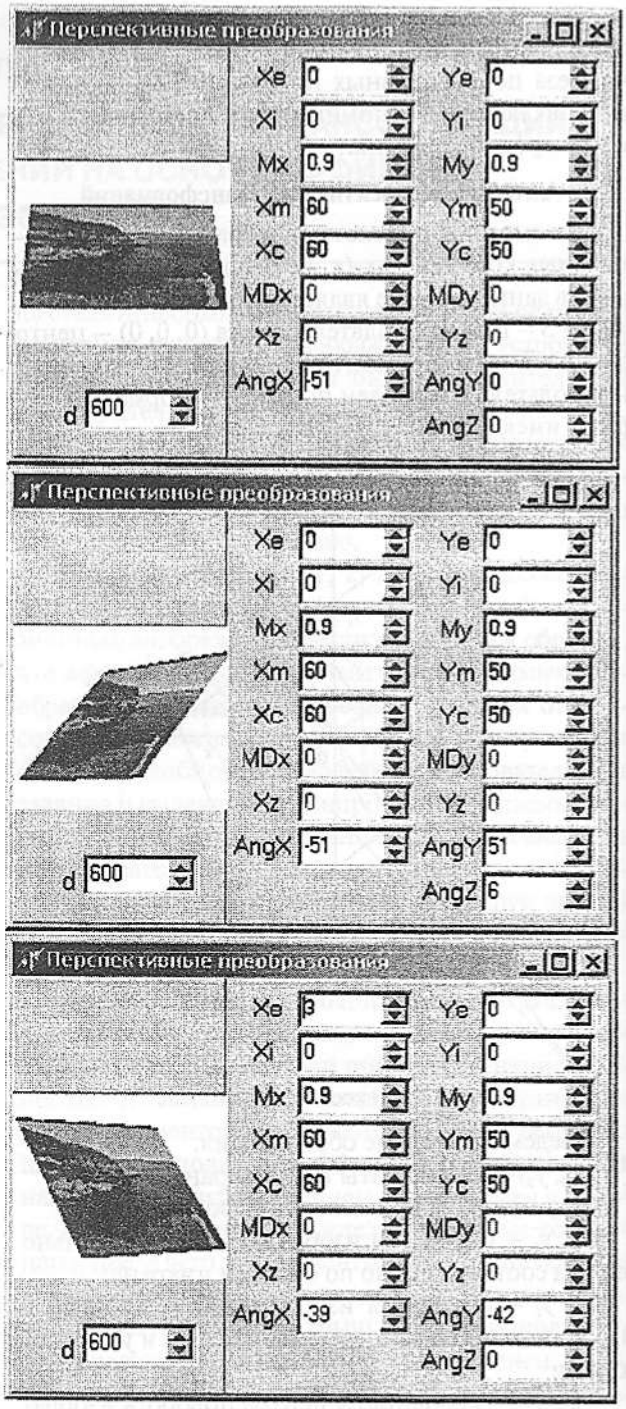


Рис. 2. Компьютерное моделирование перспективных преобразований

Выводы

Таким образом, предложенный в данной статье алгоритм позволяет синтезировать перспективные проекции телевизионных изображений на основе аффинных преобразований. Как и в случае аффинных преобразований, время выполнения преобразований (7), (8) не зависит от степени детализации изображений и поэтому данный алгоритм может быть использован для получения соответствующих перспективных трансформаций изображений в реальном времени. Кроме того, аппаратно данный алгоритм может быть реализован на основе аффинных процессоров [1, 2].

Список литературы: 1. *Rudenko O.G.*, Sotnikov O.M., Anjutin A.A. A Device for Fast Geometric Transformations of Color TV Images //Pattern Recognition and Image Analysis. 1991. Vol. 1, N 2. P. 265–266. 2. *Елаков С.Г.*, Елисеев А.А., Торба А.А. Аффинный видеопроцессор // Междунар. науч.-техн. конф. «Теория и техника передачи, приема и обработки информации»: тез. докл. Харьков. 1995. С. 87–88.

Поступила в редколлегию 28.10.2004

УДК 519.71

А.А. Бессонов, В.А. Бобух, О.Г. Руденко

ПРИМЕНЕНИЕ НЕЙРОННОЙ СЕТИ *CMAC* ПРИ РЕШЕНИИ ЗАДАЧ ОБРАБОТКИ ИЗОБРАЖЕНИЙ И ЕЕ АППАРАТНАЯ РЕАЛИЗАЦИЯ

Введение

Искусственная нейронная сеть *CMAC – Cerebellar Model Articulation Controller* (церебральная модель артикуляционного контроллера), описывающая процессы управления движением, происходящие в мозжечке и первоначально предложенная для управления роботом манипулятором [1, 2], оказалась достаточно эффективной не только в задачах управления, но и в задачах аппроксимации, восстановления и интерполяции одно- и многомерных функций, цифровой обработки сигналов и т.д.

Отличительной особенностью CMAC является использование в ней специального кодирования, позволяющего резко уменьшить объем памяти, необходимый для хранения обрабатываемой информации. Кодирование информации заключается в том, что каждому N-мерному входному вектору $\mathbf{x}(i)$ ставится в соответствие n-мерный ($n<N$) вектор ассоциаций $a(i)$, $\rho = const << n$ элементов которого могут принимать отличные от нуля значения из интервала [0, 1] (в классической CMAC они принимаются равными 1), то есть только ρ элементов памяти являются активными. Дополнительное уменьшение памяти достигается применением хеширования информации.

В статье рассматривается применение сети CMAC для фильтрации и кодирования изображений и предлагается один из вариантов ее аппаратной реализации.

Нейронная сеть CMAC

Сеть CMAC, пример которой приведен на рис. 1, в общем случае осуществляет следующие преобразования:

$$S: X \Rightarrow A, \quad (1)$$

$$H: A \Rightarrow A', \quad (2)$$

$$P: A' \Rightarrow y, \quad (3)$$

где X – N-мерное пространство непрерывных входных сигналов; A – n-мерное пространство ассоциаций; A' – преобразованное алгоритмом хеширования пространство ассоциаций; y – вектор выходных сигналов.

Преобразование (1) соответствует кодированию информации

$$a = S(\mathbf{x}), \quad (4)$$

(2) – хешированию

$$a' = H(a), \quad (5)$$

а (3) – вычислению выходного сигнала

$$\hat{\mathbf{y}} = P(a') = (a')^{\mathrm{T}} w = (H(a))^{\mathrm{T}} w. \quad (6)$$

Важным вопросом при реализации сети является выбор базисных функций.

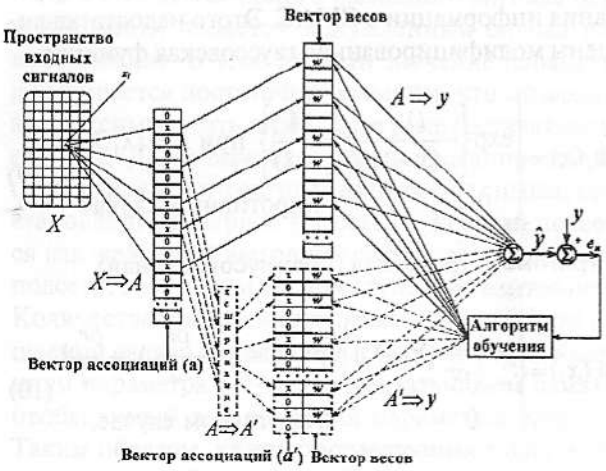


Рис. 1. Структура сети *CMAC*

Выбор базисных функций

Выражение (6) описывает преобразование, осуществляемое в традиционной CMAC с использованием хеширования информации при выборе прямоугольных базисных функций. Если же в сети используются нейроны с активационными функциями, отличными от прямоугольной, преобразование (6) принимает вид

$$\hat{\mathbf{y}} = H(a^{\mathrm{T}} \Phi(x))\, w, \quad (7)$$

где $\Phi(x) = \begin{bmatrix} \Phi_1(x) & 0 & \dots & 0 \\ 0 & \Phi_2(x) & \dots & 0 \\ \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & \Phi_n(x) \end{bmatrix}$,

где $\Phi_i(x) = \prod_{j=1}^{N} \phi_{ij}(x_j)$; $\phi_{ij}(x_j)$ – значение выбранной базисной функции в точке x_j.

В традиционной CMAC используются прямоугольные базисные функции, что позволяет осуществлять кусочно-постоянную аппроксимацию. В этом случае вычислительные затраты будут минимальными, вследствие чего значительно сокращается время реакции сети на поступивший вход-

ной сигнал. Компоненты вектора ассоциаций при этом могут принимать значения 0 или 1. Скорость обучения сети при выборе базисных функций прямоугольной формы будет максимальной [3].

В качестве базисных в сети CMAC часто используются гауссовские функции

$$\Phi_i(x_j) = \exp\left\{-\frac{(x_j-\mu_i)^2}{\sigma_i^2}\right\}. \quad (8)$$

Однако, хотя гауссовская функция и обладает свойством локального возбуждения, достаточно сложно четко выделить ее границы возбуждения, что является важным для осуществления кодирования информации в CMAC. Этого недостатка лишены модифицированная гауссовская функция

$$\Phi_i(x) = \begin{cases} \exp\left\{-\dfrac{(\lambda_2-\lambda_1)^2/4}{(x-\lambda_1)(\lambda_2-x)}\right\} & \text{при } x \in (\lambda_1,\lambda_2), \\ 0 & \text{в противном случае} \end{cases} \quad (9)$$

и тригонометрическая (косинусоидальная)

$$\Phi_i(x_j) = \begin{cases} \cos\left(\dfrac{\pi}{\rho r_j}(x_j-\lambda_i)\right) & \text{при } x_j \in (\lambda_i-\dfrac{\rho r_j}{2},\lambda_i+\dfrac{\rho r_j}{2}], \\ 0 & \text{в противном случае,} \end{cases} \quad (10)$$

где λ_i – i-й центр области квантования; r_j – шаг квантования по j-ой компоненте входного сигнала.

Алгоритмы хеширования информации в CMAC

Хеширование заключается в отображении большой требуемой памяти в меньшую, физически реализуемую. Объем большей памяти определяется размерностью вектора ассоциаций, полученного на этапе кодирования. Если адреса ячеек большей памяти лежат в интервале [1, M], а меньшей – в интервале [1, m], то функция преобразования адреса (хеш-функция) должна удовлетворять требованию

$$1 \le H(k) < m, \ \forall\, 1 \le k \le M.$$

Выбор функции $H(k)$ является весьма трудной задачей, однако многочисленные исследования показали, что хорошие результаты обеспечивает применение двух основных типов хеш-функций, один из которых основан на делении, а другой – на умножении [4].

Хеш-функция, реализующая метод, основанный на делении, имеет вид

$$H(k) = 1 + k \bmod m, \quad (11)$$

где $k \bmod m$ означает деление по модулю m.

Адрес ячейки памяти с меньшим объемом при методе хеширования, основанном на умножении, вычисляется по следующей формуле:

$$H(k) = 1 + \left\lfloor m\left[\left(\frac{F}{w}k\right)\bmod 1\right]\right\rfloor, \quad (12)$$

где w – размер машинного слова; F – некоторая целая константа, взаимно простая с w; $\lfloor\bullet\rfloor$ – означает округление в сторону ближайшего целого числа.

В качестве F возможно использование чисел Фибоначчи.

Хеш-функция, основанная на делении, равномерно распределяет большую память в меньшую в соответствии с адресом.

Функция, описываемая уравнением (12), «тасует» элементы памяти.

Хеш-коллизии

Какой бы метод хеширования ни применялся, невозможно полностью устранить проблему хеш-коллизий. При хешировании информации в сети CMAC возможно возникновение двух типов коллизий. Коллизии первого типа возникают при активации одним входным сигналом одной ячейки памяти более одного раза. Вероятность того, что один входной сигнал обратится два и более раз к одной ячейке физической памяти, можно определить по формуле

$$P = \sum_{i=2}^{\rho-1}\frac{i}{m}, \quad (1$$

где m – объем меньшей памяти.

Так, например, для случая $\rho = 20$, $m = 2000$ в вероятность многократного обращения к одной ячейке будет равна примерно 0,1, то есть является достаточно малой. Возникновение такой коллизии означает то, что значение, хранящееся в многократно активизируемой ячейке, несколько раз участвует в формировании выходного сигнала.

Причиной возникновения коллизий второго типа является активации одной и той же ячейки памяти различными далеко расположенными друг от друга (в смысле расстояния Хэмминга) входными сигналами. Эти коллизии приводят к более серьезной проблеме – появлению нежелательной корреляции двумя далеко расположенными (в смысле расстояния Хэмминга) входными сигналами. Эти коллизии не могут существенно повлиять на работу сети, пока выполняется условие

$$a_1' \wedge a_2' << \rho,$$

где a_1' и a_2' — преобразованные хеш-функцией векторы ассоциаций двух различных входных сигналов.

Для любых двух входных сигналов, для которых не выполняется $a_1' \wedge a_2' = \phi$ (здесь ϕ – пустое множество), вероятности обращения к общим ячейкам можно определить следующим образо

$$P_n(k) = C_n^k p^k q^{n-k}, k = \overline{1,n}, \quad (14)$$

где C_n^k — число сочетаний; $p=1-q$.

Для нашего примера $q = \frac{20}{2000}$. В этом случае вероятность того, что $a_1' \wedge a_2' = 0$, то есть $P_{20}(20)$, равна 0,818. Вероятности же того, что $a_1' \wedge a_2' = 1$, $a_1' \wedge a_2' = 2$, $a_1' \wedge a_2' \geq 3$ будут равны соответственно 0,165; 0,016; 0,001.

Обучение

Обучение сети CMAC состоит в настройке вектора ее весовых параметров **w** на основании предъявления ей обучающих пар $\{\mathbf{x}(k), \mathbf{y}(k)\}$ и сравнения реакции сети $\hat{\mathbf{y}}(k)$ с требуемой $\mathbf{y}(k)$. Соответствующий алгоритм обучения может быть записан следующим образом:

$$\mathrm{w}(k+1) = \mathbf{w}(k) + \frac{\mathbf{y}(k) - \hat{\mathbf{y}}(k)}{\left\| \Phi(x(k))a(k) \right\|^2} \Phi(\mathbf{x}(k))a(k). \quad (15)$$

Однако в ряде случаев для улучшения аппроксимирующих свойств сети возникает необходимость настройки не только весовых параметров сети, но и параметров базисных функций. Наиболее эффективным оказывается применение такого подхода для настройки центров μ и дисперсий σ гауссовских функций активации (8).

Настройка данных параметров может быть осуществлена, например по алгоритму, аналогичному алгоритму Уидроу-Хоффа:

$$m(k+1) = m(k) + \frac{\mathbf{y}(k) - \hat{\mathbf{y}}(k)}{\left\| \frac{\partial \hat{\mathbf{y}}(k)}{\partial m(k)} \right\|^2} \frac{\partial \hat{\mathbf{y}}(k)}{\partial m(k)},$$

где m — вектор настраиваемых параметров.

В случае использования гауссовских функций в качестве m используются центры μ и дисперсии σ, а векторы частных производных $\frac{\partial \hat{\mathbf{y}}(k)}{\partial \mu(k)}$ и $\frac{\partial \hat{\mathbf{y}}(k)}{\partial \sigma(k)}$ вычисляются по формулам

$$\frac{\partial \hat{\mathbf{y}}(k)}{\partial \mu_{ji}(k)} = 2w_i(k) \prod_{s=1}^{N} e^{\frac{(x_s - \mu_{si}(k))^2}{\sigma_{si}(k)^2}} \frac{(x_j - \mu_{ji}(k))}{\sigma_{ji}(k)^2},$$

$$\frac{\partial \hat{\mathbf{y}}(k)}{\partial \sigma_{ji}(k)} = 2w_i(k) \prod_{s=1}^{N} e^{\frac{(x_s - \mu_{si}(k))^2}{\sigma_{si}(k)^2}} \frac{(x_j - \mu_{ji}(k))^2}{\sigma_{ji}(k)^3},$$

где i — номер возбужденного нейрона; j — номер компонент векторов ассоциаций и весов, соответствующих данному возбужденному нейрону.

Задача фильтрации изображений

Для фильтрации изображений предлагается использовать иерархическую сеть CMAC, состоящую из нескольких подсетей, расположенных последовательно и отличающихся количеством ступеней квантования (число ступеней квантования определяется параметром ρ), и представленную на рис. 2 [5, 6].

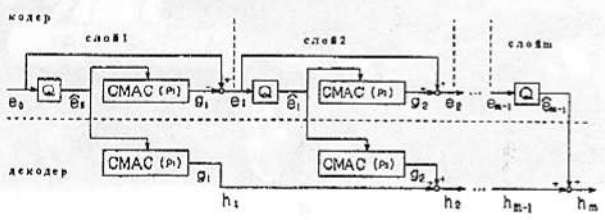


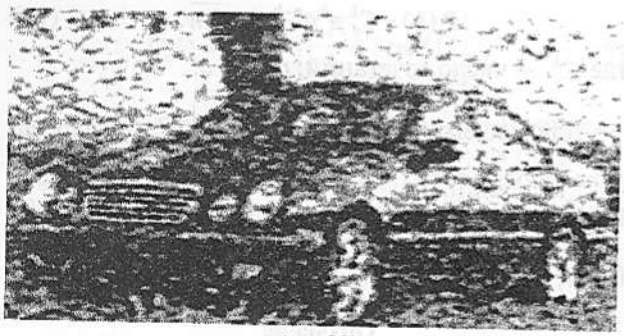
Рис. 2. Схема фильтрации изображений

В качестве желаемого выходного сигнала первой подсети подается зашумленный сигнал или изображение. В этой подсети значение параметра ρ выбирается достаточно большим, что позволяет аппроксимировать низкочастотные составляющие сигнала. Далее вычисляется ошибка аппроксимации (из исходного сигнала вычитается сигнал, восстановленный первой подсетью), которая подается как желаемый выходной сигнал для следующей подсети, с меньшим числом ступеней квантования. Количество подсетей определяется характером решаемой задачи. Следует отметить, что с уменьшением параметра ρ будет возрастать объем памяти, необходимый для хранения параметров подсети. Таким образом, каждая последующая подсеть будет содержать большее число настраиваемых параметров.

Пусть оригинальное изображение имеет размеры $N \times N$ и обозначается $e_0(x,y)$. Для CMAC 1-го слоя выбирается параметр $\rho_1 \approx 0{,}5N$ и обучается данная сеть на дискретизированном оригинальном изображении, причем сети в качестве обучающих пар предъявляются не все пиксели изображения, а лишь те, которые лежат на пересечении строк и столбцов, номера которых кратны ρ_1 (этот принцип обучения используется для всех подсетей). Информация о пропущенных пикселях частично восстанавливается текущей подсетью, а затем будет восстановлена последующими подсетями. Изображение, аппроксимированное первой подсетью, обозначим $g_1(x, y)$. Тогда ошибка аппроксимации после обучения первой подсети равна $e_1(x, y) = e_0(x, y) - g_1(x, y)$. Ошибка $e_1(x, y)$ является входным сигналом второй подсети. Для этой подсети рекомендуется выбирать параметр ρ_2 равный $0{,}5\rho_1$, Уточнение иерархической структуры прекращается при достижении желаемой точности восстановления изображения.

На рис. 3 представлены результаты фильтрации изображения размером 400 × 200 пикселей. На рис. 3*а* показано оригинальное изображение, а на рис. 3*б* — зашумленное. На рис. 3*в*, 3*е* приведены результаты восстановления изображения иерархической сетью CMAC, состоящей из четырех слоев.

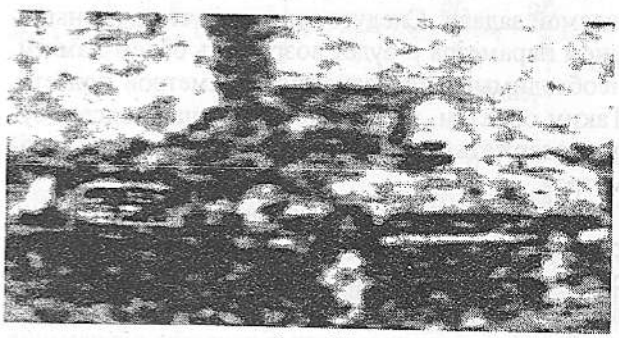
а



г (ρ = 20)

б

д (ρ = 10)



в (ρ = 40)

е (ρ = 5)

Рис. 3. Пример фильтрации изображения с помощью сети CMAC

В первом слое параметр *c* выбирался равным 40, во втором – 20, в третьем – 10, а в четвертом – 5. Во всех подсетях использовались косинусоидальные базисные функции. Общий объем требуемой памяти составил $n_{общ}$= 32644 ячеек.

Кодирование изображений

В работе [5] показано, что нейронная сеть CMAC может быть эффективно применена для решения задач кодирования и сжатия изображений. В данной работе исследовалось влияние выбора параметров сети CMAC (*R* и ρ), а также настройки параметров базисных функций на качество кодирования. На рис. 4*а* представлено оригинальное изображение. На рис. 4*б* – изображение, восстановленное с помощью нейронной сети CMAC с параметрами *R*=256, ρ = 20, в которой не осуществлялась настройка параметров базисных функций (то есть они были заданы априорно). Коэффициент сжатия при этом равен *k* = 17,33. На рис. 4*в* приведены результаты кодирования изображения с использованием настройки параметров. При этом количество хранимых параметров несколько увеличилось, и коэффициент сжатия составил *k* = 13,37.

Аппаратная реализация

Рассмотрим кодирование переменной x_i, представляющей собой *n*-разрядное двоичное число. Так как входной слой сети CMAC содержит ρ ступеней квантования, таких, что для любой ступени существует другая ступень, на которой области квантования сдвинуты относительно рассматриваемой ступени на один уровень квантования входной переменной x_i, являющейся двоичным числом, то такой сдвиг соответствует единице младшего разряда входной переменной. Присвоим ступени квантования индекс $j = \overline{0,\rho-1}$.

В связи с тем, что область квантования содержит ρ уровней квантования входной переменной,

а

б

в

Рис. 4. Пример кодирования изображения с помощью сети CMAC

активационная функция нейрона, соответствующего любой области квантования, определена в с точках. Аргументом этой функции будет являться значение

$$u = (x_i + j) \mod \rho, \qquad (16)$$

где mod – функция вычисления остатка от целочисленного деления.

Номер активного нейрона на ступени j можно определить как

$$v = (x_i + j) \operatorname{div} \rho, \qquad (17)$$

где div – функция целочисленного деления.

Как известно, аппаратная реализация операции деления на произвольное число является либо ресурсоёмкой либо медленной. Поэтому предлагается выбирать значения ρ, равные 2^m. В этом случае операция деления может быть заменена операцией сдвига на m разрядов вправо, а остаток от деления, записанный в m младших разрядов, исчезает.

В классическом варианте сети CMAC активационные функции нейронов областей квантования имеют прямоугольную форму. Однако, как показывают исследования, лучший результат достигается при использовании функций высших порядков (линейных, квадратичных, кубических), а также тригонометрических и степенных функций.

Аппаратная реализация широкого спектра активационных функций, отличных от прямоугольных, сложна и ресурсоёмка. Поэтому целесообразно функцию активации задавать таблично в ОЗУ, содержащем 2^m ячеек. Разрядность ячеек ОЗУ h определяется областью допустимых значений функции активации, а также необходимой точностью задания значений функции активации. Если, например, $m = 8$, то необходимо ОЗУ ёмкостью 256 ячеек. Блоки ОЗУ такого объёма имеются в большинстве современных ПЛИС (например семейство ACEX1K фирмы ALTERA). Подобная реализация функций активации является простой и гибкой, так как позволяет задать функцию любого вида.

Структура блока кодирования переменной x_i представлена на рис. 5. Код переменной x_i защелкивается в регистре RgX, откуда поступает на сумматор Σ1. На второй вход этого сумматора поступает m – разрядное значение номера ступени j, на которой в данный момент происходит кодирование. Это значение формируется устройством случайной выборки ступени, служащим для устранения диагональной структуры ассоциативного поля, и, соответственно, улучшения аппроксимирующих свойств сети. С выхода сумматора n-разрядный код поступает на логический сдвигатель (LSh – Logical Shifter), позволяющий изменять значение m. Значение u определяют m младших разрядов выходного кода LSh, а значение v – соответственно старшие $k = n - m$ разрядов. Значение u через мульти-

плексор MX1 поступает на адресные входы ОЗУ хранения активационной функции. Мультиплексор MX1, а также буфер B1 служат для инициализации ОЗУ значениями активационной функции от устройства управления. С выхода ОЗУ h – разрядное значение функции активации поступает на регистр Rg VFS, а с регистра – на выход VFS_i (Value of Function on Stair i) блока кодирования (CB – Coding Block). Значение v поступает на регистр Rg NNS, а оттуда – на выход NNS_i)Number of Neuron on Stair i) блока. Тактирующие импульсы на регистры, а также сигналы управления ОЗУ формируются устройством управления, общим для всей структуры.

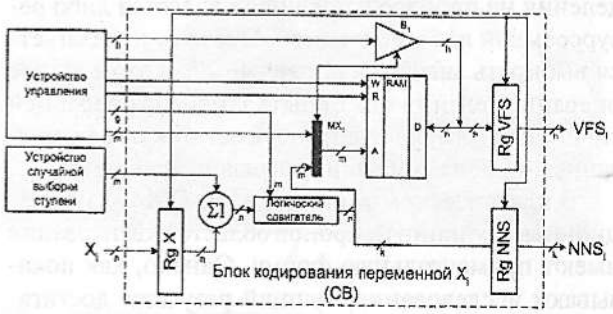


Рис. 5. Блок кодирования переменной x_i

Таким образом, для кодирования переменной x_i с помощью m ступеней квантования необходимо 2^m тактов. При $m < 5$ такое кодирование может осуществляться с частотой 10–20 МГц, то есть в реальном времени для обработки телевизионных изображений. В течение этих 2^m тактов значение x_i хранится во входном регистре RgX и обновляется по их истечении.

Устройство случайной выборки ступени может представлять собой m-разрядный линейный рекуррентный регистр с обратными связями, определяемыми простым полиномом. Такой регистр генерирует 2^m-1 различных кодовых комбинаций, обладающих статистическими свойствами равномерно распределённой случайной последовательности. Одна комбинация является запрещённой, так как на очередном такте работы регистра приводит к генерации самой себя. Так как работа блока кодирования предусматривает использование 2^m комбинаций, необходимо предусмотреть её аппаратную генерацию и подстановку вместо кода регистра на любом такте (при этом не производится очередной сдвиг регистра). Вид последовательности определяется начальной установкой регистра. Эта операция производится однократно перед первым обучением сети. Регистры, отвечающие за различные входные переменные, должны быть проинициализированы различными значениями. Эти значения, равно как и начальные значения весов в сети, могут быть сформированы аппаратным датчиком случайных чисел [7]. Рассмотрим возможную структуру аппаратной реализации сети CMAC, представленную на рис. 6. Сеть обрабатывает две входные переменные, каждая из которых кодируется блоками CB, содержащими равное число ступеней квантования. Функционирование такой структуры состоит в следующем. На j-м такте блоки CB формируют значения номеров нейронов в текущих ступенях (NNS_1 и NNS_2) и значения их функций активации (VFS_1 и VFS_2). Значения VFS поступают на умножитель M1, формирующий произведение функций активации текущих ступеней. Значение этого произведения поступает на умножитель M2, вычисляющий произведение веса и произведения функций активации. Адрес веса в ОЗУ весов определяется алгоритмом хеширования, входным параметром которого является комбинация значений NNS. Накапливающий сумматор Σ2 вычисляет сумму произведений весов и функций активации и после выполнения m тактов в регистре RgY формирует значение функции, соответствующее входным переменным. С помощью сумматора Σ3 формируется код ошибки e_k, определяемый как разность полученного y и требуемого y_k значений функции. Значение ошибки поступает на блок реализации алгоритма обучения, который в течение m тактов вычисления накапливал адреса выборки весов, значения весов и произведений функций активации. Получив значение ошибки, алгоритм обучения модифицирует участвовавшие в вычислении веса и записывает их в ОЗУ весов по соответствующим адресам. Координацию работы узлов сети осуществляет устройство управления.

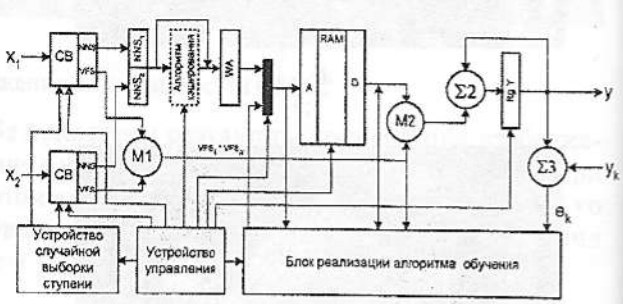


Рис. 6. Вариант аппаратной реализации сети CMAC

Основные результаты и выводы

Как показывают результаты моделирования, нейронная сеть CMAC является достаточно эффективной и удобной при решении задач фильтрации и кодирования изображений. Существенная экономия требуемой памяти в сочетании с простыми алгоритмами обучения делают ее особенно привлекательной для реализации в микроконтроллерных системах. Следует отметить, что приведенная в статье аппаратная реализация сети CMAC является универсальной и может использоваться не только

в задачах обработки изображений, но и для аппроксимации функций, идентификации нелинейных объектов и управления ими. Это связано с тем, что, во-первых, при этом возможно использование любых базисных функций, а во-вторых – в схеме предусмотрен блок хеширования информации. Дело в том, что вопрос выбора базисных функций нейронной сети CMAC остается открытым и определяется характером решаемой прикладной задачи. Хеширование же необходимо при обработке больших объемов информации (изображения высокого качества, объекты большой размерности).

Необходимо также отметить, что применение сети CMAC для фильтрации изображений является менее эффективным, однако в ряде случаев возможным. Сложность реализации иерархической CMAC не позволяет использовать ее, если фильтрация изображений должна происходить в реальном времени.

Список литературы: 1. *Albus J.S.* A new approach to manipulator control: the cerebellar model articulation controller (CMAC) // ASME Trans., J. Dynamic Systems, Measurement and Control, 1975. Vol. 97. №3. P. 220-227. 2. *Albus J.S.* Data storage in cerebellar model articulation controller (CMAC) // ASME Trans. J. Dynamic Systems, Measurement and Control, 1975. Vol. 97. №3. P. 228–233. 3. *Руденко О.Г.,* Бессонов А.А. О выборе базисных функций в нейронной сети CMAC // Проблемы управления и информатики. 2004. №2. С. 143-154. 4. *Кнут Д.Э.* Искусство программирования, том 3. Сортировка и поиск, 2-е изд., М.: Вильямс, 2000. С. 549-597. 5. *Руденко О.Г.,* Бессонов А.А., Бобух В.А. Фильтрация изображений с помощью нейронной сети *CMAC* // Вестник Херсонского гос. техн. уни-та. 2004. №1(19) . С. 378-384. 6. *Iiguni Youji.* Hierarchical image coding via Cerebellar Model Arithmetic Computers (CMAC) // IEEE Trans. on Image Processing, 1996. Vol.5. №10. P. 1393-1401. 7. *Торба А.А.,* Елаков С.Г., Степченко А.З. Генерация равновероятных случайных последовательностей на основе физических датчиков // «Радиотехника»: Всеукр. межвед. науч.-техн. сб. Харьков. ХТУРЭ. 2001. Вып. 119. С. 108-113.

Поступила в редколлегию 22.11.2004

УДК 004.8

Е.П. Киричук, И.Б. Сироджа

ИНТЕЛЛЕКТУАЛЬНАЯ СИСТЕМА ПОДДЕРЖКИ ПРИНЯТИЯ ТЕХНОЛОГИЧЕСКИХ РЕШЕНИЙ ПРИ ВЫБОРЕ ОБОРУДОВАНИЯ В ТЕХНОЛОГИЧЕСКОЙ ПОДГОТОВКЕ АВИАЦИОННОГО ПРОИЗВОДСТВА

Введение

На современном этапе технологическая подготовка производства (ТПП) характеризуется как процесс крупномасштабный, плохо формализованный, сложный с точки зрения автоматизации.

Без качественной и своевременной технологической подготовки производства в современных экономических условиях не может успешно работать ни одно предприятие. Сроки разработки документации, несколько лет назад считавшиеся приемлемыми, сегодня не устраивают ни руководителей, ни самих технологов. Не последнюю роль в этом играет возросшая конкуренция между предприятиями. Задержка в исполнении заказов приводит к экономическим потерям и существенно влияет на положение предприятия на рынке.

При ТПП одной из основных задач, которая характеризуется значительной сложностью из-за большого парка оборудования, многовариантностью выбора, высокой степенью неопределенности, что усложняет процесс принятия решений технологом, является выбор оборудования.

При выборе оборудования технолог должен решать плохо формализованные задачи, к которым относятся, например, задачи выбора типа оборудования для различных технологических операций. В таких случаях лишь интуиция, опыт и умения технолога, как эксперта в своей области, позволяют найти приемлемое решение.

Эти факты привели к потребности сохранить умение опытных технологов в базе знаний (БЗ) для использования менее опытными технологами [1].

Таким образом, разработка интеллектуальных систем поддержки принятия решений (ИСППР) [2] при выборе оборудования в ТПП является актуальной задачей.

1. Постановка задачи

Цеховой технолог должен создать технологическую карту (ТК) на изготовление детали, в которой описывается технологический процесс (ТП) с указанием необходимого оборудования, по следующим исходным данным, полученным от главного технолога: чертёж детали, серийность производства, марка материала, толщина материала, срок выполнения заказа, точность изготовления детали.

Для решения поставленной задачи технолог решает как формализованные, так и неформализованные задачи принятия решений. Поэтому для обеспечения интеллектуальной поддержки принятия решений технологом по выбору оборудования необходимо создать ИСППР, позволяющую:

— создавать и использовать БЗ для накопления и использования знаний по выбору оборудования;

— выполнять технологические расчеты;

— автоматически генерировать технологическую документацию;

— предоставлять данные о текущем состоянии парка оборудования.

2. Решение задачи

Авторами создан исследовательский прототип «КВАНТ_ВО», как ядро интеллектуальной информационной технологии в инженерии квантов знаний, с целью обеспечения возможности аккумулировать и накапливать опыт специалистов технологов в идентификации производственных ситуаций по выбору оборудования.

В качестве теоретической базы для представления и манипулирования знаниями используется vPAKЗ метод, созданный проф. Сироджей И.Б. [3–5]. Методика выбора оборудования для реализации ИСППР описана в работе [6]. ИСППР представляет собой исследовательский прототип интерактивного программного комплекса (ИПК) «КВАНТ_ВО», являющегося подсистемой ИПК «КВАНТ+» [1, 2], содержащего следующие подсистемы: подсистема управления проектом, подсистема выбора оснастки для листовой штамповки, подсистема мониторинга текущего состояния производства, подсистема механообработки, подсистема наплавки, подсистема горячей объемной штамповки.

ИСППР помогает технологу принимать решения в условиях многокритериальности выбора и различных неопределенностей. Последние обусловлены неполнотой и нечеткостью данных, преобладанием качественных характеристик производственных ситуаций, заранее неизвестным правилом принятия решений, отсутствием четкой определенности критериев или признаков, которые влияют на принимаемое решение. Подобные условия приводят к проблемам, в решении которых особую важность имеет человеческая интуиция.

Рассматриваемая задача принятия решений при выборе оборудования в ТПП является типичным

примером плохо формализованной проблемы принятия технологических решений, где неизвестны надежные количественные модели и закономерности, описывающие связь конкретной производственной ситуации с соответствующим принимаемым решением.

Разработка ИСППР «КВАНТ_ВО» осуществляется путем реализации синтезированных алгоритмов [5] средствами объектно-ориентированного программирования.

ИПК «КВАНТ_ВО» позволяет создавать и использовать БЗ для поддержки принятия решений технологом листоштамповочного цеха при проектировании ТП и при выборе типа оборудования для выполнения операций вырубки, пробивки, зачистки, чистовой вырубки пробивки, формоизменяющих листоштамповочных операций, резки листа на полосы, резки полосы на штучные заготовки.

«КВАНТ_ВО» работает в следующих режимах:

1. Режим работы технолога-эксперта:

— обучение на примерах (ввод выборки сценарных примеров обучающих знаний для создания БЗ);

— дообучение и переобучение (необходимо использовать тогда, когда система затрудняется принять решение или дает неверные решения);

2. Режим работы технолога-пользователя:

— выбор оборудования (принятие решений с помощью БЗ для проектирования технологического процесса (ТП) и выбора типа оборудования).

Архитектура подсистемы «КВАНТ_ВО» представлена на рис. 1 и включает следующие модули:

1. Проектирование технологического процесса (модуль проектирует ТП, используя БЗ).

2. Расчёт работы (модуль рассчитывает работу для выполнения штамповочной операции).

3. Расчет усилия (модуль рассчитывает усилие для выполнения штамповочной операции).

4. Выбор типа оборудования (модуль выбирает тип оборудования, используя БЗ).

5. Выбор пресса (модуль выбирает пресс с учетом работы, усилия, исправности, занятости станка).

6. Заполнение ТК (модуль производит автоматическое заполнение ТК в Excel).

7. Построение логической сети вероятных рассуждений (ЛСВР) и преобразование ее в v-квантовую сеть вывода решений (v-КСВР) (модуль осуществляет обучение ЛСВР на знаниях в виде сценарных примеров обучающих знаний (СПОЗ), затем преобразует ЛСВР в v-КСВР с помощью алгоритма vАЛОБ_КВАНТ).

8. Модуль оптимизации v-КСВР (модуль осуществляет минимизацию сети по свертке и по следствию, а также минимизацию квантов по законам идемпотентности с помощью алгоритма vАЛОПТС_КВАНТ).

9. Модуль поддержки принятия решений рассчитывает показатели достоверности (ПД) целевых заключений посредством алгоритма АЛПД(B_V) под управлением vАЛПР, объясняет принятое решение, обеспечивая ответ на вопрос «почему?».

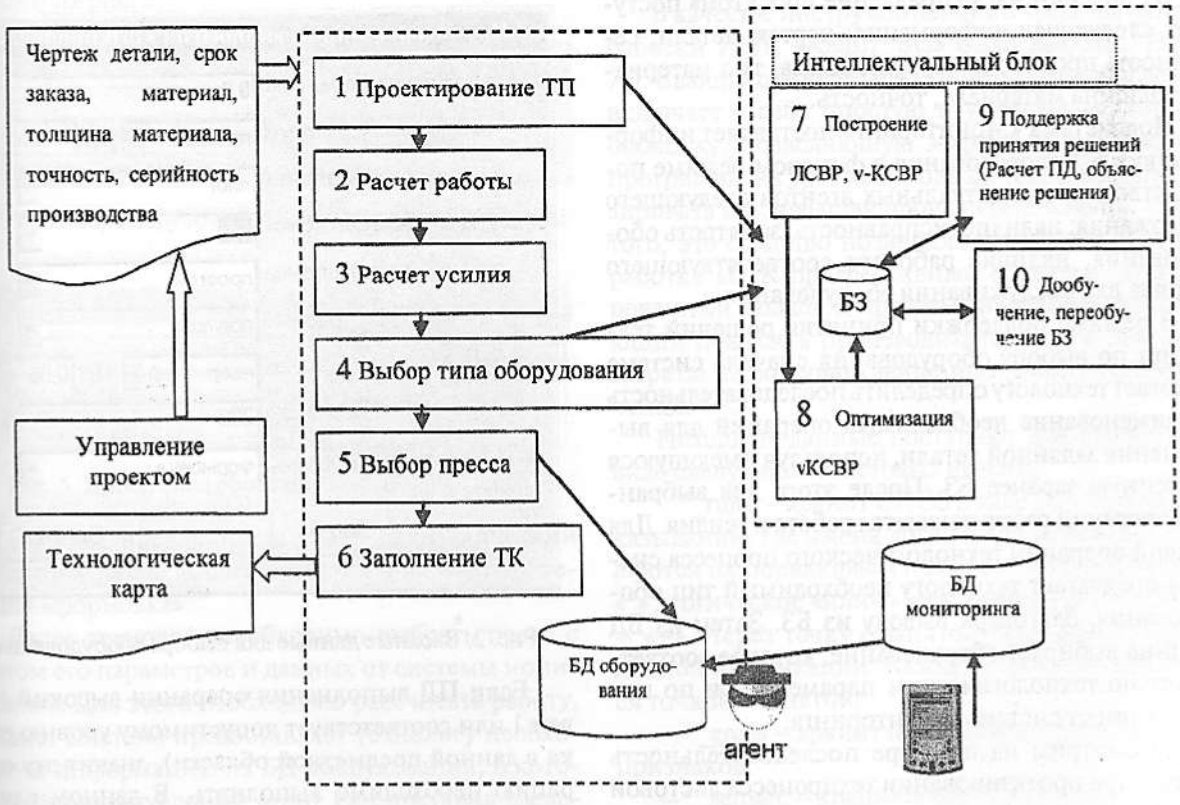


Рис. 1. Архитектура «КВАНТ_ВО»

10. Модуль дообучения, переобучения (модуль производит дообучение или переобучение БЗ, используя дополнительные обучающие примеры под управлением алгоритма vАЛПР).

11. В БД оборудования хранится информация о технологических параметрах оборудования, а также информация от системы мониторинга о его наличии, исправности, занятости, наличии соответствующего рабочего.

12. БЗ формируется на основе v-КСВР и одновременно является механизмом вывода решений.

В режиме обучения необходимо задать СПОЗ и перечень целевых заключений. Машина автоматически создаст ЛСВР. В случае необходимости в диалоговом режиме машина будет сообщать о недостающей информации для завершения обучения или об ошибках в СПОЗ (например наличие циклов в рассуждениях). После этого автоматически осуществляется преобразование ЛСВР в v-КСВР, оптимизация сети.

По запросу технолога-пользователя машина будет осуществлять поддержку принятия решений посредством расчета показателя достоверности (ПД) целевых заключений и объяснения принятых решений. Целевое заключение с максимальным ПД является решением. Если все ПД ниже 0,5, необходимо выполнить дообучение БЗ, добавив новые СПОЗ. Если решения, принятые машиной, не удовлетворяют технолога, необходимо осуществить переобучение БЗ, создав новую порцию СПОЗ или изменив старую.

От подсистемы «Управление проектом» поступает следующая информация: чертеж детали, серийность производства, срок заказа, тип материала, толщина материала, точность.

Подсистема «Мониторинг» поставляет информацию в БД оборудования в фоновом режиме посредством интеллектуальных агентов следующего содержания: наличие, исправность, занятость оборудования, наличие рабочего соответствующего разряда для обслуживания оборудования.

В режиме поддержки принятия решений технолога по выбору оборудования сначала система помогает технологу определить последовательность и наименование необходимых операций для выполнения заданной детали, используя имеющуюся обученную заранее БЗ. После этого для выбранных операций рассчитываются работа и усилия. Для каждой операции технологического процесса система предлагает технологу необходимый тип оборудования, благодаря выводу из БЗ. Затем из БД машина выбирает оборудование, которое соответствует по технологическим параметрам и по информации от системы мониторинга.

Рассмотрим на примере последовательность работы при проектировании техпроцесса листовой штамповки детали.

Постановка задачи – спроектировать технологический процесс изготовления детали и выбрать оборудование для каждого этапа технологического процесса с учетом состояния оборудования в цеху.

Исходные данные для выбора оборудования:

1. Данные, полученные из отдела главного технолога («Управление проектом»):
   — толщина материала — 1 мм;
   — заказ - несрочный (ПД 0,85);
   — изменение формы детали – формовка.
2. Данные, полученные от подсистемы выбора оснастки для листовой штамповки:
   — исходный материал —лист;
   — тип штампа – совмещенный;
   — раскрой – встречный;
   — длина контура детали — 2600 мм;
   — площадь детали — 420 000 мм^2;
   — сопротивление срезу $\sigma_{ср}$ = 36 кГ/мм^2.
3. Данные, полученные от подсистемы мониторинга состояния производства:
   — наличие пресса – есть, нет;
   — исправность пресса – исправен, неисправен;
   - занятость пресса – занят, свободен;
   — наличие рабочего – есть, нет.

Сначала технолог выполняет проектирование технологического процесса. Система запрашивает и предоставляет технологу исходные данные (рис. 2), а затем предлагает решение в виде последовательности операций технологического процесса (рис. 3).

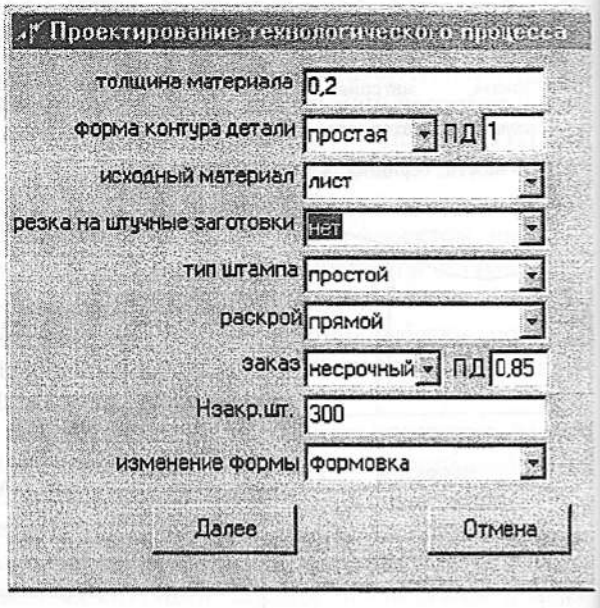


Рис. 2. Входные данные для выбора оборудования

Если ПД выполнения операции высокий (равен 1 или соответствует допустимому уровню риска в данной предметной области), значит эту операцию необходимо выполнять. В данном случае необходимо выполнить резку листа на полосы

формовку. Технолог может получить объяснение принятого решения. Он подтверждает выбранный машиной вариант технологического процесса (ТП) и может начать выбор оборудования для каждого этапа ТП.

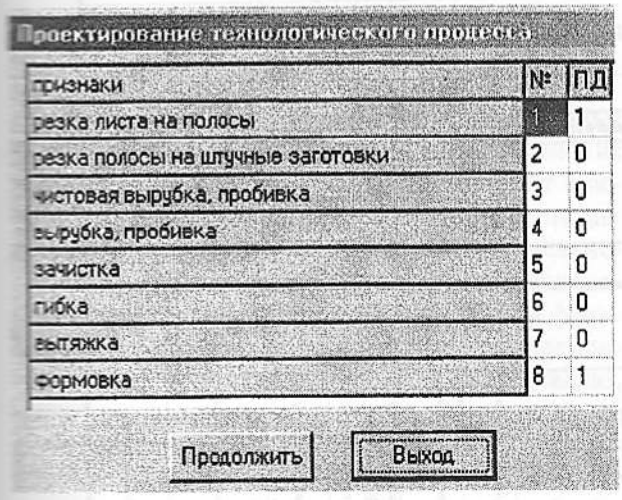


Рис. 3. Результат проектирования ТП

Для выбора типа оборудования необходимо нажать кнопку «Начать» напротив соответствующего этапа ТП. Результат выбора оборудования для резки листа на полосы представлен на рис. 4.

Можно выбрать гильотинные ножницы или дисковые ножницы с параллельными осями. Результат выбора оборудования для формовки представлен на рис. 5.

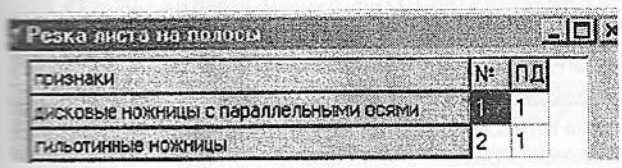


Рис. 4. Выбор оборудования для резки листа на полосы

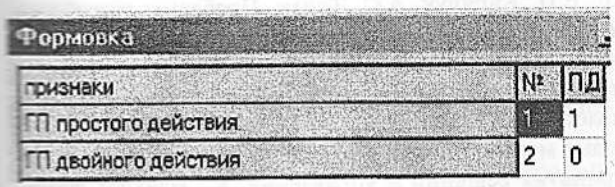


Рис. 5. Выбор типа оборудования для формовки

Система предлагает выбрать гидравлический пресс простого действия, так как выполняется операция «формовка».

Далее технологу необходимо выбрать пресс с учетом его параметров и данных от системы мониторинга. Для этого необходимо рассчитать работу, усилие. Система предоставляет технологу необходимую информацию из БД оборудования, в которую интеллектуальный агент уже поставил сведения от системы мониторинга (рис. 6).

Эти сведения вводят мастера участков цеха при помощи PALM-компьютера, а интеллектуальный агент обеспечивает их доставку в БД.

После завершения выбора оборудования для всех этапов ТП автоматически генерируется технологическая карта, представленная на рис. 7.

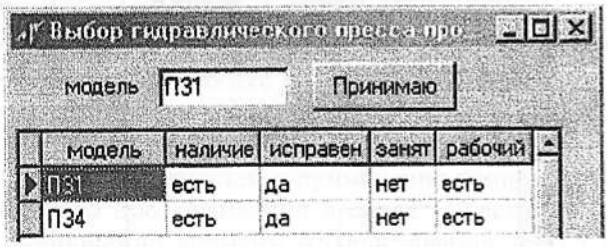


Рис. 6. Выбор гидравлического пресса простого действия для формовки

Технолог может редактировать карту, созданную машиной.

В автоматически сгенерированной технологической карте содержится следующая информация:

1. Расчетная информация:
-- вес отходов;
— % отходов;
— усилие.
2. Информация, полученная с использованием БЗ:
— тип раскроя;
— технологический процесс;
— тип штампа;
— оборудование для выполнения операций.

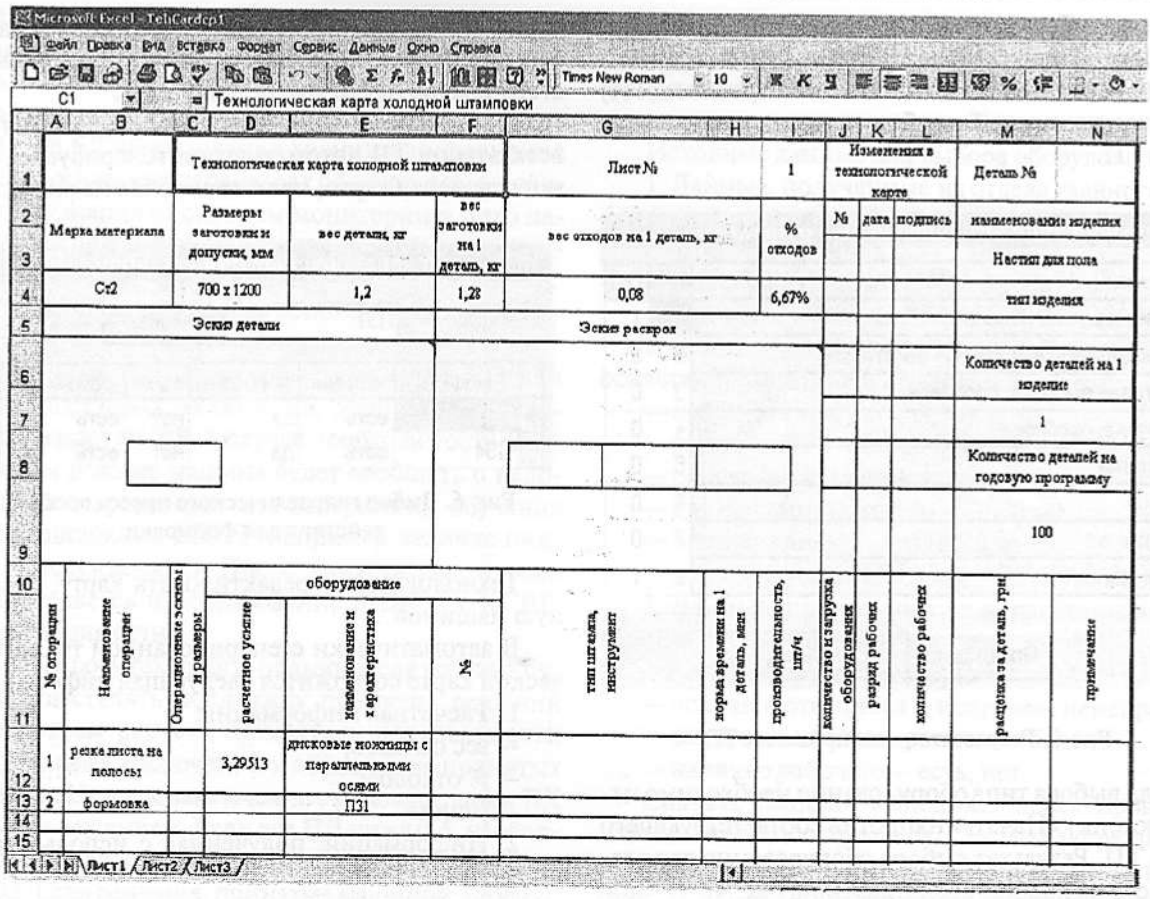
В качестве инструментального средства для разработки ИПК «Квант_ВО» было выбрано Delphi 7.0. Выбор мотивируется тем, что указанная версия включает в себя простую и удобную диалоговую оболочку, позволяющую эффективно отлаживать программы на уровне исходных текстов и оптимизировать их, повышая производительность. Кроме того, это средство позволило применить при разработке ИПК прогрессивный объектно-ориентированный подход к программированию, позволяющий повысить производительность и сократить затраты на сопровождение программного обеспечения.

Исходные данные поступают для обработки в виде следующих текстовых файлов:

– *_rules – хранит СПОЗ в виде логических высказываний, где имена логических переменных являются их порядковым номером, логическое «и» - «*», логическое «или» - «+», логическое следствие – «=». Через точку с запятой указывается ПД заданной импликации. Каждая строка заканчивается точкой с запятой;

– *_goals – хранит порядковые номера целевых признаков;

– *_semant – хранит семантику признаков ОПР. Сначала указывается номер признака, а через де-



Microsoft Excel - TehCard.cp1

Файл Правка Вид Вставка Формат Сервис Данные Окно Справка

Times New Roman 10

C1 = Технологическая карта холодной штамповки

	Технологическая карта холодной штамповки				Лист №		Изменения в технологической карте			Деталь №	
							№	дата	подпись		
Марка материала	Размеры заготовки и допуски, мм	вес детали, кг	вес заготовки на 1 деталь, кг	вес отходов на 1 деталь, кг	% отходов					наименование изделия	
										Настил для пола	
Ст2	700 x 1200	1,2	1,28	0,08	6,67%					тип изделия	
Эскиз детали			Эскиз раскроя								
										Количество деталей на 1 изделие	
										1	
										Количество деталей на годовую программу	
										100	

№ операции	Наименование операции	Операционные эскизы и размеры	оборудование			тип штампа, инструмент	норма времени на 1 деталь, мин	производительность, шт/ч	количество на загрузка оборудования	разряд рабочих	количество рабочих	расценка за деталь, грн.	примечание
			расчетное усилие	наименование и характеристика	№								
1	резка листа на полосы		3,29513	дисковые ножницы с параллельными осями									
2	формовка			П31									

Лист1 / Лист2 / Лист3

Рис. 7. Технологическая карта

фис – его семантическое значение. В конце строки – точка с запятой;

– *_pd – хранит ПД входных признаков. Сначала указывается номер признака, а через дефис – его ПД. В конце строки – точка с запятой.

Наличие указанных четырех файлов обязательно для обеспечения работы ИПК «Квант_ВО» в режиме поддержки принятия решений. В режиме обучения машина в диалоговом режиме помогает пользователю создать все необходимые файлы.

В режиме поддержки принятия решений машина в диалоговом режиме предлагает технологу решения, основанные на выводе из БЗ, которые он может принять или изменить, выполняет необходимые расчеты, генерирует автоматически технологические документы. Но окончательное решение по выбору оборудованию принимает технолог, а не компьютер.

В режиме обучения «КВАНТ_ВО» обеспечивает построение БЗ на основании СПОЗ для последующей поддержки принятия технологических решений.

3. Заключение

Созданная, с использованием vPAK3 метода для представления знаний, ИСППР «КВАНТ_ВО» позволит сократить сроки ТПП, снизить затраты, повысить качество принимаемых решений по выбору оборудования и обеспечить жизнеспособность предприятия в условиях рыночной конкуренции. Это достигается путем использования средств искусственной интеллектуализации процесса принятия решения при выборе оборудования.

Список литературы: 1. *Гордиенко Л.А., Киричук Е.П.* Проблема интеллектуальной поддержки принятия технологических решений в листовой штамповке // Авиационно-космическая техника и технология. 2004. Вып. 3. С. 83–89. 2. *Шостак И.В.* / И.В.Шостак, Л.А. Гордиенко, Е.П. Киричук, А.С. Топал. Проблемы разработки мультиагентной интеллектуальной интегрированной системы поддержки принятия решений в авиационном производстве // Авиационно-космическая техника и технология. 2003. Вып. 8. С. 14–22. 3. *Сироджа И.Б.* Квантовые модели и методы искусственного интеллекта для принятия решений и управления. К.: Наукова думка, 2002. 490 с. 4. *Сироджа И.Б., Петренко Т.Ю.* Метод разноуровневых алгоритмических квантов знаний для принятия производственных решений при недостатке или нечетких данных. К.: Наукова думка, 2000. 247 с. 5. *Киричук Е.П.* V-квантовые модели, методы и алгоритмы поддержки принятия решений при выборе листоштамповочного оборудования // Міжнародна наук.-техн. конф. "Інтегровані комп'ютерні технології в машинобудуванні": Тези доповідей. Харків "ХАІ", 2003. С. 262. 6. *Киричук Е.П.* Методика выбора оборудования для листовой штамповки // Радиоэлектронные и компьютерные системы. 2004. Вып. 2. С. 75–79.

Поступила в редколлегию 23.11.2004

УДК 519.713:681.326

Ваде Гриби, В.В. Елисеев, Хассан Ктейман, В.И. Хаханов, И.В. Хаханова

АНАЛИЗ НЕИСПРАВНОСТЕЙ ПРИ ПРОЕКТИРОВАНИИ ДИСКРЕТНЫХ ЛОГИЧЕСКИХ СЕТЕЙ

1. Введение

Проблема создания высокопроизводительных мозгоподобных ЭВМ и принципов построения их процессоров в виде логических сетей лежит в основе настоящих исследований. Актуальность работы определяется необходимостью значительного повышения быстродействия средств моделирования и генерации тестов для структурно- и функционально-сложных логических сетей, имплементация которых в цифровые системы на кристаллах, насчитывает миллионы вентилей. Существующие автоматические системы тестирования и моделирования известных фирм: Cadence, Mentor Graphics, Synopsys, Logic Vision способны обрабатывать устройства на кристаллах размерностью порядка миллионов вентилей за несколько часов (временные затраты относятся к рабочим станциям с частотой и оперативной памятью в 0,5 ГГц и 0,5 Гбайт). Тем не менее, предлагаемые модели и алгоритмические решения для их моделирования могут быть конкурентоспособными на рынке тестирования цифровых устройств большой размерности. Они позволяют на порядок повысить быстродействие анализа проекта после выполнения синтеза в целях определения качества тестов проверки неисправностей.

В настоящей статье разрабатываются методы диагностики логических сетей мозгоподобных ЭВМ. Применительно к нейронным сетым мозга диагностика логических сетей соответствует обнаружению дефектов в нейронных сетях и их исправлению.

Объект тестирования – цифровая система, реализующая логическую сеть в кристалле ASIC или PLD, спецификация которой представлена на языке VHDL.

Цель исследования – разработка быстродействующего метода моделирования одиночных константных неисправностей для оценки качества синтезируемых тестов для логических сетей, имплементируемых в кристаллы, содержащие миллионы вентилей.

Задачи исследования: 1. Создание обобщенной модели дедуктивно-параллельного анализа логической цифровой схемы. 2. Разработка дедуктивной модели структурно-функционального анализа цифровых схем в целях поиска множества сходящихся разветвлений (СР) и определения схемных древовидных структур (ДС). 3. Разработка метода моделирования неисправностей, основанного на раздельной обработке СР и ДС. 4. Алгоритмическая реализация моделирования на основе реконфигурирования модели устройства в процессе анализа неисправностей и применения процедур обратного прослеживания в целях существенного уменьшения времени моделирования тестов и оценки их качества.

Основу ОДП-метода (Backtraced Deductive-Parallel) – обратное дедуктивно-параллельное моделирование неисправностей – составляют: методы повышения быстродействия анализа неисправностей [1–3], дедуктивная модель транспортирования неисправностей [4, 5], параллельный метод обработки списков дефектов функционального элемента [4, 6] и алгоритм обратного прослеживания примитивов [7] при обработке цифрового устройства.

2. Обобщенная модель дедуктивно-параллельного анализа неисправностей

Модель дедуктивно-параллельного синхронного анализа неисправностей дискретного объекта позволяет за одну итерацию обработки схемы вычислять все дефекты, проверяемые на двоичном тест-векторе. Такая модель основана на решении следующего уравнения [3,4]:

$$L = T \oplus F, \qquad (1)$$

где $F = (F_{m+1}, F_{m+2}, ..., F_i, ... F_n)(i = \overline{m+1,n})$ – совокупность функций исправного поведения устройства; m – число его входов; $Y_i = F_i(X_{i1}, ..., X_{ij}, ..., X_{in_i})$ – n_i-входовой i-й элемент схемы, реализующий F_i для определения состояния линии (выхода) Y_i на тест-векторе T_t; где X_{ij} – j-й вход i-го элемента; двоичный тест $T = (T_1, T_2, ..., T_t, ..., T_k)$ – упорядоченная совокупность двоичных векторов, доопределенная в процессе исправного моделирования на множестве входных, внутренних и выходных линий, где тест-вектор задается в виде

$$T_t = (T_{t1}, T_{t2}, ..., T_{ti}, ..., T_{tn}), \qquad (2)$$

а его невходная координата вычисляется моделированием функции $T_{ti} = Y_i = F_i(X_{i1}, ..., X_{ij}, ..., X_{in_i})$ на тест-векторе T_t; $L = (L_1, L_2, ..., L_t, ..., L_k)$ – множество дедуктивных схем или моделей, определяемых выражением (1), где $L_t = (L_{t1}, L_{t2}, ..., L_{ti}, ..., L_{tn})$;

$$L_{ti} = T_t \oplus F_i \qquad (3)$$

– дедуктивная функция (ДФ) параллельного моделирования неисправностей на тест-векторе T_t, со-

ответствующая исправному элементу F_i, которая дает возможность вычислять список входных неисправностей, транспортируемых на выход элемента F_i [8].

Понятие синхронности введенной модели (1) определяется условием: $\Delta t = (t_{j+1} - t_j) >> \tau >> \tau_i$, когда интервал времени между сменой входных наборов $(t_{j+1} - t_j)$, подаваемых на схему, намного больше максимальной задержки схемы τ и элемента τ_i. Это позволяет исключить время как несущественный параметр [8], что используется в технологиях моделирования и синтеза тестов.

В общем случае, когда функция устройства представлена таблицей истинности, применение формулы (1) позволяет получить для заданного тест-вектора T_t таблицу транспортирования неисправностей, по которой можно записать ДФ моделирования дефектов. Пример получения такой функции представлен в следующем виде (первое слагаемое – тест-вектор, второе и результат – таблицы истинности и транспортирования дефектов):

X_1	X_2	Y_1
0	1	0

$\oplus$

X_1	X_2	Y_1
0	0	0
0	1	0
1	0	1
1	1	1

$=$

X_1	X_2	L_1
0	1	0
0	0	0
1	1	1
1	0	1

Здесь дедуктивная функция $L_1 = X_1 X_2 \vee X_1 \overline{X}_2$ записана в виде дизъюнктивной нормальной формы по конституентам единицы таблицы транспортирования дефектов.

С учетом разбиения теста на составляющие векторы уравнение (1) получения ДФ для $T_t \in T$ принимает следующий вид: $L_t = T_t \oplus F$. Если функциональное описание цифрового устройства представлено компонентами (примитивами), формирующими состояния всех линий схемы, то в качестве формулы преобразования исправной модели примитива F_i на тест-векторе T_t в дедуктивную функцию L_{ti} выступает выражение

$$L_{ti} = T_t \oplus F_i = f_{ti}[(X_{t1} \oplus T_{t1}),(X_{t2} \oplus T_{t2}),\ldots$$
$$\ldots,(X_{ij} \oplus T_{tj}),\ldots,(X_{in_i} \oplus T_{tn_i})] \oplus T_{ti}, \qquad (4)$$

которое является основой дедуктивного анализа цифровых проектов [3, 6].

Пример 1. Получить дедуктивные функции параллельного моделирования неисправностей на исчерпывающем тесте для базиса логических элементов И, ИЛИ, НЕ.

Используя выражение (4), выполним следующие очевидные преобразования для логического элемента И:

$$L[T = (00,01,10,11), F = (X_1 \wedge X_2)] =$$
$$= L\{(\overline{x}_1 \overline{x}_2 \vee \overline{x}_1 x_2 \vee x_1 \overline{x}_2 \vee x_1 x_2) \wedge$$
$$\wedge[(X_1 \oplus T_{t1} \wedge X_2 \oplus T_{t2}) \oplus T_{t3})]\} =$$
$$= (\overline{x}_1 \overline{x}_2)\{[(X_1 \oplus 0) \wedge (X_2 \oplus 0)] \oplus 0\} \vee$$
$$\vee (\overline{x}_1 x_2)\{[(X_1 \oplus 0) \wedge (X_2 \oplus 1)] \oplus 0\} \vee$$
$$\vee (x_1 \overline{x}_2)\{[(X_1 \oplus 1) \wedge (X_2 \oplus 0)] \oplus 0\} \vee$$
$$\vee (x_1 x_2)\{[(X_1 \oplus 1) \wedge (X_2 \oplus 1)] \oplus 1\} =$$
$$= (\overline{x}_1 \overline{x}_2)(X_1 \wedge X_2) \vee (\overline{x}_1 x_2)(X_1 \wedge \overline{X}_2) \vee$$
$$\vee (x_1 \overline{x}_2)(\overline{X}_1 \wedge X_2) \vee (x_1 x_2)(X_1 \vee X_2);$$

Аналогичные преобразования для логического элемента ИЛИ дают следующий результат:

$$L[T = (00,01,10,11), F = (X_1 \vee X_2)] =$$
$$= (\overline{x}_1 \overline{x}_2)(X_1 \vee X_2) \vee (\overline{x}_1 x_2)(\overline{X}_1 \wedge X_2) \vee$$
$$\vee (x_1 \overline{x}_2)(X_1 \wedge \overline{X}_2) \vee (x_1 x_2)(X_1 \wedge X_2).$$

Здесь $T_t = (T_{t1}, T_{t2}, T_{t3}), (t = \overline{1,4})$ – тест-вектор, имеющий 3 координаты, где последняя определяет состояние выхода двухвходового элемента И (ИЛИ).

В следующем преобразовании для инвертора тест-вектор имеет 2 координаты: $T_t = (T_{t1}, T_{t2})(t = \overline{1,2})$, где последняя – состояние выхода элемента:

$$L[T = (0,1), F = \overline{X}_1] = L\{(\overline{x}_1 \vee x_1)[(\overline{X_1 \oplus T_{t1}}) \oplus T_{t2}]\} =$$
$$= \overline{x}_1 \wedge [(\overline{X_1 \oplus 0}) \oplus 1] \vee x_1 [(\overline{X_1 \oplus 1}) \oplus 0] =$$
$$= \overline{x}_1 \overline{\overline{X}_1} \vee x_1 \overline{\overline{X}_1} = \overline{x}_1 X_1 \vee x_1 X_1.$$

Последнее выражение иллюстрирует инвариантность операции инверсии ко входному набору для транспортирования дефектов. Она трансформируется в повторитель. Поэтому данная функция не фигурирует на выходах дедуктивных элементов. Совместная аппаратурная реализация ДФ для двухвходовых элементов И, ИЛИ на исчерпывающем тесте представлена универсальным функциональным примитивом (рис. 1) дедуктивно-параллельного анализа неисправностей.

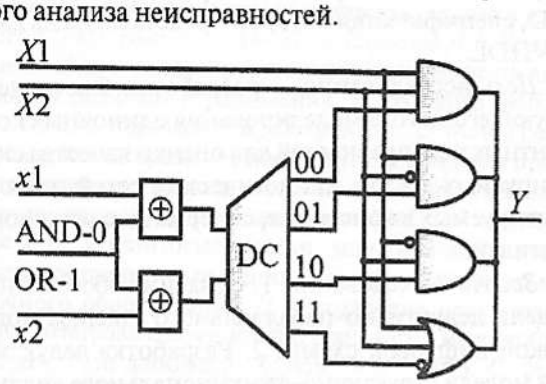


Рис. 1. Симулятор неисправностей

В симуляторе представлены булевы ($x1$, $x2$) и регистровые ($X1$, $X2$) переменные, сигнал выбора

типа исправной функции (AND, OR), выходная регистровая переменная Y. Состояния двоичных входов $x1$, $x2$ и AND/OR выбирают одну из четырех дедуктивных функций для получения вектора Y проверяемых неисправностей. Для иллюстрации параллельного моделирования входных 4-разрядных векторов неисправностей в целях получения на выходе Y множества проверяемых дефектов для логических элементов 2И, 2ИЛИ используется следующая таблица:

And/Or	x_1x_2	X_1	X_2	Y
0	00	0111	1011	0011
1	00	0111	1011	1111
0	11	0101	0110	0111
1	11	1101	0111	0101

Применение такого симулятора дает возможность трансформировать вентильную модель F исправного поведения схемы в дедуктивную L, которая инвариантна в смысле универсальности тестовым наборам и не предполагает в процессе моделирования использовать модель F. Поэтому симулятор, как аппаратурная модель ДФ, ориентирован на создание встроенных средств дедуктивно-параллельного моделирования, повышающих быстродействие анализа в 10–1000 раз по сравнению с программной реализацией. Но при этом соотношение объемов вентильных (после синтеза) моделей исправного моделирования и анализа неисправностей составляет 1:10. Подход аппаратурного анализа неисправностей направлен на расширение функциональных возможностей встроенных средств исправного моделирования (HES™ – Hardware Embedded Simulator) фирмы Aldec [9]. Вычислительная сложность обработки цифрового устройства, состоящего из n вентилей, равна $Q=(2n^2\tau)/W$, где τ – время выполнения регистровой операции (AND, OR, NOT); W – разрядность регистра.

3. Дедуктивный метод структурного анализа

Метод необходим для определения сходящихся разветвлений в структуре цифрового устройства.

Имеется вершина $V_j \in V$ в ориентированном графе G линий схемы. По отношению к смежным вершинам-предшественникам, которые являются прообразами $f^{-1}(V_j)$, упомянутая вершина $V_j \in V$ является образом (рис. 2), где $f^{-1}(...)$, – оператор взятия прообраза.

Определение 1. Каждая вершина бесконтурного графа имеет единственное объединение ее образа и суперпозиции прообразов:

$$f^*(V_j) = f^{-1}\left(f^{-1}\left(\ldots\left(f^{-1}(V_j)\right)\right)\right) \cup V_j, \qquad (5)$$

которое называется экстраобразом. Иначе, экстраобраз есть множество всех предшественников для V_j, включая и вершину V_j.

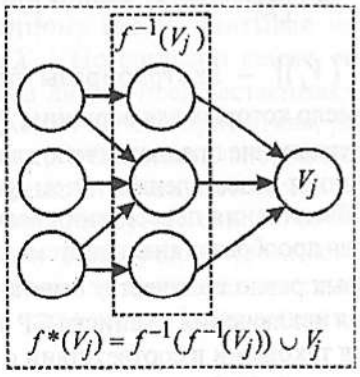


Рис. 2. Смежность как соответствие прообразов образу

Утверждение 1. Все вершины (линии), входящие в контур, имеют одинаковые экстраобразы.

Это является следствием таких фактов: 1) все линии, принадлежащие контуру, достижимы по отношению друг к другу; 2) каждая линия, входящая в контур, имеет в качестве предшественника любую вершину, от которой существует путь к одной из контурных линий. Отсюда следует, что все вершины $V_j \in V$, принадлежащие контуру, имеют одинаковые экстраобразы [10].

Сходящемуся разветвлению могут предшествовать линии, которые не являются таковыми, но проявляются на вершине-схождении как СР. Пример таких линий есть вершины, составляющие путь без ветвлений, заканчивающийся на линии СР.

Утверждение 2. Для того чтобы устранить всех предшественников сходящегося разветвления, входящих в множество V^j кандидатов в СР для вершины V_j, но не являющихся СР, необходимо и достаточно вычесть из множества V^j объединение пересечений всех пар сочетаний $C^2_{n_j}$ экстраобразов $f^*\left(f^{-1}(V_j)\right)$ для непосредственных предшественников вершины V_j, где n_j – число предшественников.

Процедура дедуктивного анализа. Если граф – ориентированный, без контуров глобальных обратных связей, то стратегия поиска сходящихся разветвлений сводится к одноразовому проходу всех его вершин на основе процедуры дедуктивной обработки $V_j \in V$, включающей операции:

$$1)\ V^j = \bigcup_{C^2_{nj}} \left[f^*\left(f_p^{-1}(V_j)\right) \bigcap_{i=\overline{1,n_{j-1}}}^{\substack{p=i \\ q=\overline{i+1,n_j}}} f^*\left(f_q^{-1}(v_j)\right) \right];$$

$$2)\ V^j = V^j \setminus \bigcup_{C^2_{mj}} \left[f^*\left(V_p^j\right) \bigcap_{i=\overline{1,m_{j-1}}}^{\substack{p=i \\ q=\overline{i+1,m_j}}} f^*\left(V_q^j\right) \right];$$

$$3)\ V^{RC} = V^{RC} \bigcup V^{j}; \qquad (6)$$

$$4)\ V_j = \left[\bigcup_{i=1}^{n_j} f^{*}\left(f_i^{-1}\left(V_j\right)\right) \cup V_j\right] \setminus V^{RC}.$$

где $f^{*}\left(f_i^{-1}\left(V_j\right)\right)$ – экстраобразы прообразов $f^{-1}\left(V_j\right)$, число которых для вершины V_j равно n_j.

Первое уравнение предназначено для определения сходящихся разветвлений путем выполнения операции объединения пересечений всех пар экстраобразов для прообразов анализируемой вершины, число которых равно количеству сочетаний – $C_{n_j}^{2}$. Второе – для исключения из списка СР вершин, не являющихся таковыми в соответствии с утверждением 2. Здесь $f^{*}\left(V_i^{j}\right)$ – экстраобраз вершины, включенной в список кандидатов в СР при анализе $V_j \in V$, число которых равно m_j. Третье уравнение предназначено для накопления множества СР в целях его последующего вычитания из списков предшественников для каждой вершины. Оно необходимо для уменьшения мощности прообразов обрабатываемой вершины, которое будет существенней при больших коэффициентах разветвлений у линий предшественников, уже отнесенных к списку V^{RC}. Четвертое уравнение предназначено для формирования экстраобраза обрабатываемой вершины $V_j \in V$.

Время, затрачиваемое на поиск СР при работе описанного препроцессора (6), практически не отличается от времени моделирования дефектов на входном наборе дедуктивным или параллельным методом.

Определение 2. Линия, принадлежащая контуру обратной связи, является СР, если она имеет более одной исходящей дуги, а также схождение на линии, не принадлежащей данному контуру.

Если в схеме присутствуют глобальные обратные связи, то перед выполнением процедуры поиска СР необходимо сделать их псевдоразрыв, например по алгоритму, приведенному в [8], а затем применить описанные выше модели идентификации СР.

4. Топологический ОДП-метод моделирования

Использованию процедуры обратного прослеживания [7] в общем случае препятствует невозможность выполнения одномерной активизации дефектов, которая может быть проиллюстрирована двумя вариантами некорректности, представленными следующим примером.

Пример 2. Выполнить анализ схем (рис. 3) в целях определения множества проверяемых дефектов на заданных тест-векторах с помощью процедуры обратного прослеживания.

В левой схеме константная неисправность 2^1 ложно проверяется на одномерном пути 2-5-6 с помощью процедуры обратного прослеживания. Однако при этом не учитывается тот факт, что данный дефект изменяет состояние линии 4 с 1 на 0, что создает условия запрета транспортирования неисправности на выход схемы. Поэтому неисправность на линии $a - 2^1$ является ложнопроверяемой на наборе (101) при рассмотрении ее транспортирования по одномерному пути активизации.

Правая схема является примером альтернативной ситуации – к выходу 6 нет одномерного пути транспортирования дефектов с внешних входов и, в частности, от линии b. Тем не менее, неисправность 2^1 на упомянутой линии проверяется, поскольку ее присутствие создает кратный дефект на линиях 4 и 5, который изменяет состояние выходной линии схемы.

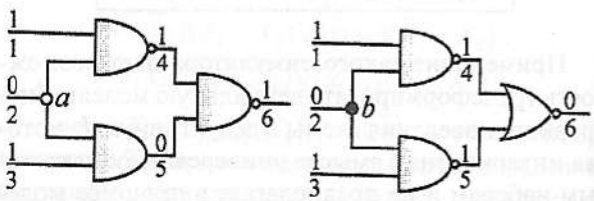


Рис. 3. Ложная проверка и непроверка дефектов

В обоих случаях имеется существенная некорректность, которая не позволяет использовать идею одномерности при отслеживании пути транспортирования неисправностей от выходов ко входам схемы.

Учитывая, что некорректность связана исключительно с одномерной активизацией неисправностей сходящихся разветвлений, далее предлагается сначала выполнить их моделирование, а затем исключить из рассмотрения, сделав структуру схемы древовидной и пригодной для одномерного обратного прослеживания.

Поскольку ниже рассматривается анализ неисправностей в привязке к топологии схемы цифрового устройства, то естественным представляется назвать метод анализа дедуктивно-топологическим (ДТ) на основе обратного прослеживания. Основные пункты стратегии моделирования (см. рис. 4) можно представить шагами:

1. Препроцессор структурного анализа модели схемы – осуществляет поиск СР в схеме цифрового устройства. Вычислительная сложность данной процедуры $Q_r = n^2$, но она является разовой и поэтому практически не влияет на быстродействие моделирования входных наборов в целом.

2. Фаза исправного моделирования цифровой схемы. Предназначена для определения реакции всех невходных линий устройства на входной набор $T_i \in T = [T_{ti}]$.

3. Преобразование схемы на текущем тест-векторе в дедуктивную модель по правилам (4).

4. Моделирование неисправностей сходящихся разветвлений по дедуктивной модели схемы, количество которых значительно меньше остальных линий;

5. Обратное одномерное прослеживание проверяемых неисправностей древовидных подграфов по дедуктивной модели устройства.

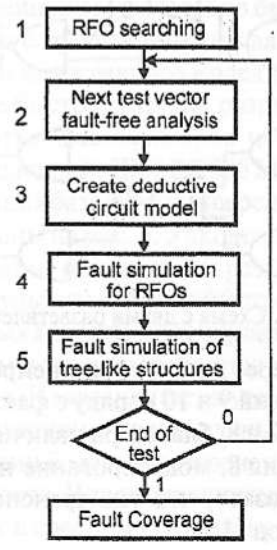


Рис. 4. Стратегия ДТ-моделирования

Теоретическим обоснованием применения данного пункта является доказательство следующих теорем и формулировка следствий из них.

Лемма. Сходящиеся разветвления в комбинационной схеме являются причиной появления кратных дефектов на входах примитивов.

Доказательство. Пусть комбинационная схема не имеет сходящихся разветвлений. В этом случае она представлена древовидной структурой. Тогда пересечение подграфов-предшественников (экстраобразов) для любых входов примитивного элемента определяется как

$$f^*(X_i) \cap f^*(X_j) = \varnothing . \qquad (7)$$

Это справедливо для примитивов первого уровня, где входы элементов являются входами схемы, занумерованными различными идентификаторами линий. Если примитив находится внутри схемы, то в силу древовидности (7) любые два входа не будут иметь хотя бы одного общего предшественника. Иначе нарушится условие древовидности графовой структуры. Таким образом, два любых входа примитива древовидной схемы, не имея общих предшественников, не будут иметь и общих неисправностей на конкретном двоичном входном наборе, которые могут быть транспортированы через рассматриваемый элемент.

Теорема 1. *Для древовидной структуры дедуктивной схемы инверсный вход примитива И: $\overline{X}_i \in L$ запрещает транспортирование всех неисправностей, принадлежащих линиям-предшественникам.*

Доказательство. Во-первых, инверсия на входе может иметь место на примитиве, имеющем две и более входные линии, во-вторых, она означает вычитание списка неисправностей, принадлежащего данному входу, учитывая изоморфизм $\overline{X}_i X_j \approx X_j \setminus X_i$. Но согласно лемме, пересечение двух списков линий-предшественников, относящихся ко входам одного примитива, равно пустому множеству (7). Следовательно, вычитание списка линий-предшественников никогда не уменьшит мощность любого другого списка, относящегося к неинверсному входу рассматриваемого элемента. Таким образом, для древовидных структур комбинационных схем всегда будет выполняться условие:

$$f^*(X_i) \setminus f^*(X_j) = f^*(X_i) \leftarrow f^*(X_i) \cap f^*(X_j) = \varnothing .$$

Теорема 2. *Если в древовидной структуре дедуктивной схемы L существует примитив И, имеющий более одного неинверсного входа $X_i X_j$, то такой элемент запрещает транспортирование всех неисправностей, принадлежащих линиям-предшественникам всех его входов.*

Доказательство. Наличие двух и более неинверсных, прямых входов в дедуктивном элементе означает пересечение списков линий, являющихся предшественниками для рассматриваемых входов. Но согласно лемме, входы одного примитива ДС не имеют общих предшественников, отсюда следует, что пересечение упомянутых списков будет всегда равно пустому множеству:

$$f^*(X_i) \cap f^*(X_j) = \varnothing \leftarrow X_i X_j .$$

Если же дедуктивный элемент имеет входы с инверсией, то по теореме 1 вычитание из полученного пустого множества предшественников любого непустого, принадлежащего входу с инверсией, дает также пустой результат.

Следовательно, дедуктивный элемент И, имеющий более одного неинверсного входа, не пропустит через себя неисправности линий предшественников.

Следствия: 1) Инверсный вход дедуктивного элемента И запрещает активизацию всех неисправностей линий-предшественников, относящихся к данному входу. 2) Если дедуктивный примитив имеет более одного неинверсного входа, то все его входы следует определить инверсными. 3) Дедуктивный элемент И пропустит неисправности линий-предшественников только по неинверсному входу, который должен быть единственным. 4) Дедуктивный элемент ИЛИ не может иметь инверсных входов. 5) Сходящееся разветвление отмечается инверсией (кружком на линии ветвления), если его неисправность не проверяется на тест-векторе. 6) Инверсия на линии, входной или выходной, является условием разрыва активизации неисправ-

ностей и прекращения обратного прослеживания дефектов по рассматриваемой ветви древовидной структуры. 7) Проверяемую на тест-векторе линию сходящегося разветвления (на топологическом рисунке схемы далее отмечается жирным кружочком) при выполнении процедуры обратного прослеживания следует рассматривать как наблюдаемый выход схемы. 8) Интерпретация результата топологического моделирования: неисправности линий, не отмеченные знаками инверсий на дедуктивной модели схемы, проверяются.

В качестве иллюстрации основных шагов топологического моделирования ниже предлагаются два примера.

Пример 3. Пусть дано цифровое устройство (рис. 5, первая схема, при рассмотрении сверху вниз), содержащее 3 сходящихся разветвления. Входной вектор: 1011111. Необходимо определить проверяемые дефекты константного типа. Состояния линий после исправного моделирования представлены в скобках. Результат процесса преобразования исправной модели устройства в дедуктивную и моделирование неисправностей линий СР зафиксирован на второй схеме. Все неисправности разветвлений (2, 10, 13), инверсные по отношению к исправному состоянию этих линий, являются проверяемыми. Этот факт отмечен на схеме черными кружочками.

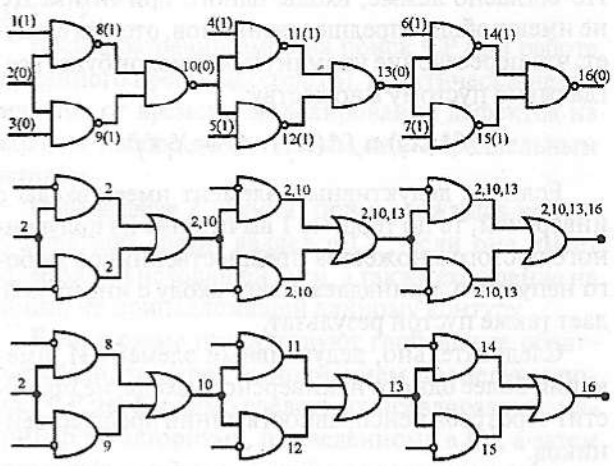


Рис. 5. Моделирование схемы с разветвлениями

Процедура обратного прослеживания неисправностей в целях определения списка проверяемых заключается в построении максимальных древовидных подграфов, ограниченных на топологии прозрачными кружочками. При этом черные кружочки есть проверяемые линии разветвлений, которые следует рассматривать на третьей схеме (см. рис. 5) как наблюдаемые выходы. Здесь же обозначены все неисправности, проверяемые на тест-векторе: {2, 8, 9, 10, 11, 12, 13, 14, 15, 16}. Иллюстрация результата моделирования: неисправности линий, не отмеченные прозрачными кружочками (знаками инверсии), проверяются.

Пример 4. Определить список проверяемых на тест-векторе 10111 неисправностей для цифрового устройства, представленного на рис. 6, содержащего 2 сходящихся разветвления.

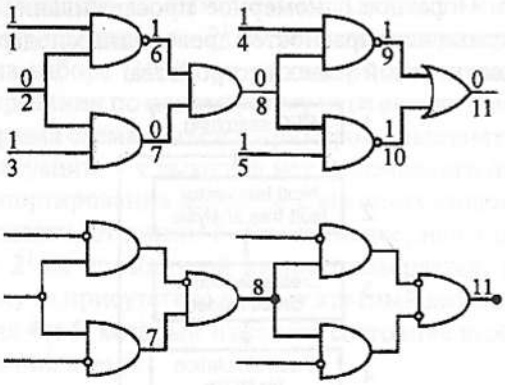


Рис. 6. Схема с двумя разветвлениями

Здесь интерес представляет непроверка неисправностей линий 9 и 10 наряду с фактом проверяемости линий 7 и 8, благодаря наличию сходящегося разветвления 8, моделирование неисправности которого показало, что она транспортируется на внешний выход 11.

На рис. 7 представлена структура эволюционного развития и интеграции методов моделирования неисправностей. В основу положены дедуктивный (Д – D) и параллельный (П – P) алгоритмы. Далее был разработан универсальный, но относительно медленный, дедуктивно-параллельный метод (ДП – DP) [11]. Затем был реализован (ОДП – BDP) метод [12], ориентированный на быструю обработку моделей цифровых систем. Последний был модифицирован к ТД (TD)-методу, который ориентирован на вентильный уровень представления цифровых систем. При этом в описании устройства предварительно осуществляется поиск множества сходящихся разветвлений и выделение древовидных структур как дополнение к СР. Для моделирования сходящихся разветвлений (RFO) используется дедуктивно-параллельный алгоритм для анализа древовидных структур (TLS) – ТД-метод. Такое разделение функций обработки цифровых схем большой размерности позволяет как минимум на порядок повысить быстродействие моделирования неисправностей по сравнению с дедуктивным и параллельным методами.

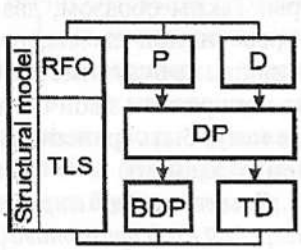


Рис. 7. Эволюция методов моделирования

5. Заключение

Предложенный метод моделирования неисправностей ориентирован на обработку логических сетей, имплементируемых в кристаллы, содержащие миллионы вентилей. Тестовые эксперименты программной реализации метода на сотнях цифровых комбинационных и последовательностных схем дали хорошие результаты по быстродействию по сравнению с традиционными алгоритмами параллельного и дедуктивного моделирования. Примеры анализа быстродействия разработанного метода (обработка тест-примеров на 1000 входных последовательностей, IBM PC 500 МГц, 256 Мбайт) и существующих базовых (университетские реализации) показаны на рис. 8. Ускорение моделирования составляет не менее десяти раз. На рис. 9 представлены результаты анализа быстродействия трех реализованных методов моделирования цифровых схем на одном и том же компьютере при обработке 1000 векторов. Показано преимущество ТД- метода перед дедуктивно-параллельным. Выигрыш в быстродействии более существенен для схем большой размерности. Число сходящихся разветвлений в тест-схемах в среднем составляет 20% от общего количества линий.

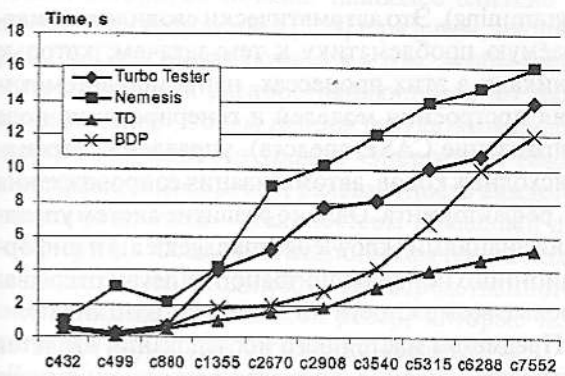


Рис. 8. Анализ быстродействия систем моделирования

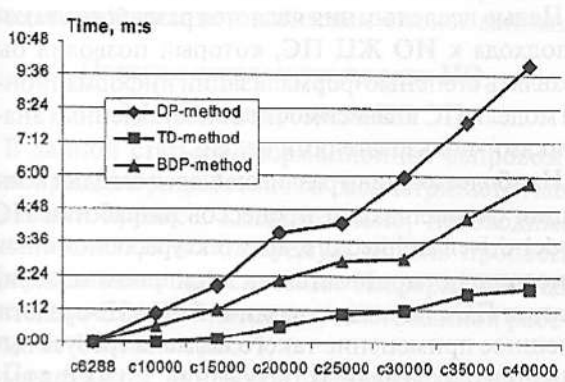


Рис. 9. Анализ быстродействия методов моделирования

Таким образом, основным результатом данной работы является усовершенствование дедуктивно-параллельного метода моделирования неисправностей цифровых систем, заключающееся в:

1) создании обобщенной модели процесса дедуктивно-параллельного анализа цифровых схем, имеющей вычислительную сложность n^2 от числа линий схемы;

2) разработке дедуктивного алгоритма структурно-функционального анализа цифровых устройств в целях определения сходящихся разветвлений и декомпозиции структуры для раздельного моделирования СР и древовидных структур;

4) разработке топологического алгоритма моделирования неисправностей для древовидной структуры цифрового устройства, имеющего линейную вычислительную сложность от числа линий схемы.

Литература: 1. *Wang X., Hill F.G., Mi Zh.* A sequential circuit faulf simulation by surrogate fault propagation // Proc. 1989 IEEE International test conference, IEEE Computer society, 1989. P. 9-18. 2. *Nishida T., Miyamoto S., Kozawa T., Satoh K.* RFSIM: Reduced faült simulator // IEEE Transactions on computer-aided design. 1987. Vol. CAD-6, No 3. P. 392-402. 3. *Hahanov V.I., Babich A.V., Hyduke S.M.* Test Generation and Fault Simulation Methods on the Basis of Cubic Algebra for Digital Devices. Proceedings of the Euromicro Symposium on Digital Systems Design DSD2001. Warsaw, Poland. September, 4-6, 2001. P. 228-235. 4. *Хаханов В.И., Хак Х.М. Джахирул, Масуд М.Д. Мехеди.* Модели анализа неисправностей цифровых систем на основе FPGA, CPLD // Технология и конструирование в электронной аппаратуре. 2001. № 2. С. 3-11. 5. *Levendel Y.H., Menon P.R.* Comparison of fault simulation methods – Treatment of unknown signal values // Journal of digital systems. 1980. Vol. 4. P. 443-459. 6. *Abramovici M., Breuer M.A. and Friedman A.D.* Digital systems testing and testable design. Computer Science Press. 1998. 652 p. 7. *Убар Р.Р.* Анализ диагностических тестов для комбинационных цифровых схем методом обратного прослеживания неисправностей // Автоматика и телемеханика. 1977. №8. С.168-176. 8. *Автоматизированное проектирование цифровых устройств* / С.С.Бадулин, Ю.М.Барнаулов и др./ Под ред. С.С. Бадулина. М.: Радио и связь. 1981. 240 с. 9. *Active-HDL* User's Guid. Second Edition. Copyright. Aldec Inc. 1999. 213p. 10. *Горбатов В.А.* Основы дискретной математики. М.: Высш. шк., 1986. 311 с. 11. Hahanov V.I., Sysenko I.Y., Skvortsova O.B. Test Generator Used Genetic Algorithms and Reconfigurable Deductive-Parallel Fault Simulation Method for Digital Devices.– 9-th International Conference. Mixed Design of Integrated Circuits and Systems. – Poland. – Wroclaw – 2002. – P. 549-554. 12. V. Hahanov, R. Ubar and S. Hyduke. Back-Traced Deductive-Parallel Fault Simulation for Digital Systems.– Proceedings of Euromicro Symposium on Digital Systems Design.– Belek-Antalya, Turkey.– 2003.– P. 370-377.

Поступила в редколлегию 11.11.2004

УДК 004.89

А.Л. Ерохин, Р.А. Полунин

ОБ ИНТЕЛЛЕКТУАЛИЗАЦИИ ПРОЦЕССА ПОДДЕРЖКИ ЖИЗНЕННОГО ЦИКЛА ПРОГРАММНЫХ СИСТЕМ

1. Введение

В процессе поддержки жизненного цикла (ЖЦ) программных систем (ПС) важную роль играет информационное обеспечение. Термин «информационное обеспечение» заимствован в сферу информационных технологий из теории АСУ, сама по себе поддержка ЖЦ больших коммерческих ПС на сегодня является достаточно сложным процессом. В широком смысле под информационным обеспечением системы понимают создание информационных условий функционирования системы, обеспечение необходимой информацией, включение в систему средств поиска, получения, хранения, накопления, передачи, обработки информации, организацию банков данных. Под информационным обеспечением (ИО) в задачах, входящих в процесс поддержки ЖЦ ПС, например моделирования, проектирования и разработки, обычно понимают комплекс мер по организации создания, хранения и поиска тех или иных информационных ресурсов, имеющих отношение к ЖЦ ПС. Примерами таких ресурсов могут служить различные формальные и неформальные информационные модели (например диаграммы в нотациях UML, IDEF), специализированные базы данных (сведения о логической и других структурах конкретной ПС, о взаимозависимостях ее подсистем, временные планы работ по задачам разработки), и, что самое важное, различного рода информация, которая может быть автоматически или полуавтоматически получена из исходных кодов ПС.

В настоящее время задачи ИО поддержки ЖЦ ПС решаются исключительно в контексте автоматизации тех или иных стандартизованных процессов разработки, таких как RUP (Rational Unified Process). Постановка решаемых задач ИО логически вытекает из структуры этих процессов и преследуемых целей: повышение качества кода, рефакторинг, тестирование. Однако эти задачи пока что решаются на прикладном уровне, без привлечения современных подходов к управлению знаниями, таких как построение баз знаний, автоматизация семантического поиска и т.д.

С другой стороны, существует широкий спектр алгоритмов и общих подходов к семантическому поиску в текстовых и других источниках плохо структурированной информации, применяемых в современных поисковых системах. Все многообразие моделей традиционного информационного поиска принято делить на три вида: теоретико-множественные (булевская, нечетких множеств, расширенная булевская), алгебраические (векторная, обобщенная векторная, латентно-семантическая, нейросетевая) и вероятностные [1, 2]. Предметом поиска в поисковых системах выступает информационная потребность пользователя, выраженная в запросе. В рамках *информационного поиска* изучаются все составляющие процесса поиска, а именно: предварительная обработка информации (индексирование), обработка и исполнение запроса, ранжирование, пользовательский интерфейс и обратная связь.

2. Постановка задачи исследования

До недавнего времени задачи информационного обеспечения поддержки жизненного цикла (ЖЦ) ПС рассматривались исключительно в рамках стандартизированных процессов разработки (таких как SADT, SSADM, RUP, SCRUM, eXtreme Programming). Это автоматически сводило рассматриваемую проблематику к тем задачам, которые возникали в этих процессах, например автоматизация построения моделей и генерирования кода (направление CASE-средств), управление версиями исходных кодов, автоматизация сопровождения ПС, рефакторинга. Однако развитие систем управления знаниями (knowledge management) и информационного поиска (information retrieval) открывает новые возможности интеллектуализации ИО.

Предметом настоящего исследования является интеллектуализация процесса информационного обеспечения в контексте поддержки жизненного цикла (ЖЦ) программных систем (ПС).

Целью исследования является разработка такого подхода к ИО ЖЦ ПС, который позволил бы управлять степенью формализации информационной модели ПС в зависимости от поставленных аналитиками-пользователями целей.

Наиболее активно развивающаяся сегодня концепция формализации процессов разработки ПС (model-driven architecture; архитектура, основанная на моделях) основывается на априорном моделировании ПС с помощью различных CASE-средств. Успешное применение такого подхода требует выполнения определенных требований, таких как высокая квалификация проектировщиков, высокий уровень формализации требований к ПС и организации процесса разработки. К сожалению, на практике эти требования очень часто хотя и выполнимы, но далеко не в той степени, чтобы сделать

возможным полный переход к процессам разработки ПС, основанным на моделях. Кроме того, существует большое поле задач, связанных с анализом уже созданных коммерческих ПС, для которых автоматизация построения моделей (например с помощью *обратного проектирования*) крайне затруднена. Этими факторами и обусловлена **актуальность исследования**.

В связи с этим для существующей ПС целесообразно несколько снизить требования к степени формализации (для упрощения разработки моделей) и всеохватности строящихся по ней моделей (для ускорения их автоматического построения для данной ПС). При решении прикладных задач анализа ПС компенсировать увеличение количества моделей и снижение их степени формализации можно за счет применения принципов и технологий информационного поиска (information retrieval) и поиска данных (data retrieval). Теоретические основы информационного поиска были заложены в 60–70-х годах (Swets, Rijsbergen, Robertson, Good, Cooper) и с тех пор активно разрабатываются ведущими исследовательскими группами и активно применяются как для поиска в глобальных и локальных сетях и базах данных, так и для организации, например, рефакторинга программного кода.

Идеи алгоритма поиска, наиболее близкие к нашей задаче, изложены в [3, 4] и развиты, например, в [5]. Суть [3] заключается в том, чтобы применить для поиска распространяющуюся по ребрам графа волну. Объем набора результатов будет зависеть от начального уровня «энергии» и весовых коэффициентов ребер, релевантность каждого элемента результата – количеством дошедшей до него «энергии». Основным недостатком такого подхода является необходимость непосредственного хранения в базе знаний всех ребер, которые используются при распространении «волны».

Для решения поставленной цели рассмотрим новый подход к интеллектуализации задач информационного обеспечения (ИО), который ориентирован на информационный поиск и поиск данных.

3. Подход к интеллектуализации ИО

3.1. Модель процесса ИО

В данной статье информационное сопровождение процесса разработки ПО рассматривается как отдельный процесс. Следовательно, необходимо сначала описать используемую модель процесса информационного обеспечения. Информационная модель процесса ИО, используемая в данной работе, представлена на рис. 1.

Под блоком №1 подразумеваются различные среды разработки приложений, управления версиями, репозитарии документов и т. п., в которых и с помощью которых ведется проектирование и разработка (в том числе формализованные) программного продукта.

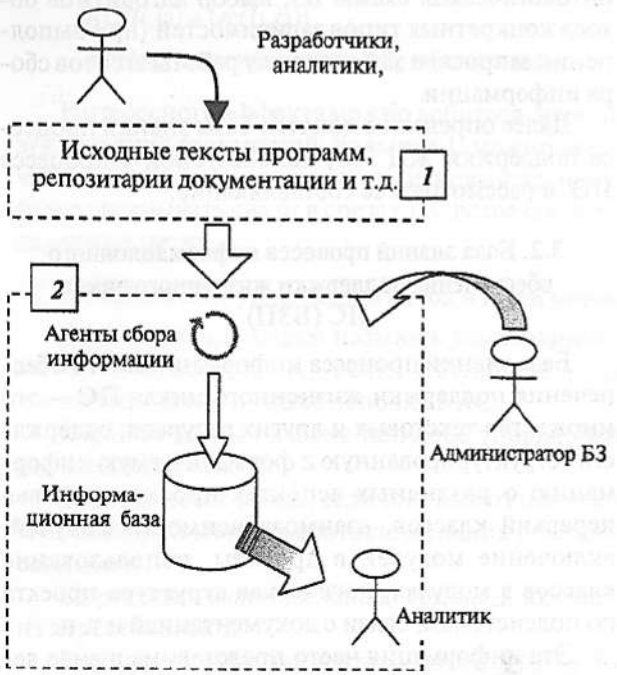


Рис. 1. Структура рассматриваемой части процесса ИО

Блок №2 включает в себя две основных деятельности, описанные выше: наполнение базы знаний и выполнение запросов к ней.

Под агентами сбора информации подразумеваются программные модули, автоматически и полуавтоматически извлекающие различные сведения непосредственно из исходных кодов программ либо путем мониторинга их во время выполнения. Эти сведения могут включать в себя: физическую структуру проектов, перечни классов/глобальных переменных/процедур и прочих языкозависимых элементов, использование определенных элементов одной среды (как процедур в БД) элементами другой среды (клиентским кодом приложения) и др. Кроме того, в эти сведения могут входить и логические структуры проектов по подсистемам и задачам, номера версий продукта, перечни сотрудников, связи сотрудников с подсистемами и др., которые могут быть введены вручную или импортированы из имеющихся источников.

Агенты сбора информации выполняют функцию наполнения и обновления БЗ, находящейся под управлением некоторого сервера. Аналитик, которому требуется выполнить запрос к этой БЗ (например получить перечень логических подсистем продукта, которые нужно протестировать после изменения определенного модуля), также обращается к серверу, используя некий формальный язык.

Настройки, которые влияют на способы формирования ответа на запрос, определяются Администратором БЗ, который отвечает за построение

онтологической схемы БЗ, выбор алгоритмов обхода конкретных типов зависимостей (при выполнении запроса) и за поддержку работы агентов сбора информации.

Далее определим понятие базы знаний процесса поддержки ЖЦ ПС, используемой в процессе ИО, и рассмотрим её составляющие.

3.2. База знаний процесса информационного обеспечения поддержки жизненного цикла ПС (БЗП)

База знаний процесса информационного обеспечения поддержки жизненного цикла ПС – это множество текстовых и других ресурсов, содержащих структурированную и формализуемую информацию о различных аспектах проекта: деревья иерархий классов, взаимозависимости модулей, включение модулей в проекты, использование классов в модулях, логическая структура проекта по подсистемам, связи с документацией и т. п.

Эта информация часто представима в виде деревьев, ориентированных и неориентированных графов и т. д. Можно понимать элементы БЗП как множество объектов $X = \{xi\}$ и множества признаков L, с помощью которых формируется представление этих объектов в ЭВМ. Каждый из признаков задает наличие у данного объекта некой характеристики. То есть он выступает в описываемой этим признаком *роли*. Чтобы представить БЗП в формальном виде, т. е. извлечь ее формальную структуру из разнородных источников, необходимо разработать онтологическую схему и построить средства извлечения информации из имеющихся источников под данную онтологию.

В то же время есть необходимость в том, чтобы при разработке подобной базы знаний уйти от непосредственного использования термина «объект», ограничившись лишь различными *описаниями* объектов. Эта необходимость обусловлена двумя причинами: 1) источники формализованной информации об отношениях элементов системы часто независимы друг от друга и, как следствие, не всегда возможно составить единый перечень объектов для решения всех задач ИО очередной ПС; 2) в данной работе рассматривается в первую очередь задача применения различных видов поиска для построения описаний некоторого термина или группы терминов в заданном контексте, а также построение семантически связанных групп элементов.

3.3. Формализация определения БЗП

БЗП определим формально как тройку (LBL, REL, CUT), где LBL – система меток, REL – множество некоторых топологически формализуемых отношений и CUT – множество так называемых срезов БЗ, реализующих эти отношения для данного процесса разработки.

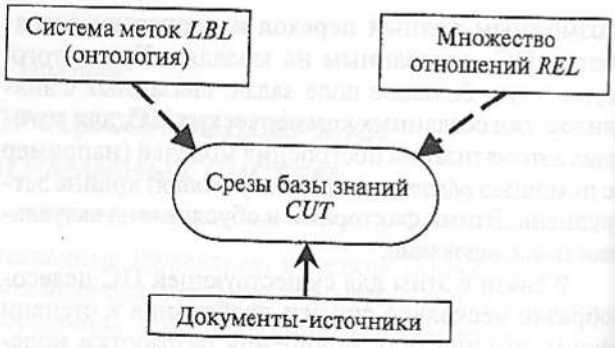


Рис. 2. Элементы базы знаний процесса ИО

Система меток, словарь отношений и срезы определяются спецификой конкретной ПС, доступными источниками информации. Множество отношений REL образует структуру БЗП. Это только те отношения, которые возможно полностью формализовать, причем автоматически или полуавтоматически. Последнее требование связано с тем, что эти отношения часто содержат очень большое количество элементов. Сами по себе эти отношения могут быть как специфическими для данного процесса ИО или конкретной ПС, так и общими для многих ПС.

Система меток в базе знаний процесса ИО выполняет онтологическую функцию. Множество срезов CUT представляет собой содержимое базы знаний данного процесса ИО, формализованной в терминах системы меток LBL через отношения REL. Наполнение срезов и поддержка их в актуальном состоянии происходит с помощью агентов сбора информации параллельно с модификацией самой базы знаний процесса программистами, документалистами и аналитиками (см. рис. 1).

3.4. Определение системы меток

Систему меток LBL определим следующим образом:

$LBL = (L, RG, \omega_{расш}, \omega_{суж})$,

L – множество меток,

$RG = (N, RR)$ – граф семантических связей,

где N – множество вершин,

RR – множество ребер,

$\omega_{расш}, \omega_{суж}$ – стоимости расширения и сужения контекста,

$RR = \{(l_1, l_2, \omega_1, \omega_2)\}$, l_1, l_2 – метки,

ω_1, ω_2 - стоимости перехода.

Граф связей системы меток имеет следующие свойства:

1) является семантической сетью, в которой имеются как однонаправленные, так и двунаправленные связи (это зависит от величин стоимостей перехода);

2) может быть многокомпонентным;
3) не имеет кратных ребер.

С каждой связью в графе системы меток сопоставим два числа $\omega_1, \omega_2 \in [0;1]$. Эти числа будем называть *стоимостями перехода* между метками данной связи. Если стоимость перехода в заданном направлении равна нулю, то будем говорить, что такой переход невозможен. Чем ближе стоимость перехода к 1, тем более эти метки связны в данном направлении перехода.

Связи в системе меток можно классифицировать по соотношению стоимостей прямого и обратного переходов:

— тождественная: $\omega_1 = \omega_2 = 1$;
— полутождественная: $\omega_1 \neq \omega_2$; ω_1 или $\omega_2 = 1$;
— синонимическая: $\omega_1 = \omega_2$; $0 < \omega_1 < 1; 0 < \omega_2 < 1$;
— общего вида: $\omega_1 \neq \omega_2$; $0 \le \omega_1 < 1; 0 \le \omega_2 < 1$.

3.5. Определение переходов и связности меток

Если в системе меток есть связь $rr(l_1, l_2, \omega_1, \omega_2)$, то прямым переходом по данной связи будем называть переход от метки l_1 к метке l_2, а *обратным*, соответственно, от l_2 к l_1.

Будем говорить, что связность перехода от произвольной метки l к этой же метке l (петля в графе) всегда равна 1 для любой системы меток.

В более общем случае переход от одной метки к другой — это поиск такой цепочки переходов между этими метками, связность которой будет больше нуля.

Определим *цепочку перехода* между l_1 и l_2 как *цепь* между l_1 и l_2 (т.е. *маршрут*, все ребра которого различны) в графе системы меток, имеющую ненулевую *связность*.

$$rc(l_1, l_2) = \left\{(rr_i, \omega_i^k)\right\},$$

ω_i^k — стоимость перехода по связи rr_i

$$k = \begin{cases} 1, \text{если это прямой переход} \\ 2, \text{если это обратный переход} \end{cases}$$

Связность σ цепочки перехода rc вычисляется по формуле

$$\sigma(rc) = \prod_i \omega_i, \text{ причем } \sigma(rc) > 0,$$

где ω_i — стоимость i-го перехода цепочки

Итак, возможность построения цепочки перехода между двумя метками определяется двумя факторами: принадлежностью этих меток одной компоненте графа связей и расстановкой весов связей в этом графе.

Поскольку цепочек перехода между двумя метками в данной системе меток может быть более одной, то определим *лучшую цепочку* перехода между двумя метками как одну из цепочек с наибольшей связностью во множестве всевозможных цепочек:

$$RC(l_1, l_2) = \left\{rc(l_1, l_2)\right\},$$

лучшая цепочка $rcm \in RC$, $\sigma(rcm) = \max$.

Интересного эффекта можно добиться, используя значения стоимостей, большие 1: можно *увеличивать* глубину, на которую поисковый алгоритм будет раскрывать связи в срезах БЗ, встречая определенные метки.

3.6. Определение унификации меток и групп меток

Две метки l_1, l_2 будем называть *унифицируемыми* со связностью σ, если $RC(l_1, l_2) \neq \varnothing$, где $\sigma = \sigma(rcm)$, rcm — лучшая цепочка в RC.

Группой меток назовем непустое упорядоченное подмножество L.

Группы меток *равны*, если они имеют одинаковый размер, и метки в соответствующих позициях совпадают.

Определим *интерпретацию группы* L как одну из перестановок L.

В общем случае две группы меток L_1 *и* L_2 будем называть унифицируемыми со связностью σ, если существует хотя бы одна интерпретация L_1^* и L_2^*, при которых $\sigma(L_1^*, L_2^*) > 0$. Но, поскольку таких интерпретаций может быть несколько, будем понимать унификацию L_1 *и* L_2 как поиск и выбор одной из интерпретаций, при которых связность максимальна.

Правило вычисления связности двух групп меток (или их перестановок) L_1 и L_2, имеющих одинаковую размерность, запишем как:

$$\sigma(L_1, L_2) = \prod_{i,j} \sigma(l_i, l_j), \text{ где } l_i \in L_1,\ l_j \in L_2,\ i = j,$$

ясно, что $\sigma(L_1, L_2) = 1$, если $L_1 = L_2$.

В общем случае $\sigma(L_1, L_2) \neq \sigma(L_2, L_1)$, т.е. операция вычисления связности некоммутативна.

Если L_1 *и* L_2 имеют различное количество меток, то говорим, что происходит расширение или сужение контекста. При этом из большей группы для вычисления $\sigma(L_1, L_2)$ необходимо выбрать такое подмножество меток, равное по размеру меньшей группе, чтобы одна из перестановок этого подмножества максимизировала $\sigma(L_1, L_2)$.

Будем говорить, что если размерность L_1 больше размерности L_2, то имеет место *сужение* контекста. А если количество меток в L_1 меньше, чем в L_2, то имеет место *расширение* контекста. С расширением и сужением контекста также связаны определенные стоимости, величины которых задаются константами $\omega_{\text{расш}}$, $\omega_{\text{суж}}$, общими для системы меток, используемой при унификации данных групп меток. Эти стоимости включаются в виде множителей в общее произведение в формуле вычисления связности $\sigma(L_1, L_2)$.

Таким образом, на величину связности двух групп меток влияют два фактора: стоимости переходов, сопоставленные со связями системы меток, и стоимости расширения/сужения контекста, указанные в определении системы меток.

Приведем выражение для подсчета связности групп меток с учетом различного количества их элементов:

$$\sigma(L_1, L_2) = \prod_{i,j} \sigma(l_i, l_j) * \omega^m,$$

где $l_i \in L_1$; $l_j \in L_2$; $i = j$; m – разница в количестве элементов;

$$\omega = \begin{cases} \omega_{\text{расш}}, \text{если количество элементов } L_2 \text{ больше } L_1 \\ \omega_{\text{суж}}, \text{если количество элементов } L_1 \text{ больше } L_2. \end{cases}$$

3.7. Определение срезов базы знаний

Структура БЗП определяется множеством отношений $R \in REL$, которые представлены в ней соответствующими *срезами*. Определим срез базы знаний процесса разработки как ориентированный граф, одно- или многокомпонентный, узлы которого связаны друг с другом отношением R. Срезы БЗ могут быть построены по топологически формализуемым отношениям либо автоматически, либо вручную. Примерами срезов БЗ могут служить: лес иерархий классов в исходном коде, дерево зависимостей объектов в базе данных, логическое разбиение проекта на подсистемы.

Определим срез формально (*cut*):

$$cut = (Name, R, N, L, M), cut \in CUT, R \in REL,$$

где $Name$ – имя среза;

R – отношение, по которому построен граф среза;

N – множество узлов (вершин графа среза);

L – множество связей (ребер графа среза), реализующих отношение R, между узлами из N; $L = \{l_i\}, l_i = (n_1, n_2)$.

Таким образом, срез представляет собой одно- или многокомпонентный ориентированный граф, связи которого отвечают отношению R. В базе знаний для каждого отношения может содержаться более одного среза. Объединение всех срезов (построенных по отношению R) эквивалентно самому отношению R.

Узлом среза будем называть вершину графа среза, с которой сопоставлена *сигнатура*, т. е. пара $(Name, M)$, где $Name$ – имя данного узла, M – группа меток, обязательно содержащая все элементы образующего множества меток данного среза. Узел среза не может быть однозначно идентифицирован своей сигнатурой. Имя узла несет смысловую нагрузку, непосредственно связанную с тем процессом разработки, в рамках которого строится БЗП.

3.8. Определение поиска по БЗП

По аналогии с [3] определим задачу *поиска* по базе знаний процесса разработки ПС как задачу построения структурированного множества узлов одного или более срезов БЗ, входными данными которой является сама БЗ и *опорное множество* сигнатур узлов $\{s_i\}$, где s_i – сигнатура узла.

Поиск заключается в пошаговом расширении протокола поиска (структурированного множества узлов и информации о ходе поиска) за счет выполнения двух видов *переходов*: а) экстенсивного, б) интенсивного.

Экстенсивным переходом $Ex(n, n_2)$ назовем выбор связи $l(n, n_2)$ в графе среза, которому принадлежит n, и добавление в рабочий набор узла n_2. Экстенсивный переход осуществляется за счет наличия в срезе базы знаний непосредственной связи между узлами. Эта связь отражает отношение R, которое присутствует в сигнатуре данного среза.

Интенсивным переходом $In(n, n_2)$ назовем поиск такого узла n_2, который унифицируется с узлом n, и добавление его в рабочий набор. Особенностью интенсивного перехода является то, что n_2 может принадлежать любому срезу, а не только тому, которому принадлежит n. Интенсивный переход осуществляется за счет наличия косвенной связи между группами меток через систему меток.

С каждым узлом, записываемым в протокол поиска, ассоциируется некоторая величина связности, которая должна быть больше или равной некоторому нижнему пределу в диапазоне (0; 1]. Это число позволит ограничить «чувствительность» поиска, то есть длину цепочек, выводимых от элементов опорного множества.

3.9. Алгоритм поиска

Входные данные: опорное множество сигнатур (групп меток) и нижний предел связности.

Выходные данные: многоуровневый протокол поиска.

Порядок работы:

1. Инициировать пустой протокол поиска.
2. Считать текущий уровень нулевым.
3. Построить базу поиска и добавить ее элементы в протокол на текущем уровне.
4. Увеличить номер текущего уровня.
5. В текущий уровень занести множество всех узлов, которые унифицируемы (со связностью, большей нижнего предела) хотя бы с одним узлом предыдущего уровня и еще не содержатся в протоколе поиска. Если узел может быть получен в результате перехода с более чем одного узла предыдущего уровня, величина связности каждого добавляемого узла выбирается наибольшей из всех возможных.
6. Если в текущем уровне есть хотя бы один узел, вернуться к шагу 4.

Иллюстрация возможного протокола поиска приведена на рис. 3.

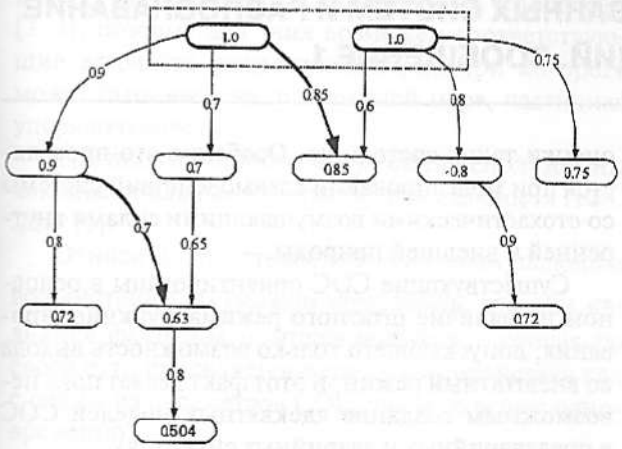


Рис. 3. Протокол поиска

На рис. 3 показаны узлы и переходы, выполненные при поиске. Числами указаны стоимости переходов и уровни связности цепочки в каждом узле-результате, пунктиром выделено опорное множество узлов. В показанном протоколе поиска имеются 4 уровня (горизонтальные группы узлов сверху вниз), от 0-го до 3-го. На нулевом уровне находится *база поиска* (строим по опорному множеству).

В начале работы алгоритма поиска протокол заполняется узлами, сигнатуры которых унифицируемы хотя бы с одной сигнатурой из опорного множества. Будем называть это первичное множество узлов *базой поиска*. Для каждого элемента базы поиска устанавливается величина связности, равная наибольшей из связностей сигнатуры этого узла с сигнатурами опорного множества. При расширении протокола поиска (путем интенсивных и экстенсивных переходов) новым добавляемым узлам будет сопоставлена величина связности, равная произведению связности исходного узла и стоимости соответствующего перехода.

При выполнении каждого перехода происходит понижение связности, вплоть до некоторого заданного минимума. Это позволит контролировать глубину поиска. С другой стороны, ширину поиска, то есть охват срезов БЗ, в принципе возможно контролировать путем установления нижней границы связности для каждого среза БЗ отдельно, например путем нормирования нижней границы по количеству узлов каждого среза.

3.10. Нижние пределы связности и унифицируемости

Унификация групп меток производится при поиске узлов для выполнения интенсивного перехода. В начале построения очередной ветви цепочки поиска от опорного множества алгоритм устанавливает текущий уровень связности в 1. Каждый интенсивный переход между узлами увеличивает длину цепочки поиска и уменьшает уровень связности путем домножения его на стоимость очередного перехода. Как только уровень связности падает ниже некоторой заданной константы, развитие данной цепочки прекращается. Эту константу назовем *нижним пределом связности цепочки*.

Кроме того, если связность групп меток узлов ниже некоторой константы, то интенсивный переход между ними считается невозможным. Эту константу назовем *нижним пределом унифицируемости групп меток*.

4. Заключение

В настоящее время появляется необходимость в максимально гибких средствах анализа информации, создаваемой в процессе поддержки ЖЦ программных систем, однако на пути создания таких гибких средств стоит проблема формализации источников и сложности строящихся моделей. Одним из возможных решений может стать предлагаемое снижение степени формализации информационной базы ЖЦ ПС и даже создание возможности управления степенью ее формализации за счет применения поисковых технологий.

Таким образом, в данной работе обоснован новый подход к интеллектуализации задач информационного обеспечения, предложены новые модели и алгоритм информационного поиска.

В настоящей работе кратко рассмотрен лишь один из возможных алгоритмов поиска по предлагаемой структуре данных. Направлением дальнейших исследований является разработка методов и алгоритмов автоматизации анализа исходных кодов с использованием базы знаний процесса ИО, основанной на системе меток и срезах. Необходимо также рассмотреть возможности «обучения» БЗП, оптимизации запросов.

Список литературы: 1. *Baezo-Yates R., Ribeiro-Neto B.* Modern Information Retrieval. ACM Press, Addison Wesley. 1999. 2. *C. J. van RIJSBERGEN.* INFORMATION RETRIEVAL, 2nd edition. 1979. 3. *Preece, Scott.* A spreading activation network model for information retrieval. PhD thesis, CS Dept., Univ. of Illinois, 1981. 4. *Cohen P., Kjeldsen R.* Information Retrieval by Constrained Spreading Activation on Semantic Networks // Information Processing and Management. 1987. 23(4) C. 255-268. 5. *Maciej Ceglowski, Aaron Coburn, John Cuadrado.* Semantic Search of Unstructured Data using Contextual Network Graphs // National Institute for Technology and Liberal Education, 2003.Vermont, 05753 USA. 6. *ГОСТ 24.205-80.* Требования к содержанию документов по информационному обеспечению.

Поступила в редколлегию 21.11.2004

УДК 681.518:004:912

В.Н. Бурцев, А.Л. Ерохин

ФОРМАЛИЗАЦИЯ СЛОЖНООРГАНИЗОВАННЫХ СИСТЕМ И РАСПОЗНАВАНИЕ АВАРИЙНЫХ СИТУАЦИЙ. СООБЩЕНИЕ 1

1. Введение

Обеспечение устойчивости функционирования сложноорганизованных систем (СОС) относится к многокритериальными динамическим задачам в многомерном фазовом пространстве. В качестве критерия устойчивости функционирования сложной системы может быть выбрана инвариантность структуры множества связей-отношений между параметрами ее входов и выходов.

Изменения связей-отношений, возникающие, например, при воздействиях внешних факторов стохастической природы, могут служить признаками возникновения нестандартной (внештатной) ситуации, которая в дальнейшем может развиться в аварийную.

Однако для использования в системном анализе структурной инвариантности необходима соответствующая формализация системы. Начало исследований в области такой формализации положено в [1, 8] .

В формальном виде, абстрагируясь от вида целевого продукта, можно выделить в СОС множество каналов-связей, организующих сетевую структуру, которые формируют множество входов-выходов, определяемых в системе по типам и структурам связей-отношений между подсистемами. Такие системы определим как канализированные [1, 2].

Цель исследования – разработка модели взаимодействий сложно-организованной системы со стохастическими возмущениями и создание на ее основе способа визуализации параметров режимов работы сложно-организованных канализированных систем при нештатных ситуациях

Переходя на формализованный уровень подсистем СОС, выделим системные каналы (СК) как множество элементарных каналов (ЭК), и тогда вход A и выход B в каждом СК представляется упорядоченными взаимно сопряженными множествами A_i и B_i.

В таком представлении эволюция системы в многомерном фазовом пространстве характеризуется множеством изменений в системе параметров на множестве входов-выходов A_i и B_i, которые определяются декартовым произведением подмножеств [1]

$$U \subset A \times B . \qquad (1)$$

При такой формализации основные сложности возникают при моделировании переходных и аварийных режимов СОС и разработке критериев оценки таких состоянию. Особенно это проявляется при моделировании взаимодействий системы со стохастическими возмущающими силами внутренней и внешней природы.

Существующие СОС ориентированы в основном на наличие штатного режима функционирования, допускающего только возможность выхода во внештатный режим, и этот факт делает пока невозможным создание адекватных моделей СОС в предаварийных и аварийных ситуациях.

Состояние СОС можно определить как множество Ф подмножеств флуктуаций многомерных векторов базовых параметров при воздействии множества Т стохастических возмущений:

$$\begin{gathered}\left(\overrightarrow{\varphi_I}\right),\left(\overrightarrow{\varsigma_J}\right),\left(\overrightarrow{\theta_L}\right),\left(\vec{P}_{FS}\right) \subset \Phi \\ \left(St_Q\right) \in T,\end{gathered} \qquad (2)$$

где $\left(\overrightarrow{\varphi_I}\right)$ – флуктуации технологических параметров со скалярными переменными $\Delta\varphi_I$; $\left(\overrightarrow{\varsigma_J}\right)$ – флуктуации параметров управления с переменными $\Delta\varsigma_J$; $\left(\overrightarrow{\theta_L}\right)$ – флуктуации параметров планирования с переменными $\Delta\theta_L$; $\left(\vec{P}_{FS}\right)$ – флуктуации психофизиологического состояния лица, принимающего решения (ПФС ЛПР); $\left(St_Q\right)$ – стохастические возмущающие воздействия.

Таким образом, задачами исследования являются построение модели отображения параметров режимов при воздействии множества T стохастических возмущений и разработка такого способа отображения множества параметров СОС, который позволил бы повысить эффективность распознавания, классификации и в конечном итоге, идентификации аварийных ситуаций (АС).

2. Решение задачи

Проведем анализ исходных данных на основе (1) и (2).

Параметры $\left[\left(\overrightarrow{\varphi_I}\right),\Delta\varphi_I\right],\left[\left(\overrightarrow{\varsigma_J}\right),\Delta\varsigma_J\right],\left[\left(\overrightarrow{\theta_L}\right),\Delta\theta_L\right]$ являются метрическими, нормируемыми и упорядоченными.

Пусть изменения флуктуаций этих параметров подчиняются нормальному закону распределения вероятности и относятся к детерминированным (эргодическим) процессам.

Относительно параметров ПФС ЛПР можно утверждать, что $\left(\vec{P}_{FS}\right)$ по своей природе относятся к детерминированным хаотическим процессам [3, 4], поэтому для них возможен соответствующий аттрактор существования, внутри которого может быть введена, по крайней мере, частичная упорядоченность.

Границы аттрактора могут быть использованы как доверительные интервалы оптимальности ПФС ЛПР [5].

Относительно стохастических возмущающих воздействий $\left(Si_Q\right)$ можно утверждать, что они являются фактором неопределенности, которая по своей сути проявляется везде, взаимодействуя как с одним из параметров СОС, так и со всеми одновременно.

Важность исследований взаимодействий метрических параметров СОС со стохастическими процессами определяется новыми подходами в осмыслении вклада хаотических процессов в сложных системах [6, 7].

Решение задачи разобьем на несколько этапов.

В данной статье рассмотрен первый этап решения задачи, посвященный разработке математической модели.

3. Этап 1. Разработка математической модели

В [1, 8] была сделана теоретическая попытка представления метаустойчивости СОС посредством описания формальной модели многомерного единичного шара, образованного векторами флуктуаций параметров (многомерной флуктуационной капсулы).

Используем этот подход для анализа внештатных ситуаций в СОС. Для этого построим единичный многомерный многогранник V, образованный объединением подмножеств флуктуаций параметров СОС

$$\left(\overrightarrow{\varphi_I}\right)\in N,\left(\overrightarrow{\varsigma_J}\right)\in N,\left(\overrightarrow{\theta_L}\right)\in K,\left(\vec{P}_{FS}\right)\in R \qquad (3)$$

с началом координат, помещенным в центр пересечения их диагоналей.

Так как параметры управления посредством прямых и обратных связей взаимодействуют с технологическими параметрами, то их отнесем к одному множеству ($\theta_i \leftrightarrow \varphi_j$ N).

Для параметров планирования и ПФС ЛПР также существует множество параметров корректировок μ_M, ν_N, которые обеспечивают оптимальные значения этих параметров.

Встроим многомерные многогранники параметров планирования и ПФС ЛПР внутрь многогранника технологических параметров, совместив начало координат. Тогда многомерный многогранник будет иметь вид:

$$V \subseteq \bigcup_{\varphi,\theta,\varsigma,P_{KS}}\left[\sum_{i=1}^{n}\left(\overrightarrow{\varphi_I}\right)\vee\sum_{i=1}^{m}\left(\vec{\varsigma}_j\right)\vee\sum_{k=1}^{r}\left(\overrightarrow{\theta_L}\right)\vee\right.$$
$$\left.\vee\sum_{M=1}^{R}\left(\mu_M\right)\vee\sum_{KS=1}^{s}\left(\overrightarrow{P_{KS}}\right)\vee\sum_{N=1}^{S}\left(\nu_N\right)\right]. \qquad (4)$$

Максимальные нормированные подмножества θ_i, φ, ζ, t, P являются вершинами многомерных многогранников $\Phi(\Theta)$, (Ξ), (T) и (P), вложенных друг в друга.

Оболочка, натянутая на указанные вершины, образует флуктуационную капсулу U, при этом считаем, что сама оболочка капсулы обладает некоторыми энергетическими параметрами, способными вернуть капсулу в исходное метастабильное состояние.

Такими энергетическими параметрами будем считать множество скалярных значений параметров управления $(\varphi_I \pm \Delta\varsigma_j) \in N$ технологических параметров и множество значений параметров корректировок $\left(\pm\Delta\mu_M\right)$ и $\left(\pm\Delta\nu_N\right)$, аналогично воздействующих на параметры планирования $(\theta_L \pm \Delta\mu_M) \in K$ и ПФС ЛПР $(P_{KS} \pm \Delta\nu_N) \in M$.

Управление технологическими параметрами является динамическим процессом (сильное взаимодействие), корректировка параметров планирования и ПФС ЛПР являются более инерционными (слабое взаимодействие).

Зададим, например, подмножество векторов флуктуаций технологических параметров

$$\overrightarrow{\phi_i} = f\left(\vec{\varphi}_i, \vec{\theta}_j, \overrightarrow{P_{KS}}\right);\quad \phi_t, \theta_j, P_{KS} \le 1;\quad t_S = 0, \qquad (5)$$

где a_i, b_i, c_i — угловые коэффициенты векторов, упорядоченные по меридианам и широтам единичной сферы; $\phi_t, \theta_j, P_{KS} \le 1$ — нормированные скалярные значения векторов; $t_S = 0$ — стохастические воздействия на СОС отсутствуют.

Так как основными возмущающими воздействиями на СОС являются параметры стохастических воздействий и параметры ПФС ЛПР, то для дальнейшего рассмотрения поставленной задачи введем два предположения:

— результирующее взаимодействие технологических параметров с параметрами управления и параметров планирования с параметрами корректировок не влияет на положение векторов $\left(\vec{\varphi}_I\right), \left(\overrightarrow{\theta_L}\right)$, то есть не изменяет координат соответствующих вершин на поверхности капсулы;

— результирующее взаимодействие метрических параметров со стохастическими возмущающими силами и ПФС ЛПР приводит к изменению положение векторов $\left(\vec{\varphi}_I\right)$, $\left(\overrightarrow{\theta_L}\right)$.

Редуцируем проекции вершин многогранников многомерного шара U^* на двух гиперплоскостях

$$b_1x_1 + b_2x_2 + \ldots + b_nx_n + \ldots + b_{v-1}x_{v-1} \le 0$$
$$c_1x_1 + c_2x_2 + \ldots + c_nx_n + \ldots + c_{v-1}x_{v-1} \le 0. \qquad (6)$$

Первая из плоскостей (рис. 1, поз. 1) является касательной к одной из вершин многогранника, например технологического параметра. Гиперплоскость 2 располагается перпендикулярно первой.

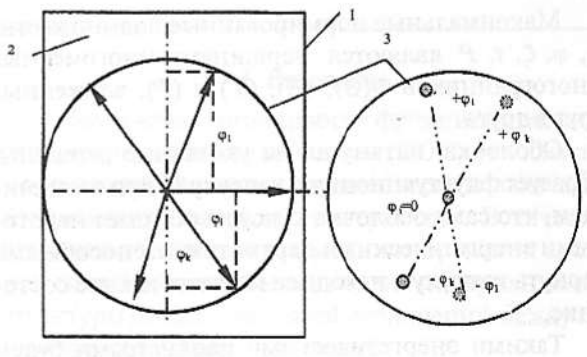


Рис. 1. Редукция проекции вершин многогранников многомерного шара параметров СОС

Выберем из множества $N_i^R \in N^R$ три технологических параметра

$$\varphi_1, \varphi_i, \varphi_k \in N_j^* \subset N^R \qquad (7)$$

и зададим положение параметров в трехмерном метрическом пространстве в виде трех линейных зависимостей

$$\begin{aligned} a_1 x + b_1 y + c_1 z &= 0 \\ a_i x + b_i y + c_i z &= 0 \\ a_k x + b_k y + c_k z &= 0, \end{aligned} \qquad (8)$$

где коэффициенты a, b, c задают угловые характеристики векторов флуктуаций этих параметров.

Выберем такое положение плоскости 1, чтобы она касалась ϕ_1 вершины, представленной линейной зависимостью $a_1 x + b_1 y + c_1 z = 0$. В этом случае все возможные флуктуации параметров φ_i, φ_k проецируются на поверхность касательной плоскости в виде открытых линейных интервалов

$$(-\varphi_1 + \varphi_1), (-\varphi_k + \varphi_k) \le 1, \qquad (9)$$

как показано на рис. 1.

Приняв первое предположение о неизменности положения векторов на поверхности единичного шара при взаимодействии технологических параметров, считаем, что координаты вершин соответствующего многомерного многогранника являются граничными значениями изменений скалярных переменных флуктуаций.

В предположении сильного или слабого взаимодействия параметров со стохастическими и ПФС ЛПР воздействиями вектор может изменить свое положение, и интервал может быть задан на плоскости в виде двумерного множества.

Примером взаимодействия является предаварийная ситуация в распределении электрических потоков между потребителями по аварийным изменениям активных и реактивных мощностей [9].

Аварийное отключение индуктивной мощности W_{IN} неизбежно ведет к оперативному перераспределению высвободившейся мощности между другими потребителями или подключению к энергосистеме «поля с реактивными нагрузками» или другие решения из-за перекоса векторов мощностей в системе.

Спроецируем на касательной плоскости 1 распределение флуктуаций реактивной индуктивной мощности W_{IN} и реактивной емкостной мощности W_{CA} при наличии стохастического возмущения T_S.

Линейный интервал нормированных флуктуаций реактивных мощностей

$$(-W_{IN} + W_{IN}), (-W_{CA} + W_{CA}) \le 1 \qquad (10)$$

может быть представлен двумерным множеством Φ^2, и тогда внештатная ситуация характеризуется изменениями флуктуаций технологических параметров

$$\left(\overrightarrow{\phi_I} \pm \Delta\varphi_I\right) \in \Phi^2. \qquad (11)$$

Условиями выхода СОС во внештатный режим флуктуаций технологического параметра будут являться выход скалярной переменной $\pm\Delta\varphi_I$ за единичный интервал (9, 10) и выход вектора $\overrightarrow{\phi_I}$ за максимальное значение замыкания множества Φ^2 (11).

Рассмотрим пример, представленный на рис. 2.

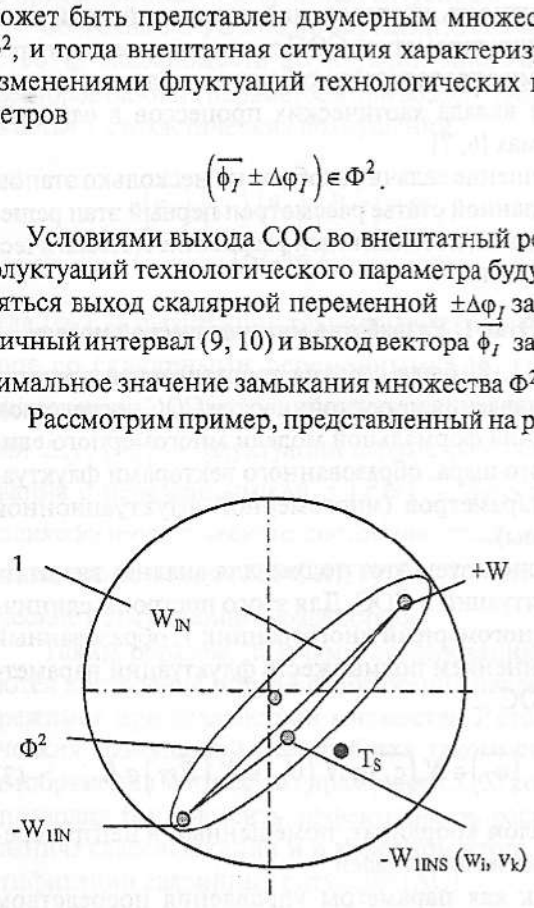


Рис. 2. Пример флуктуационной капсулы

Введем стохастическое возмущение T_S. В предположении взаимодействия параметров флуктуации скалярной переменной индуктивной мощности W_{1IN} считаем, что вектор $\overrightarrow{W_{IN}}$ изменяет свои первоначальные координаты на единичной сфере и установится в новом положении, например, — $W_{1INS}(w_d, v_k)$.

За счет преобразования одномерных интервалов (10,11) в двумерное открытое подмножество

$$\varphi_i \in \left(\Phi_n^2\right) \subset \Phi \qquad (12)$$

получаем двумерное распределение всех флуктуаций технологических параметров $ц_i$., измененных под воздействием стохастического возмущения T_S.

В качестве двумерного множества $\left(\Phi_n^2\right)$ выберем эллипс, у которого большой диаметр $D_{\max}$ соответствует интервалу (10), а малый $D_{\min}$ – определяет флуктуации векторов технологических параметров при взаимодействии их с возмущающими силами T_S.

Таким образом, значения изменений векторов $\overrightarrow{\Delta\varphi_I} > \Delta_K$, распределенных относительно малого радиуса эллипса $D_{\min} \subset \left(\Phi_n^2\right)$, дают возможность использовать для классификации внештатных ситуаций в СОС аппарат нечеткой логики [11, 12], а совместно с анализом изменений амплитуд флуктуаций $\Delta\varphi_i^{R1} > 1$ по большому диаметру $D_{\max} \subset \left(\Phi_n^2\right)$ – для классификации аварийных ситуаций.

Определим в дальнейшем множество (12) как прецессию параметра φ_i на поверхности единичного многомерного шара.

Рассмотрим вариант, когда изменение флуктуации параметра $\overrightarrow{\varphi_i}, \varphi_i \in \left(\Phi^2\right)$ относительно малого диаметра эллипса составляет

$$\overrightarrow{\varphi}_i^{R2} \le \Delta_K, \qquad (13)$$

где Δ_K — максимально возможное его изменение при стохастическом воздействии, не приводящее к аварийной ситуации.

Изменение флуктуации скалярной переменной указанного параметра относительно большого диаметра — $\Delta\varphi_i^{R1} > 1$.

Считаем, что за счет параметров управления не происходит выхода СОС из штатного режима.

Тогда в двумерной области $\left(\Phi_n^2\right)$ параметры

$$\Delta_{MAX} \ge \overrightarrow{\varphi}_i^{R1} > \Delta_K \text{ и } \Delta_{MAX} \ge \Delta\varphi_i^{R1} > \Delta_K \qquad (14)$$

могут рассматриваться как критерии внештатных и предаварийных ситуаций в СОС.

4. Выводы

1. Разработан этап 1 решения задачи создания формальной модели метаустойчивости СОС, представленной в виде многомерного единичного шара – флуктуационной капсулы, на поверхности которой определенным образом распределены вершины многомерных многогранников векторов флуктуаций, находящихся в поле воздействий стохастических нагрузок и ПФС ЛПР.

2. Предложена модель взаимодействия метрических и нормированных параметров СОС со стохастическими возмущениями. Результатом таких взаимодействий является смещение векторов метризуемых и нормируемых параметров внутри двумерного открытого множества $\left(\Phi_n^2\right)$ гомеоморфного кругу, замыкание которого может быть принято за верхнюю границу существования внештатной ситуации.

3. Анализ проекций векторов параметров СОС на двумерном множестве дает теоретическую возможность разработать критерии классификации внештатных и аварийных ситуаций в СОС, например, посредством математического аппарата нечеткой логики.

4. Объектом теоретических и экспериментальных исследований классификации внештатных и аварийных ситуаций в СОС является определение множества связей–отношений между метрическими и нормированными параметрами со стохастическими возмущениями и влиянием ПФС ЛПР.

5. В следующей статье будет рассмотрен второй этап решения задачи построения модели отображения параметров режимов сложно-организованной системы при воздействии множества стохастических воздействий, посвящены построению и исследованию модели визуализации параметров.

Список литературы: 1. *Бондаренко М.В., Єрохін А.Л.* Про моделі позаштатної поведінки інтелектуальних систем // Проблеми біоніки. 2004. Вип. 60. С. 7-16. 2. *Бурцев Вал.Н., Бурцев Вл.Н., Ерохин А.Л.* Анализ связей сложно-организованных систем с гомеостатическим и гетеростатическим управлением // Системный анализ, управление и информационные технологии. Вестник НТУ "ХПИ", 2001. Сб. науч. тр. Тематич. вып.: Автоматика и приборостроение. Харьков: НТУ "ХПИ". 2001. № 4. С. 20-23. 3. *Пригожин И.* Природа, наука и новая рациональность (В поисках нового миропонимания). М.,1991. 4. *Пригожин И., Стенгерс И.* Порядок из хаоса. М.: Прогресс, 1986. 432 с. 5. *Бурцев Вал.Н., Бурцев Вл.Н., Ерохин А.Л.* Обеспечение устойчивости системы психофизиологического состояния ЛПР в системах поддержки принятия решений // Проблемы бионики: Всеукр. межвед. науч.-техн. сб. Х.: Изд-во ХНУРЭ. 2002. Вып. 57. С. 91-94. 6. *Рузавин Г.Н.* Синергетика и диалектическая концепция развития // Философские науки, № 5. 1978. 7. *Князева Е.Н., Курдюмов С.П.* Синергетика как новое мировоззрение. Диалог с И. Пригожиным // Вопросы философии, №12, 1992. 8. *Бурцев В.Н., Ерохин А.Л.* Об оптимальной формализации сложноорганизованных систем / Труды III Междунар. конф. «Идентификация систем и задачи управления», Москва, 28-30 января 2004 года, ИПУ РАН. С. 440-444. 9. *Гриб О.Г., Ерохин А.Л., Светелик А.А.* О поддержке принятия решений при аварийных ситуациях в электрических сетях // Проблемы бионики: Всеукр. межвед. науч.-техн. сб. Х.: Изд-во ХНУРЭ. 2000. Вып. 53. С. 28-30. 10. *Єрохін А.Л.* Топології просторових інформаційних моделей електричних мереж // Електроенергетичні та електромеханічні системи. Вісник Нац. ун. «Львівська політехніка». № 479. Львів. 2003. С. 72-79. 11. *Заде Л.* Понятие лингвистической переменной и его применение к принятию приближенных решений: Пер. с фр. М.: Мир, 1976. 165 с. 12 *Кофман А.* Введение в теорию нечетких множеств: Пер. с фр. М.: Радио и связь, 1982. 432 с.

Поступила в редколлекию 23.11.2004

УДК 681.513

Е.В. Горшков, В.В. Колодяжный, И.П. Плисс

АДАПТИВНЫЙ АЛГОРИТМ ОБУЧЕНИЯ НЕО-ФАЗЗИ НЕЙРОНА И НЕЙРО-ФАЗЗИ СЕТИ НА ОСНОВЕ ПОЛИНОМИАЛЬНЫХ ФУНКЦИЙ ПРИНАДЛЕЖНОСТИ

1. Введение

Последние годы для решения широкого класса задач обработки данных и управления в условиях структурной и параметрической неопределенности все чаще применяются искусственные нейронные сети и системы нечеткого вывода. На стыке этих двух направлений возникли нейро-фаззи сети, обладающие способностью к обучению, подобно нейросетям, и обеспечивающие интерпретируемость результатов, подобно нечетким системам. Именно это направление вычислительного интеллекта сейчас находится на стадии бурного развития. Достаточно близко к нейро-фаззи сетям примыкает гораздо менее известная конструкция, называемая нео-фаззи нейрон [1–3], схема которого приведена на рис. 1.

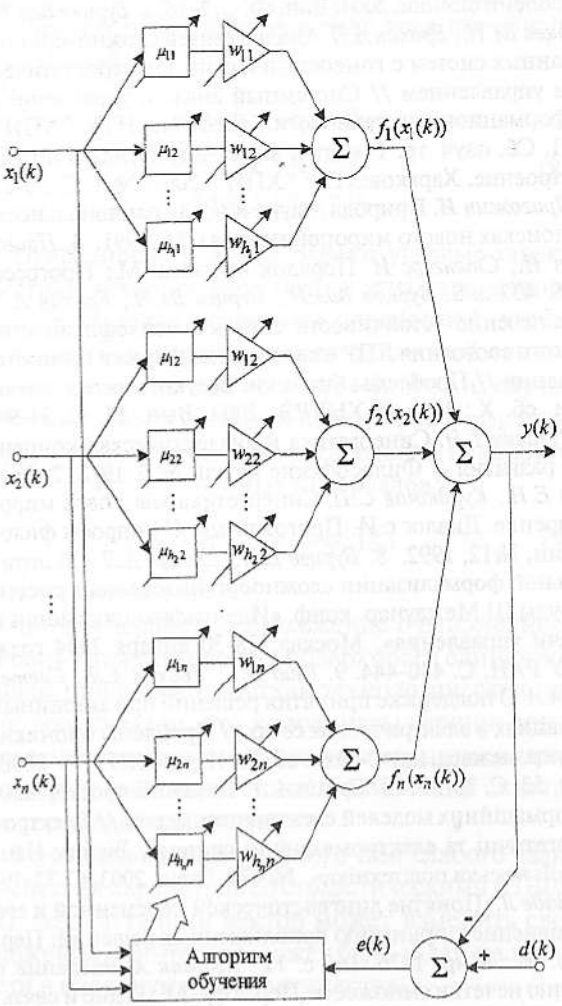


Рис. 1. Нео-фаззи нейрон

При подаче на вход нео-фаззи нейрона векторного сигнала-образа $x(k)=(x_1(k),x_2(k),\ldots,x_n(k))^T$ (здесь $k=1,2,\ldots$ – номер наблюдения в обучающей выборке или текущее дискретное время) на его выходе появляется скалярный сигнал

$$y(k)=\sum_{i=1}^{n} f_i(x_i(k))=\sum_{i=1}^{n}\sum_{j=1}^{h_i} w_{ji}(k)\mu_{ji}(x_i(k)),$$

определяемый как значениями используемых функций принадлежности $\mu_{ji}(x_i(k)), j=1,2,\ldots,h_i$; $i=1,2,\ldots,n$, так и настраиваемыми синоптическими весами $w_{ji}(k)$.

В [4] были предложены обобщение нео-фаззи нейрона на случай векторного выходного сигнала $y(k)$ и градиентный алгоритм обучения синаптических весов $w_{ji}(k)$ с постоянным параметром шага, причем во всех случаях использовалась фиксированная треугольная функция принадлежности

$$\mu_{ji}^{L}(x_i(k))=\begin{cases}1-d_{ji}^{-1}\,|c_{ji}-x_i(k)|, \text{ если } |c_{ji}-x_i(k)|\le d_{ji},\\ 0, \text{ в противном случае}\end{cases}$$

или, что то же самое,

$$\mu_{ji}^{L}(x_i(k))=(1-d_{ji}^{-1}\,|c_{ji}-x_i(k)|)\delta_{ji},$$

где c_{ji}, d_{ji} – параметры центра и ширины функции принадлежности μ_{ji} соответственно;

$$\delta_{ji}=\begin{cases}1, \text{ если } |c_{ji}-x_i(k)|\le d_{ji},\\ 0, \text{ в противном случае.}\end{cases}$$

При этом в качестве критерия обучения использовалась стандартная квадратичная форма

$$E(k)=\frac{1}{2}(d(k)-y(k))^2=\frac{1}{2}e^2(k)=$$
$$=\frac{1}{2}\left(d(k)-\sum_{i=1}^{n}\sum_{j=1}^{h_i} w_{ji}(k)\mu_{ji}(x_i(k))\right)^2, \quad (1)$$

где $d(k)$ – обучающий сигнал.

В [5] для повышения скорости обучения был введен оптимальный по быстродействию одношаговый градиентный алгоритм, а для улучшения аппроксимирующих свойств было предложено использовать квадратичные функции принадлежности

$$\mu_{ji}^{Q}(x_i(k))=(1-d_{ji}^{-2}(c_{ji}-x_i(k))^2)\delta_{ji} \quad (2$$

с настраиваемыми параметрами центра и ширины

Проведенный анализ показал, что существенное увеличение скорости сходимости не сопровождалось значимым улучшением аппроксимирующих свойств, что, как выяснилось, связано с тем, что производные квадратичной функции принадлежности терпят разрыв в точке пересечения с осью абсцисс ($\frac{\partial \mu_{ji}^{Q}(x_i)}{\partial x_i} \to \pm d_{ji}^{-1}$ при $x_i \to c_{ji} \pm d_{ji}$). Именно это свойство побудило предпринять настоящее исследование.

2. Обучаемый нео-фаззи нейрон с полиномиальными функциями принадлежности

Введем в рассмотрение функции принадлежности в виде полинома четвертой степени

$$\mu_{ji}^{F}(x_i(k)) = (1 - d_{ji}^{-2}(c_{ji} - x_i(k))^2)^2 \delta_{ji}, \qquad (3)$$

форма которых приведена на рис. 2.

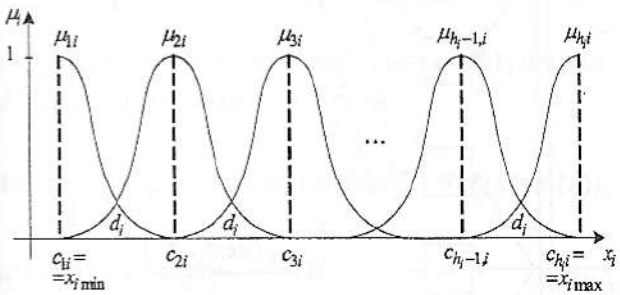


Рис. 2. Полиномиальные функции принадлежности

Как видно, это колоколообразные функции, производные которых не претерпевают разрыва на всей области определения.

Вводя в рассмотрение производные критерия (1) по настраиваемым параметрам с учетом (2) и (3)

$$\begin{cases} \dfrac{\partial E(k)}{\partial w_{ji}} = -e(k)\mu_{ji}^{F}(x_i(x)), \\ \dfrac{\partial E(k)}{\partial c_{ji}} = 4e(k)w_{ji}\mu_{ji}^{Q}(x_i(x))d_{ji}^{-2}(c_{ji} - x_i(k)), \\ \dfrac{\partial E(k)}{\partial d_{ji}^{-2}} = 2e(k)w_{ji}\mu_{ji}^{Q}(x_i(k))(c_{ji} - x_i(k))^2, \end{cases}$$

можно записать градиентный алгоритм обучения нео-фаззи нейрона с функциями принадлежности четвертой степени

$$\begin{cases} w_{ji}(k+1) = w_{ji}(k) + \eta^{w}(k)e(k)\mu_{ji}^{F}(x_i(k)), \\ c_{ji}(k+1) = c_{ji}(k) - \eta^{c}(k)e(k)w_{ji}(k)\mu_{ji}^{Q}(x_i(x)) \cdot \\ \quad \cdot d_{ji}^{-2}(k)(c_{ji}(k) - x_i(k)), \\ d_{ji}^{-2}(k+1) = d_{ji}^{-2}(k) - \eta^{d}(k)e(k)w_{ji}(k)\mu_{ji}^{Q}(x_i(k)) \cdot \\ \quad \cdot (c_{ji}(k) - x_i(k))^2, \end{cases} \qquad (4)$$

где $\eta^{w}(k), \eta^{c}(k), \eta^{d}(k)$ — переменные параметры шага обучения, подлежащие определению.

Записывая далее $(h_i \times 1)$-векторы параметров i-го нелинейного сигнала нео-фаззи нейрона

$$\mu_i^{F}(x(k)) = (\mu_{1i}^{F}(x_i(k)), \mu_{2i}^{F}(x_i(k)), \ldots, \mu_{h_i i}^{F}(x_i(k)))^T,$$

$$\mu_i^{Q}(x(k)) = (\mu_{1i}^{Q}(x_i(k)), \mu_{2i}^{Q}(x_i(k)), \ldots, \mu_{h_i i}^{Q}(x_i(k)))^T,$$

$$c_i(k) = (c_{1i}(k), c_{2i}(k), \ldots, c_{h_i i}(k))^T,$$

$$d_i^{-}(k) = (d_{1i}^{-2}(k), d_{2i}^{-2}(k), \ldots, d_{h_i i}^{-2}(k))^T,$$

$$w_i(k) = (w_{1i}(k), w_{2i}(k), \ldots, w_{h_i i}(k))^T$$

и $(h \times 1)$-векторы параметров нео-фаззи нейрона $(h = \sum_{i=1}^{n} h_i)$

$$\mu^{F}(x(k)) = (\mu_1^{FT}(x(k)), \mu_2^{FT}(x(k)), \ldots, \mu_n^{FT}(x(k)))^T,$$

$$\mu^{Q}(x(k)) = (\mu_1^{QT}(x(k)), \mu_2^{QT}(x(k)), \ldots, \mu_n^{QT}(x(k)))^T,$$

$$c(k) = (c_1^{T}(k), c_2^{T}(k), \ldots, c_n^{T}(k))^T,$$

$$d^{-}(k) = (d_1^{-T}(k), d_2^{-T}(k), \ldots, d_n^{-T}(k))^T,$$

$$w(k) = (w_1^{T}(k), w_2^{T}(k), \ldots, w_n^{T}(k))^T,$$

$$X(k) = (x_1(k)E_1^{T}, x_2(k)E_2^{T}, \ldots, x_n(k)E_n^{T})^T$$

(здесь E_i — $(h_i \times 1)$-вектор, состоящий из единиц), можно ввести векторную форму алгоритма (4) в виде

$$\begin{cases} w(k+1) = w(k) + \eta^{w}(k)e(k)\mu^{F}(x(k)), \\ c(k+1) = c(k) - \eta^{c}(k)e(k)w(k) \odot \mu^{Q}(x(k)) \odot \\ \quad \odot d^{-}(k) \odot (c(k) - X(k)), \\ d^{-}(k+1) = d^{-}(k) - \eta^{d}(k)e(k)w(k) \odot \mu^{Q}(x(k)) \odot \\ \quad \odot (c(k) - X(k)) \odot (c(k) - X(k)), \end{cases}$$

где $\odot$ — символ скоттова произведения.

Для увеличения скорости сходимости процесса обучения следует перейти от градиентных процедур со скалярным параметром шага $\eta(k)$ к процедурам, использующим матричный коэффициент усиления, наиболее популярным из которых является алгоритм Левенберга-Марквардта.

С этой целью введем $(h \times 1)$-векторы $\mu^{c}(x(k)) = 4w(k) \odot \mu^{Q}(x(k)) \odot d^{-}(k) \odot (c(k) - X(k))$ и $\mu^{d}(x(k)) = 2w(k) \odot \mu^{Q}(x(k)) \odot (c(k) - X(k)) \odot (c(k) - X(k))$, после чего можно записать

$$\begin{cases} w(k+1) = w(k) + e(k)(\mu^{F}(x(k))\mu^{FT}(x(k)) + \alpha^{w}I)^{-1} \cdot \\ \quad \cdot \mu^{F}(x(k)), \\ c(k+1) = c(k) - e(k)(\mu^{c}(x(k))\mu^{cT}(x(k)) + \alpha^{c}I)^{-1} \cdot \\ \quad \cdot \mu^{c}(x(k)), \\ d^{-}(k+1) = d^{-}(k) - e(k)(\mu^{d}(x(k))\mu^{dT}(x(k)) + \alpha^{d}I)^{-1} \cdot \\ \quad \cdot \mu^{d}(x(k)), \end{cases}$$

где $\alpha^w, \alpha^c, \alpha^d$ — неотрицательные регуляризирующие параметры; I — $(h \times h)$ -единичная матрица.

Используя далее технику преобразований, описанную в [6, 7], можно ввести модификацию алгоритма (5), обладающую как следящими, так и сглаживающими свойствами:

$$\begin{cases} w(k+1) = w(k) + \dfrac{e(k)\mu^F(x(k))}{\alpha^w(k)}, \\ \alpha^w(k+1) = \alpha\alpha^w(k) + \|\mu^F(x(k))\|^2, \\ c(k+1) = c(k) - \dfrac{e(k)\mu^c(x(k))}{\alpha^c(k)}, \\ \alpha^c(k+1) = \alpha\alpha^c(k) + \|\mu^c(x(k))\|^2, \\ d^-(k+1) = d^-(k) - \dfrac{e(k)\mu^d(x(k))}{\alpha^d(k)}, \\ \alpha^d(k+1) = \alpha\alpha^d(k) + \|\mu^d(x(k))\|^2, \\ 0 \le \alpha \le 1. \end{cases} \quad (6)$$

Как видно, алгоритм (6), при $\alpha = 0$ совпадающий с процедурой Левенберга-Марквардта, не использует операцию обращения матриц, что резко упрощает его численную реализацию.

Можно отметить, что улучшение обобщающих свойств нео-фаззи нейрона в ряде случаев может быть обеспечено применением несимметричных функций принадлежности, у которых параметр ширины задается соотношениями

$$\begin{cases} d_{ji}(k) = d_{ji}^L(k) + d_{ji}^R(k), \\ d_{ji}^L(k) = c_{ji}(k) - c_{j-1,i}(k), \\ d_{ji}^R(k) = c_{j+1,i}(k) - c_{ji}(k). \end{cases}$$

При этом автоматически отпадает необходимость в использовании третьего рекуррентного выражения в (6), а кроме того обеспечивается защита от возникновения нежелательных «мертвых зон» между соседними функциями принадлежности.

3. Нейро-фаззи сеть с полиномиальными функциями принадлежности

Достаточна близка по архитектуре к нео-фаззи нейрону нейро-фаззи сеть Ванга-Менделя [8], схема которой без выходного слоя нормализации приведена на рис. 3. Следует заметить, что если число настраиваемых синаптических весов нео-фаззи нейрона совпадает с размерностью входного пространства n, то нейро-фаззи сеть имеет h_i весов по числу функций принадлежности в антецеденте i -го входа и реализует отображение

$$y(k) = \sum_{j=1}^{h_i} w_j(k) f_j(x(k)) = \sum_{j=1}^{h_i} w_j(k) \prod_{i=1}^{n} \mu_{ji}(x_i(k)).$$

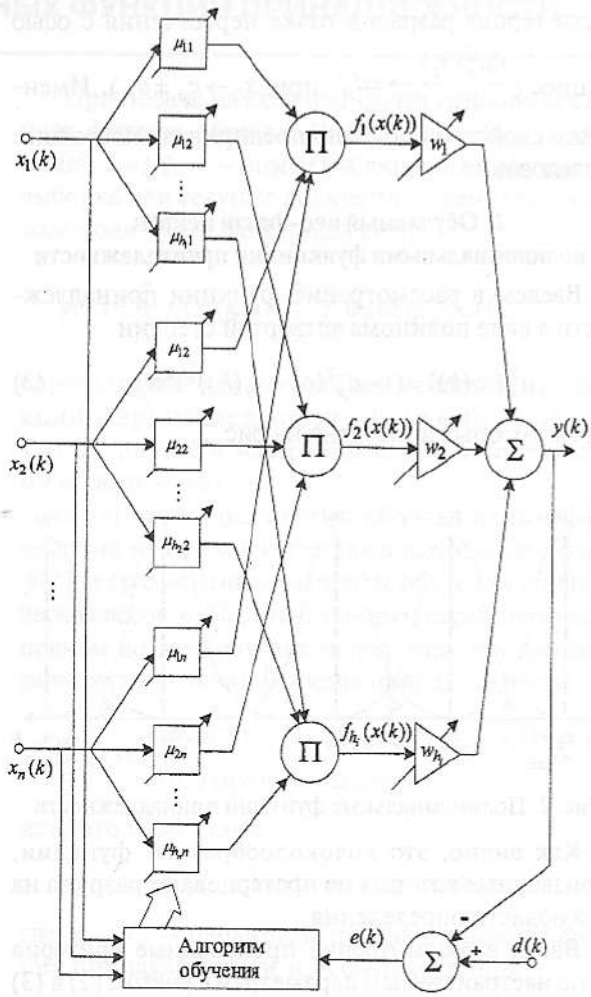


Рис. 3. Нейро-фаззи сеть

Вводя производные квадратичного критерия обучения по настраиваемым параметрам с учетом (3)

$$\begin{cases} \dfrac{\partial E(k)}{\partial w_j} = -e(k)\prod_{i=1}^{n}\mu_{ji}^F(x_i(k)), \\ \dfrac{\partial E(k)}{\partial c_{ji}} = 4e(k)w_j\left(\prod_{\substack{p=1 \\ p\neq j}}^{n}\mu_{pi}^F(x_i(k))\right)\mu_{ji}^Q(x_i(k))d_{ji}^{-2} \cdot \\ \qquad \cdot(c_{ji} - x_i(k)), \\ \dfrac{\partial E(k)}{\partial d_{ji}^{-2}} = 2e(k)w_j\left(\prod_{\substack{p=1 \\ p\neq j}}^{n}\mu_{pi}^F(x_i(k))\right)\mu_{ji}^Q(x_i(k)) \cdot \\ \qquad \cdot(c_{ji} - x_i(k))^2, \end{cases}$$

можно записать градиентный алгоритм обучения вида

$$\begin{cases} w_j(k+1) = w_j(k) + \eta^w(k)e(k)\prod_{i=1}^{n}\mu_{ji}^F(x_i(k)), \\ c_{ji}(k+1) = c_{ji}(k) - \eta^c(k)e(k)w_j(k)\left(\prod_{\substack{p=1 \\ p\neq j}}^{n}\mu_{pi}^F(x_i(k))\right) \\ \quad \cdot\mu_{ji}^Q(x_i(k))d_{ji}^{-2}(k)(c_{ji}(k)-x_i(k)), \\ d_{ji}^{-2}(k+1)=d_{ji}^{-2}(k)-\eta^d(k)e(k)w_j(k)\left(\prod_{\substack{p=1 \\ p\neq j}}^{n}\mu_{pi}^F(x_i(k))\right) \\ \quad \cdot\mu_{ji}^Q(x_i(k))(c_{ji}-x_i(k))^2. \end{cases} \quad (7)$$

Записывая далее $(h_i \times 1)$ -векторы

$$\mu^F(x(k)) = \left(\prod_{i=1}^{n}\mu_{1i}^F(x_i(k)), \prod_{i=1}^{n}\mu_{2i}^F(x_i(k)), \ldots, \prod_{i=1}^{n}\mu_{h_i i}^F(x_i(k))\right)^T,$$

$$w(k) = (w_1(k), w_2(k), \ldots, w_{h_i}(k))^T$$

и $(\sum_{i=1}^{n} h_i \times 1) = (h \times 1)$ -векторы центров $c(k)$, ширин $d^{-}(k)$, $\mu^c(x(k))$ с компонентами

$$4w_j(k)\left(\prod_{\substack{p=1 \\ p\neq j}}^{n}\mu_{pi}^F(x_i(k))\right)\mu_{ji}^Q(x_i(k))d_{ji}^{-2}(k)(c_{ji}(k)-x_i(k)),$$

$\mu^d(x(k))$ с компонентами

$$2w_j(k)\left(\prod_{\substack{p=1 \\ p\neq j}}^{n}\mu_{pi}^F(x_i(k))\right)\mu_{ji}^Q(x_i(k))(c_{ji}(k)-x_i(k))^2,$$

можно перейти к векторной форме алгоритма обучения (7), полностью совпадающей с предложенной выше системой рекуррентных соотношений (6).

4. Выводы

Предложен адаптивный алгоритм обучения нео-фаззи нейрона и нейро-фаззи сети с полиномиальными функциями активации, позволяющий настраивать в реальном времени не только синаптические веса, но и параметры введенных функций принадлежности четвертой степени, что обеспечивает высокое качество аппроксимации.

Алгоритм прост в вычислительном отношении, обладает высокой скоростью сходимости, а также следящими и сглаживающими свойствами.

Список литературы: 1. *Yamakawa T.*, Uchino E., Miki T., Kusanagi H. A neo fuzzy neuron and its applications to system identification and prediction of the system behavior // Proc. 2-nd Int. Conf. on Fuzzy Logic and Neural Networks (IIZUKA'92). Iizuka, Japan, 1992. P. 477–483. 2. *Uchino E.*, Yamakawa T. Soft computing based signal prediction, restoration, and filtering // Da Ruan (Ed.): Intelligent Hybrid Systems: Fuzzy Logic, Neural Networks, and Genetic Algorithms. Kluwer Academic Publishers, Boston, 1997. P. 331–349. 3. *Miki T.*, Yamakawa T. Analog implementation of neo-fuzzy neuron and its on-board learning // N.E. Mastorakis (Ed.): Computations Intelligence and Applications. WSES Press, Piraeus, 1999. P. 144–149. 4. *Landim R.*, Menezes B., Silva S. A Neo-Fuzzy-Neuron with Real Time Training Allied to Flux Observer for an Induction Motor // Proc. V-th Brazilian Symp. On Neural Networks (SBRN'98). Belo Horizonte, Brazil, 1998. P. 67–72. 5. *Bodyanskiy Ye.*, Kokshenev I., Kolodyazhniy V. An adaptive learning algorithm for a neo fuzzy neuron // Proc. 3rd Int. Conf. of European Union Society for Fuzzy Logic and Technology (EUSFLAT 2003), Zittau, Germany, 10-12 September 2003. P. 375–379. 6. *Bodyanskiy Ye.*, Kolodyazhniy V., Stephan A. An adaptive learning algorithm for a neuro-fuzzy network // B. Reusch (Ed.): Computational Intelligence. Theory and Applications. Springer-Verlag, Berlin-Heidelberg-New York, 2001. P. 68–75. 7. *Otto P.*, Bodyanskiy Ye., Kolodyazhniy V. A new learning algorithm for a forecasting neuro-fuzzy network // Integrated Computer-Aided Engineering. 10. N4. Amsterdam: IOS Press, 2003. P. 399–409. 8. *Осовский С.* Нейронные сети для обработки информации. М.: Финансы и статистика, 2002. 344 с.

Поступила в редколлегию 12.11.2004

УДК 621.396.961.1

В.М. Безрук, Н.П. Коваленко, В.А. Лысенко

ОБ ОДНОМ МЕТОДЕ АВТОМАТИЗИРОВАННОГО РАСПОЗНАВАНИЯ СТАДИЙ СНА ПО ЭЛЕКТРОЭНЦЕФАЛОГРАММАМ

1. Введение

Изучение структуры сна и его нарушений занимает важное место в современной экспериментальной и клинической медицине. Анализ структуры ночного сна играет важную роль в диагностике заболеваний головного мозга. При функциональных заболеваниях нервной системы и при психозах жалобы на нарушения сна нередко бывают основными, а иногда и единственными. Исследование структуры ночного сна оказывается полезным и при анализе функции и механизмов сна. В частности, актуальными становятся исследования структуры сна в кардиологической клинике у больных ишемической болезнью сердца, поскольку опасные кризы сердечно-сосудистой системы — нарушения сердечного ритма, инфаркт миокарда — часто возникают во время ночного сна [1, 2].

При изменении функционального состояния мозга изменяется и характер его электрической активности. Установлено, что электроэнцефалограмма (ЭЭГ) может служить индикатором уровня бодрствования мозга и адекватно отражать глубину сна. Успехи, достигнутые в области изучения организации сна, тесно связаны с применением электроэнцефалографических методов исследования. Однако распознавание стадий сна по записи ЭЭГ связано со значительными затратами времени, труда и средств, поскольку запись ЭЭГ ночного сна занимает сотни метров бумажной ленты. Кроме того, при визуальном анализе используются только описательные критерии для идентификации стадий сна, что порождает расхождения среди специалистов при оценке моментов перехода между стадиями. Поэтому возникает необходимость объективизации и автоматизации процесса исследования структуры сна. Для этого могут быть использованы разные математические методы распознавания сигналов [1–6]. Однако автоматизированное определение стадий сна по ЭЭГ с использованием математических методов связано с определенными трудностями, вызванных близостью спектрального состава ЭЭГ для отдельных фаз и стадий сна, отличием соответствующих ЭЭГ у разных людей, возникновением артефактов.

Возможны различные подходы к решению указанной задачи, которые определяются выбранным математическим описанием (математической моделью) ЭЭГ и соответствующим ему алгоритмом распознавания. Можно использовать спектральные методы распознавания стадий сна, основанные на описании ЭЭГ коэффициентами разложения в ряды по некоторым базисным функциям (ДЭФ, Уолша, Хаара) [1]. Известно также применение для описания ЭЭГ авторегрессионной (АР) модели [2]. Это дает возможность синтезировать авторегрессионные алгоритмы распознавания сигналов [2-6].

В данной статье в отличие от работы [2] для учета влияния артефактов предложено использовать метод распознавания заданных сигналов при наличии класса неизвестных сигналов [6]. Это позволит снизить вероятность ошибочных решений для решения задачи автоматического распознавания стадий сна в реальных условиях. Приведено описание предложенного авторегрессионного метода распознавания стадий сна по ЭЭГ, а также некоторые результаты его исследований, проведенные путем статистического моделирования на выборках реализаций ЭЭГ для 6-ти характерных стадий сна.

2. Математическая модель ЭЭГ и метод распознавания стадий сна

Предполагается, что ЭЭГ адекватно описывается математической моделью в виде процесса авторегрессии

$$X_t = a_1^i \cdot X_{t-1} + a_2^i \cdot X_{t-2} + \ldots + a_{p_i}^i \cdot X_{t-p} + \sigma_i v_t . \quad (1)$$

Параметры АР модели $A_i = \left(a_1^i, a_2^i, \ldots, a_{p_i}^i, \sigma_i, i, p_i\right)$ в каждый момент времени t удовлетворяют условиям стабильности и принадлежат к одному из M классов. Полагается, что параметры модели АР, выбранной для описания ЭЭГ, известны либо могут быть оценены по классифицированным обучающим выборкам реализаций ЭЭГ. Принято также считать, что параметры модели ЭЭГ остаются неизменными на значительном промежутке времени.

После получения очередного отсчета x_t реализации ЭЭГ при уже имеющихся отсчетах $\tilde{x}_t = \left(x_t, \ldots, x_{t-T}\right)$ необходимо ответить на вопрос, к какому классу принадлежит анализируемый участок ЭЭГ в момент времени $t \geq T$ (T — длительность анализируемого участка ЭЭГ).

В [2] показано решение такой задачи методом последовательной классификации по максимуму правдоподобия. Считается, что параметры модели в момент времени t принадлежат классу $\hat{i}_t$, если выполняется система неравенств

$$L_{t,\hat{i}_t}\left(A_{\hat{i}_t} / \tilde{x}_t\right) \geq L_{t,i}\left(A_i / \tilde{x}_t\right), \quad i \neq \hat{i}_t, \quad i = 1, \ldots, M, \quad (2)$$

где $L_{t,i}(A_i / \tilde{x}_t)$ — логарифмическая функция правдоподобия параметров. Логарифмическая функция правдоподобия $L_{t,i}\left(A_i / \tilde{x}_t\right)$ имеет вид [4]:

$$L_{t,i}(B_i/\tilde{x}_t) = -T\ln\sqrt{2\pi\sigma_i^2} + \\ +\frac{1}{2\sigma_i^2}\sum_{j=T-p_i}^{T}\left(x_t + a_1^{(i)}x_{t-1} + \ldots a_{p_i}^{(i)}x_{t-p_i}\right)^2. \quad (3)$$

Следует заметить, что в решающем правиле (2) функцию правдоподобия $L_{t,i}(B_i/\tilde{x}_t)$, $i=1,\ldots,M$ необходимо вычислять для каждого момента времени t, из-за чего для принятия решений необходимо выполнять значительный объем вычислений. Сокращение объема вычислений достигается при реккурентном вычислении функции правдоподобия, хотя и в этом случае процесс принятия решения оказывается очень трудоемким с точки зрения вычислительных затрат.

В ряде практических случаев длительность интервала времени анализа ЭЭГ T, по которому следует принимать решение, оказывается значительно меньше интервала времени постоянства параметров модели АР, выбранной для описания ЭЭГ. При этом можно предложить более простое по затратам решение поставленной задачи автоматизированного распознавания стадий сна с использованием рассмотренных в [5, 6] авторегрессионных алгоритмов распознавания сигналов. Кроме того, для учета появляющихся в реальных условиях работы артефактов в решающем правиле распознавания предлагается ввести дополнительный $M+1$-й класс неизвестных сигналов. С учетом выражения для многомерной плотности вероятности для гауссовских авторегрессионных последовательностей [4] алгоритм автоматизированного распознавания M заданных стадий сна по ЭЭГ при наличии класса неизвестных сигналов будет иметь следующий вид [6]:

$$H^k: K_k(\vec{x}) < \Lambda_k,\ k=\overline{1,M} \quad (4а)$$

$$K_k(\vec{x}) - K_i(\vec{x}) + \ln\frac{(2\pi\sigma_i)^{p_i-L}}{(2\pi\sigma_k)^{p_k-L}} \geq \ln\frac{P_k}{P_i}, \quad (4б)$$

$$H^{M+1}: K_k(\vec{x}) > \Lambda_k,\ k=\overline{1,M}. \quad (4в)$$

Здесь $K_k(\vec{x}) = \frac{1}{2\sigma_k^2}\sum_{l=p=1}^{L}\left[x_l - \mu_k - \sum_{j=p+1}^{p_k} a_j^k(x_{l-j}-\mu_k)\right]^2$ — соотношение, входящее в выражение для многомерной плотности вероятности (3) и определяющее нормированную ошибку предсказания в АР модели; $\Lambda_k = \ln\frac{(2\pi)^{\frac{L}{2}}\sigma_k^{L-p_k}\lambda_k}{P_k}$ — некоторые пороговые значения, определяемые из условия обеспечения заданных вероятностей правильного распознавания заданных M стадий сна; p_k, a_j^k — порядок и параметры АР модели ЭЭГ для k-ой стадии сна.

Согласно этому алгоритму распознавания решение в пользу i-й стадии сна принимается в два этапа: при выполнении хотя бы одного из неравенств (4а), а также при выполнении системы неравенств (4б). Когда выполняются неравенства (4в), решение принимается в пользу $M+1$-го класса неизвестных сигналов.

В алгоритме распознавания (4) полагаются известными параметры АР моделей. При неизвестных параметрах следует получить их оценки по классифицированным обучающим выборкам ЭЭГ [4–6]. При этом данный алгоритм работает в двух режимах: обучения и распознавания. В режиме обучения по классифицированным выборкам ЭЭГ для заданных стадий сна вычисляются оценки корреляционных функций, через которые определяются параметры модели АР (в частности из уравнения Юла-Уокера). На этапе распознавания по предъявленной для анализа реализации ЭЭГ длительностью T принимается решения о принадлежности ее к одному из M классов (соответствующих стадиям сна) либо принимается решение о ее принадлежности к классу неизвестных сигналов.

Геометрический смысл правила принятия решений согласно (4) состоит в том, что для ЭЭГ заданных стадий сна в пространстве сигналов строятся замкнутые собственные области. При попадании наблюдений в одну из собственных областей принимается решение в пользу соответствующей стадии сна. В противном случае наблюдаемая реализация ЭЭГ относится к $M+1$-му классу неизвестных сигналов.

Решающее правило (4) можно обобщить на случай принятия решений по ЭЭГ, поступающих из нескольких каналов регистрации. Для описания такой совокупности ЭЭГ может быть использована более сложная математическая модель в виде векторного процесса авторегрессии. При этом может быть получен алгоритм распознавания типа (4), соответствующий такой более сложной математической модели ЭЭГ.

3. Результаты исследований задачи автоматического распознавания стадий сна по ЭЭГ

В работах [5, 6] на выборках гауссовских случайных сигналов с различными корреляционными функциями были проведены исследования практических особенностей и рабочих характеристик авторегрессионных алгоритмов распознавания сигналов. Практический интерес представляет проведение исследований предложенного алгоритма распознавания заданных сигналов при наличии неизвестных сигналов (4) для случая решения задачи распознавания стадий сна по ЭЭГ. Для исследований разработан специальный пакет программ, реализующий алгоритм в режиме обучения и распознавания.

Приведем некоторые результаты исследований, выполненных на выборках реализаций ЭЭГ для 6-ти стадий сна. В таблице приведены параметры АР модели, которые были получены по реальным ЭЭГ для каждой из 6-ти стадий сна [2]. На рис. 1 для

иллюстрации приведены полученные с помощью этой модели реализации ЭЭГ для разных стадий сна. Соответствующие им энергетические спектры ЭЭГ приведены на рис. 2. При исследованиях использовались выборки реализаций ЭЭГ объемом по 50 000 отсчетов для каждой из 6-ти стадий сна. По классифицированным обучающим выборкам ЭЭГ найдены параметры решающего правила (4).

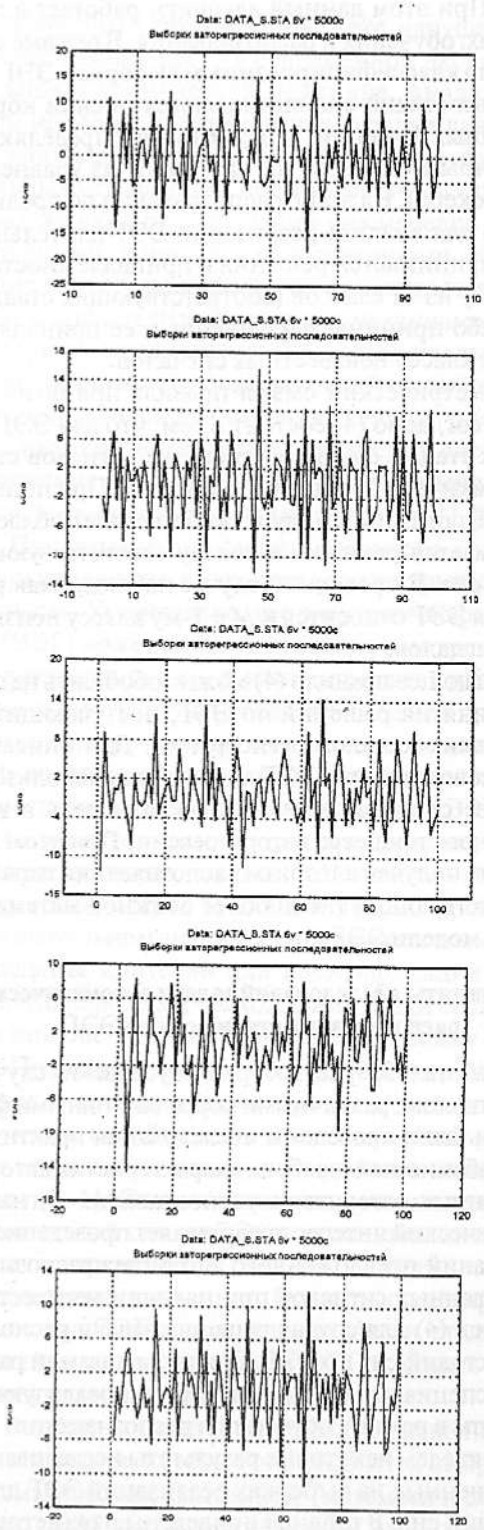


Рис. 1. Выборки реализаций ЭЭГ для 6-ти стадий сна

Контрольные выборки использованы при исследованиях практических особенностей решения задачи автоматизированного распознавания стадий сна по ЭЭГ. Исследована зависимость качества распознавания стадий сна от интервала наблюдения ЭЭГ T для текущей стадии сна (рис. 3).

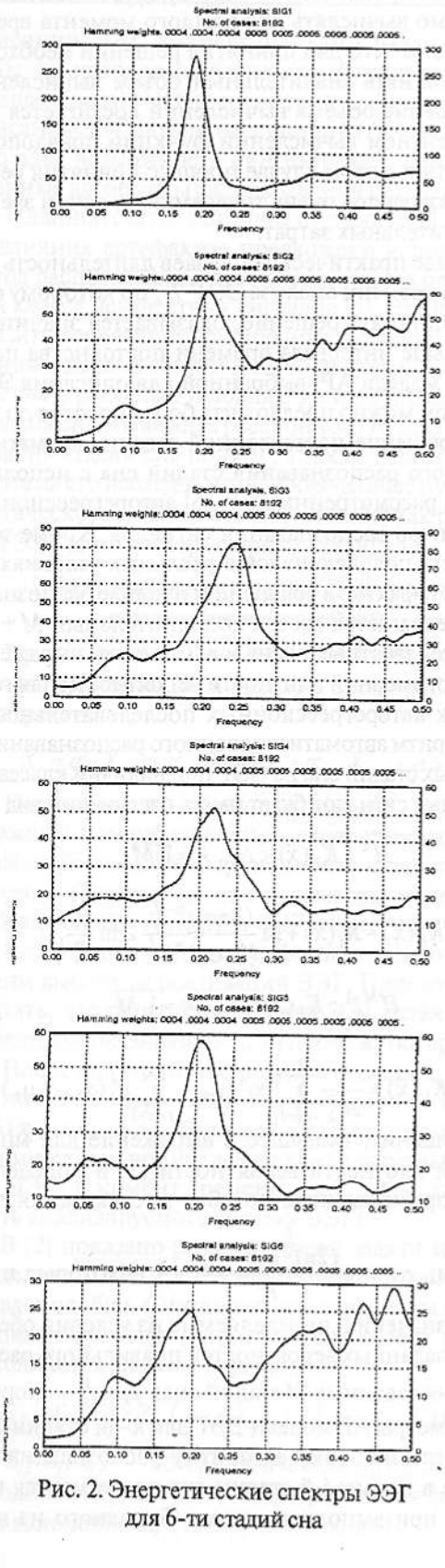


Рис. 2. Энергетические спектры ЭЭГ для 6-ти стадий сна

Таблица

Значения параметров авторегрессионной модели, полученные по реальным ЭЭГ для 6-ти разных стадий сна

	$i=1$	$i=2$	$i=3$	$i=4$	$i=5$	$i=6$
σ_i	25,4	18,3	21,5	15,7	15,9	11,23
a_1	-0,61	-0,51	-0,24	0,06	0,12	-0,38
a_2	-0,67	-0,42	-0,33	-0,18	-0,5	-0,26
a_3	-0,75	-0,37	-0,25	-0,13	-0,11	-0,33
a_4	-0,62	-0,24	-0,06	0,08	0,09	-0,28
a_5	-0,32	-0,13	-0,06	0,03	0,07	-0,17
a_6	-0,16	-0,17	-0,17	-0,04	-0,04	-0,12
a_7	-0,17	-0,22	-0,14	-0,08	-0,06	-0,16
a_8	-0,15	-0,18	-0,05	-0,02	-0,01	-0,15
a_9	-0.09	-0,13	-0,05	-0,04	-0,01	-0,09
a_{10}	-0,03	-0,04	-0,05	-0,01	-001	-0,03

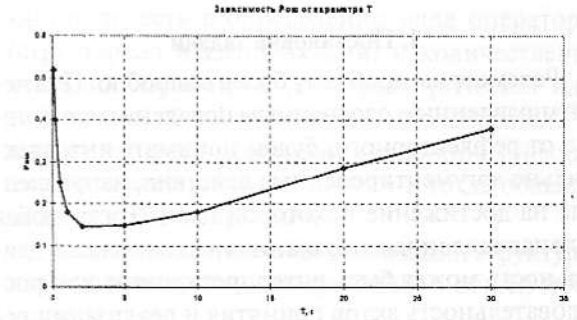


Рис. 3. Зависимость средней вероятности ошибочного распознавания стадий сна от длительности интервала ЭЭГ, используемой для принятия решений

Получено минимальное значение средней вероятности ошибочного распознавания стадий сна $P_{\text{ош. ср}} = 0{,}15$.

В результате предъявления на распознавание контрольных выборок ЭЭГ для разных стадий сна получены диаграммы изменений стадий сна, удобные для оперативного анализа структуры сна пациента и последующей постановки диагноза. Один из фрагментов диаграммы приведен на рис. 4.

Рис. 4. Диаграмма изменений стадий сна, полученная по результатам распознавания выборок ЭЭГ

4. Выводы

В статье предложен новый метод автоматизированного распознавания заданных стадий сна по ЭЭГ при наличии класса неизвестных сигналов. Метод основан на описании ЭЭГ математической моделью в виде авторегрессионных процессов. При распознавании наблюдаемой ЭЭГ решение принимается либо в пользу одной из стадий сна, либо в пользу класса неизвестных наблюдений. Такой метод распознавания позволяет снизить вероятность ошибочных решений в реальных условиях при появлении различного рода артефактов. Метод распознавания стадий сна по ЭЭГ может быть использован при создании автоматизированных систем информационной поддержки для диагностики заболеваниях нервной и сердечно-сосудистой систем.

Приведены некоторые результаты исследования практических особенностей автоматизированного распознавания стадий сна по ЭЭГ. Исследования выполнены путем статистического моделирования на выборках реализаций ЭЭГ для 6-ти характерных стадий сна. Получена зависимость ошибки распознавания стадий сна от интервала наблюдения ЭЭГ, по которой принимаются решения. Получены также временные диаграммы изменений стадий сна, представляющие практический интерес для оперативной оценки структуры ночного сна.

С целью дальнейшего уменьшения вероятности ошибочного распознавания стадий сна следует использовать ЭЭГ, получаемые из нескольких каналов электроэнцефаллографа (8 или 16). При этом для математического описания ЭЭГ необходимо применить векторную авторегрессионную модель и синтезировать соответствующий алгоритм распознавания ЭЭГ.

Для получения окончательных рекомендаций по практическому применению предложенного метода автоматизированного распознавания стадий сна необходимо продолжить клинические исследования на реальных ЭЭГ для ночного сна с использованием многоканального электроэнцефаллографа, а также с учетом различного рода артефактов, обусловленных открыванием глаз, изменением положения электродов и другим.

Список литературы 1. *Неймарк Ю.П.* Распознавание образов в медицинской диагностике. М.: Наука, 1985. 376 с. 2. *Липейка А.*, Малинаускас В., Гринявичус К., Лесене В. Автоматическое определение стадий сна по электроэнцефаллограмме //Статистические проблемы управления. Вильнюс: УМ АН Лит ССР, 1981. Вып.51. С.85-97. 3. *Клигене Н.*, Тельксних Л. Методы обнаружения моментов измерения свойств случайных процессов //Автоматика и телемеханика. 1983. № 10. С.13-18. 4. *Мисюкас М.* Несколько конструктивных выражений асимптотически байесовских классификаторов в явном виде для гауссовских стационарных временных рядов // Статистические проблемы управления. Вильнюс: ИМ АН Литвы. 1985. Вып.69.С.81-87. 5. *Кравченко Н.И.*, Безрук В.М., Тихонов В.А. Распознавание случайных сигналов в рамках авторегрессионной модели //Вероятностные модели и обработка случайных сигналов и полей. К: УМК ВО. 1991. С. 138-142. 6. *Безрук В.М.*, Коваленко Н.П. Синтез и анализ алгоритмов распознавания гауссовких случайных сигналов при наличии класса неизвестных сигналов на основе авторегрессионной модели // АСУ и приборы автоматики. 2000. Вып.111. С.115-120.

Поступила в редколлегию 05.11.2004

УДК 519.08

К.Э. Петров

ПРОБЛЕМЫ ФОРМАЛИЗАЦИИ ИНТЕЛЛЕКТУАЛЬНОЙ ДЕЯТЕЛЬНОСТИ ЧЕЛОВЕКА

Введение

Вычислительная техника (ВТ) возникла и развивается как инструмент автоматизации интеллектуальной деятельности человека. Однако необходимым условием ее конструктивного применения является наличие алгоритмического описания реализуемой интеллектуальной деятельности. Для этого, в свою очередь, необходимо формальное описание (построение математической модели) интеллектуального процесса. Таким образом, прогресс в автоматизации интеллектуальной деятельности определяется следующими факторами:

- уровнем развития ВТ и ее функциональных характеристик (объема памяти, быстродействия, средств визуализации, сервисных функций и т. д.);
- наличием машинно-ориентированных алгоритмов, которые описывают эти интеллектуальные процессы.

В ходе развития процесса автоматизации интеллектуальной деятельности можно выделить два этапа. Первый из них связан с бурным распространением вычислительной техники «вширь», проникновением ее в различные сферы человеческой деятельности на основе реализации, в основном, простейших рутинных интеллектуальных функций: запоминания, хранения, поиска данных, арифметической обработки цифровой информации. На этом этапе практически отсутствовали затруднения с алгоритмическим обеспечением, так как человечество в процессе своего развития к моменту возникновения ВТ накопило большой объем моделей и алгоритмов решения рутинных процедур обработки данных ориентированных на передачу знаний (обучение). При этом главным сдерживающим фактором являлись ограниченные функциональные характеристики универсальных ЭВМ. Однако в середине 90-х годов это ограничение было практически снято, что открыло перспективы перехода ко второму этапу, ориентированному на автоматизацию более сложных интеллектуальных процессов и процедур. Успехи в этом направлении оказались значительно скромнее, чем предполагалось вначале. Прилагательные «интеллектуальные», «знаниеориентированные» и т. п. применительно к компьютерным информационным системам отражают скорее намерения разработчиков, чем конкретные результаты. Не вдаваясь в подробности, можно констатировать, что обещания середины 90-х годов относительно экспертных и других систем, основанных на интеллектуальных процедурах принятия решений, не оправдались. Это обусловлено тем, что в настоящее время не решены проблемы синтеза и идентификации формальных (математических) моделей интеллектуальной деятельности человека и получения знаний от их носителя. Дальнейшие перспективы использования ВТ в определенной степени зависят от прогресса в этой области.

1. Постановка задачи

Рассмотрим проблему более подробно. Под целенаправленным осознанным поведением, в отличие от рефлекторного, будем понимать интеллектуально аргументированные действия, направленные на достижение некоторой цели. Тогда любая целенаправленная осознанная человеческая деятельность может быть интерпретирована как последовательность актов принятия и реализации решений. В свою очередь общий процесс принятия решений может быть структурирован следующим образом:

$$Q \to I(Q) \to C \to X \to P(x) \to x^0, \qquad (1)$$

где Q – некоторая исходная ситуация в контексте окружающей среды; $I(Q)$ – индивидуальный информационный портрет ситуации Q; C и X – соответственно цель и множество путей ее достижения (множество решений), которые формируются на основе интеллектуального анализа $I(Q)$; $P(x)$ – некоторая, в общем случае многофакторная оценка качества решений $x \in X$, известная как функция полезности [1]; x^0 — выбранное индивидуумом решение. Формальная модель выбора x^0 основана на гипотезе рационального поведения, постулирующей, что индивидуум стремится экстремизировать полезность поведения:

$$x^0 = \arg \operatorname*{extr}_{x \in X} P(x). \qquad (2)$$

Структура (1) инвариантна предметной области деятельности, то есть одинакова, например, при постановке медицинского диагноза, решении технических, социальных и тому подобных проблем. Таким образом, прогресс в создании и развитии интеллектуальных компьютерных систем связан с математической формализацией и алгоритмизацией указанных процессов, то есть с синтезом и идентификацией математических моделей интеллектуальной деятельности. Не умаляя важности и слож-

ности формализации всех перечисленных выше этапов интеллектуальной деятельности подчеркнем концептуальную важность этапа оценивания, то есть процедуры формирования конструктивной многофакторной оценки полезности допустимых альтернатив.

В самом общем виде любую математическую модель можно представить в виде

$$Y = F(\Lambda, X), \qquad (3)$$

где Y – реакция (поведение системы); X – входные воздействия; F – оператор, устанавливающий связь между входом и выходом; Λ – параметры (количественные характеристики) этого оператора. Тогда проблема синтеза модели в общем случае заключается в ее структурно-параметрической идентификации, то есть в определении вида оператора F (структурная идентификация) и количественных значений параметров Λ (параметрическая идентификация). В частном случае исследователь может располагать некоторой априорной информацией о характеристиках модели и тогда общая задача несколько упрощается.

Рассмотрим проблемы реализации структурно-параметрической идентификации в общей постановке.

Традиционный путь решения этой задачи заключается в проведении с реальной системой, модель которой строится, пассивных или активных экспериментов с целью получения количественной информации о реакции системы (Y) на входные воздействия (X). На основе анализа этой информации выдвигается гипотеза о характере связи между Y и X, и в рамках некоторого абстрактного языка описывается эта связь (формируется оператор F). Затем определяются параметры модели Λ и далее проверяются ее адекватность и точность путем сравнения реакции (откликов Y) модели и реальной системы на экспериментально измеренные входные сигналы X. На основе анализа результатов принимается решение о справедливости принятой гипотезы, и тогда производится эволюционное совершенствование модели путем изменения структуры (как правило, в сторону усложнения), а также уточнение параметров модели. В противном случае принимается новая гипотеза и процесс итерационно повторяется.

Существует два принципиальных подхода построения абстрактных моделей: синтез моделей прямой и непрямой аналогии. Первый подход ориентирован на как можно более точное описание реальных процессов, происходящих в исследуемой системе. Примером такой модели является модель генома человека, построение которого связано со скрупулезным восстановлением всех генных структур ответственных за формирование и жизнедеятельность человека. Другой подход ориентирован на создание моделей, адекватных с точки зрения выходных данных, то есть моделей, достаточно точно аппроксимирующих зависимость между входом и выходом некоторым полиномом. Оба подхода дополняют друг друга и ориентированы соответственно на микро и макро описание процесса.

Принципиальной особенностью проблемы идентификации моделей интеллектуальной деятельности является невозможность количественного измерения результатов мозговой деятельности. Это означает, что если рассмотреть последовательность этапов (1), то можно зарегистрировать только принятые индивидуумом решения x^0, но невозможно непосредственно измерить количественные значения оценки их полезности (качества) $P(x^0)$. Это обстоятельство делает принципиально невозможным использование описанного выше традиционного подхода к решению задачи идентификации и требует создания новых методов.

2. Структурная и параметрическая идентификация моделей интеллектуальной деятельности

В настоящее время для решения проблемы используется интроспективный подход известный как методология экспертного оценивания. Он основан на том обстоятельстве, что предметом исследования является осознанная интеллектуальная деятельность. Это означает, что индивидуум располагает знаниями, то есть является носителем знаний о том, как он реализует ту или иную интеллектуальную процедуру. Необходимо извлечь эти знания. Для этого индивидуум (эксперт) побуждается тем или иным способом (анкетирование, интервьюирование, различные процедуры экспертного оценивания) к осознанию, структуризации и количественному оцениванию модели собственной интеллектуальной деятельности. Это означает, что с ним проводятся активные эксперименты по извлечению знаний.

Индивидуум достаточно уверенно принимает на эвристическом уровне рациональные профессиональные или бытовые решения, но при структуризации этого процесса на большое количество этапов (процедур) с необходимостью их формализации, в большинстве случаев на количественном уровне, происходит последовательное накопление погрешностей, что делает синтезированную модель зачастую неадекватной. В частности, нарушается условие транзитивности решений или модельный выбор не соответствует действительному.

Это обусловлено тем, что интеллектуальный анализ ситуации производится чаще всего на основе качественных оценок предпочтений, то есть выполняется в качественных шкалах [1]. Оценка силы предпочтения требует перехода к количественным

шкалам. На этом этапе и возникают указанные выше погрешности.

Таким образом, необходимы альтернативные, более объективные методы идентификации моделей интеллектуальной деятельности (извлечения знаний).

Одним из таких возможных подходов является метод компараторной идентификации [2, 3], который позволяет построить достаточно широкий класс проблемно-ориентированных идентификационных процедур.

Рассмотрим особенности реализации метода компараторной идентификации применительно к задаче синтеза модели многофакторного оценивания.

Постановка задачи имеет следующий вид.

Задано множество допустимых альтернативных решений $X=\{x_j\}$, $j=\overline{1,m}$. Каждое решение характеризуется одинаковым по структуре кортежем частных разнородных характеристик (критериев) $K=\langle k_i(x_j)\rangle$, $i=\overline{1,n}$, $j=\overline{1,m}$. Из множества X индивидуум (ЛПР) выбирает некоторую альтернативу $x^0 \in X$. Это решение фиксируется. И затем на основе этой информации необходимо идентифицировать модель выбора индивидуума (ЛПР).

Теоретической основой являются следующие гипотезы.

1. Существует скалярная количественная оценка полезности альтернатив $p[k_i(x_j)]$ [4].

2. Индивидуум реализует некоторое рациональное поведение в процессе выбора, которое описывается уравнением (2).

3. Возможно установление отношения порядка для любой пары альтернатив $x_1, x_2 \in X$ (предпочтения $x_1 \succ x_2$ или эквивалентности $x_1 \sim x_2$).

4. Невозможно непосредственно измерить численное значение индивидуальной функции полезности выбранной альтернативы.

Исходя из вышесказанного, если индивидуум в качестве наилучшего решения выбрал альтернативу $x^0 \in X$, то можно записать:

$$p[k_i(x^0)] \geq p[k_i(x_j)],\ \forall j=\overline{1,m},\ i=\overline{1,n} \qquad (4)$$

или

$$p[k_i(x^0)]-p[k_i(x_j)] \geq 0,\ \forall j=\overline{1,m},\ i=\overline{1,n}. \qquad (5)$$

Таким образом получаем систему $(m-1)$ неравенств, которая определяет некоторую область возможных значений искомых переменных. Поэтому в общем случае задача компараторной идентификации является некорректной по Адамару, что порождает ряд дополнительных трудностей, связанных с необходимостью ее регуляризации.

Рассмотрим эту проблему подробнее.

На основе системы неравенств (5) необходимо решить задачу структурно-параметрической идентификации функции оценивания $p[k_i(x_j)]$.

Возможны два подхода к решению этой задачи:

- решение задачи структурной идентификации на основе аксиоматического постулирования и последующего определения параметров модели;
- решение задачи в общей постановке без априорной информации о структуре модели.

Рассмотрим перечисленные выше подходы более подробно.

В настоящее время наиболее распространенными являются аддитивная

$$P_1(x_j)=\sum_{i=1}^{n} a_i p[k_i(x_j)],\ \sum_{i=1}^{n} a_i=1,\ a_i \geq 0 \qquad (6)$$

и мультипликативная

$$P_2(x_j)=\prod_{i=1}^{n} p[k_i(x_j)] \qquad (7)$$

функции полезности.

Мультипликативная функция (7) не позволяет учитывать различную важность частных характеристик (весовые коэффициенты a_i) [1], и потому более универсальной и чаще используемой является аддитивная функция (6). Тогда для функции (6) система неравенств (5) примет вид

$$\sum_{i=1}^{n} a_i\left(p[k_i(x^0)]-p[k_i(x_j)]\right) \geq 0,\ \sum_{i=1}^{n} a_i=1,\ a_i \geq 0,$$
$$\forall j=\overline{1,m},\ i=\overline{1,n}, \qquad (8)$$

где неизвестными являются численные значения весовых коэффициентов a_i. Все неравенства системы (8) являются линейными и, следовательно, образуют в положительном ортанте выпуклый многогранник на гиперплоскости $\sum_{i=1}^{n} a_i=1$, любая точка которого удовлетворяет (8). В качестве точечных значений весовых коэффициентов a_i можно принять Чебышевскую точку [1], центр тяжести многогранника, а также определить эти значения, используя теорию генетических алгоритмов [5] или метод случайного поиска.

Задача структурно-параметрической идентификации функции оценивания в общей постановке связана с определением модели оптимальной сложности в классе полинома Колмогорова-Габора. Ее решение можно осуществить, используя метод группового учета аргументов, теорию генетических алгоритмов и, возможно, с помощью искусственных нейронных сетей. При этом критерием селекции промежуточных моделей является степень совпадения отношения порядка альтернатив на обучающей эталонной последовательности [5, 6].

Проблема заключается в исследовании трудоемкости, точности и адекватности всех перечисленных методов на тестовых примерах и выработки рекомендаций по области их применения.

Заключение

Принципиальной особенностью проблемы идентификации математических моделей интеллектуальной деятельности человека является невозможность количественного измерения результатов мозговой деятельности. Это обстоятельство делает принципиально невозможным использование традиционных подходов к идентификации и требует создания новых методов.

В связи с этим в статье предложены подходы к структурной и параметрической идентификации моделей интеллектуальной деятельности, основанные на теории компараторной идентификации. Рассмотрены основные трудности, возникающие в ходе решения этой проблемы. Показано, что решение данной задачи связано с определением модели оптимальной сложности в классе полинома Колмогорова-Габора, которое можно осуществить, используя метод группового учета аргументов, теорию генетических алгоритмов или искусственных нейронных сетей.

Список литературы: 1. *Овезгельдыев А.О., Петров Э.Г., Петров К.Э.* Синтез и идентификация моделей многофакторного оценивания и ранжирования. К.: Наукова думка, 2002. 164 с. 2. *Шабанов–Кушнаренко Ю.П.* Теория интеллекта. Проблемы и перспективы. Харьков: Изд. ХГУ, 1987. 160 с. 3. *Овезгельдыев А.О., Петров К.Э.* Компараторная идентификация моделей интеллектуальной деятельности// Кибернетика и системный анализ. 1996. № 5. С. 48 – 58. 4. *Нейман Дж., Моргенштерн О.* Теория игр и экономическое поведение. М.: Наука, 1970. 124 с. 5. *Петров Э.Г., Булавин Д.А. Петров К.Э.* Использование генетических алгоритмов для решения задачи структурно-параметрической идентификации модели индивидуального многофакторного оценивания// Проблемы бионики. 2004. № 60. С. 17 – 27. 6. *Петров Э.Г., Булавин Д.А. Петров К.Э.* Решение задачи структурно-параметрической идентификации модели индивидуального многофакторного оценивания методом группового учета аргументов// Проблемы бионики. 2003. № 58. С. 36 – 45.

Поступила в редколлегию 28.10.2004

УДК 519.6:514.1

И.В. Гребенник

МОДЕЛИ ОПТИМИЗАЦИИ НА КОМПОЗИЦИОННЫХ ОБРАЗАХ КОМБИНАТОРНЫХ МНОЖЕСТВ В СИСТЕМАХ ПОДДЕРЖКИ ПРИНЯТИЯ РЕШЕНИЙ

Введение

При разработке интеллектуальных систем поддержки принятия решений различного назначения важную роль играют математические модели объектов, составляющих предметную область, и методы их анализа. Применение математических моделей и методов позволяет повысить эффективность решений, принимаемых с помощью интеллектуальных систем. В настоящее время системы поддержки принятия решений создаются в различных областях, одной из которых является геометрическое проектирование [1].

При решении задач геометрического проектирования актуальной остается проблема создания интеллектуальных систем решения оптимизационных задач размещения. Несмотря на высокий уровень организации архитектуры баз знаний существующих интеллектуальных систем решения задач размещения геометрических объектов, их построение, как правило, основывается на использовании идеализированных математических моделей, когда погрешности метрических характеристик и параметров размещения геометрических объектов не учитываются. Это объясняется недостаточно разработанной фундаментальной основой построения математических моделей и методов решения задач данного класса с учетом погрешностей исходных данных.

Важный класс задач геометрического проектирования составляют экстремальные задачи с дискретными параметрами. Построение математических моделей таких задач основано на применении комбинаторных множеств, составляющих их области допустимых решений [1, 2].

Во многих случаях специфика решаемой задачи требует отражения в модели ее комбинаторной структуры. Эта структура может быть достаточно сложной и не позволять использовать для ее описания классические комбинаторные множества [3]. Для моделирования задач со сложной комбинаторной структурой в [4] введено понятие и предложен способ описания композиционного образа (k-образа) комбинаторных множеств. K-образами комбинаторных множеств являются комбинаторные множества, порождающие элементы которых сами представляют собой элементы других комбинаторных множеств. K-образы комбинаторных множеств могут быть использованы для описания областей допустимых решений экстремальных задач со сложной структурой.

Целью настоящей работы является постановка и решение некоторых классов задач оптимизации на k-образах комбинаторных множеств и применение этих результатов в системах поддержки принятия решений. При этом необходимо предварительно сформировать классы k-образов комбинаторных множеств как областей допустимых решений задач комбинаторной оптимизации.

1. Определение множеств парных перестановок, парных размещений и парных сочетаний с повторениями

Ориентируясь на известные модели задач с дискретными параметрами [1, 2], рассмотрим k-образы комбинаторных множеств, в которых в качестве базовых выступают евклидовы комбинаторные множества перестановок, размещений и сочетаний.

Учет погрешностей при решении задач геометрического проектирования вызывает необходимость построения адекватных математических моделей задач этого класса. Основой для построения моделей задач геометрического проектирования, учитывающих погрешности метрических характеристик и параметров размещения объектов, являются методы интервального анализа [5]. Один из способов анализа интервальных математических моделей основан на их отображении в евклидово пространство [6]. При этом образом каждого интервала в R^N является пара чисел, характеризующих центр и радиус интервала. Для моделирования образов интервальных комбинаторных множеств в R^N построим k-образы комбинаторных множеств парных перестановок, парных размещений и парных сочетаний.

Рассмотрим множество

$$A=\{a_1,a_2,...,a_n\}=\left\{g_1^{\rho_1},g_2^{\rho_2},...,g_{k_a}^{\rho_{k_a}}\right\},$$

где $\rho_i \in J_n$, $i \in J_{k_a}$ — кратности элементов множества A, $\rho_1+\rho_2+...+\rho_{k_a}=n$; $g_1<g_2<...<g_{k_a}$, и множество $B=\{b_1,b_2,...,b_n\}$, $a_i \in R^1$, $b_i \in R^1$. Построим множества $Z_1,Z_2,...,Z_n$ вида $Z_i=\{(a_i,b_i)\}$, $i \in J_n$. При этом k множеств Z_i из n являются различными, $k_a \le k \le n$.

Используя подход, приведенный в [4], сформируем следующие классы k-образов комбинаторных множеств.

1. Композиционный образ комбинаторных множеств P_{nk}, $Z_1,Z_2,...,Z_n$; порожденный множе-

ствами $\{a_1,b_1\},\{a_2,b_2\},...,\{a_n,b_n\}$, где P_{nk} — множество перестановок из n элементов, k из которых различны [1, 2]. Обозначим такой k-образ множеств через $PI_{nk}(Z_1,Z_2,...,Z_n)$ или PI_{nk} и назовем множеством парных перестановок.

2. Композиционный образ комбинаторных множеств A_{nk}^m, $Z_1,Z_2,...,Z_n$, порожденный множествами $\{a_1,b_1\},\{a_2,b_2\},...,\{a_n,b_n\}$. Здесь A_{nk}^m — общее множество размещений из n элементов, k из которых различны, по m [2]. Такой k-образ множеств обозначим через $AI_{nk}^m(Z_1,Z_2,...,Z_n)$ или AI_{nk}^m и назовем множеством парных размещений.

3. Композиционный образ комбинаторных множеств $\bar{C}_n^m$, $Z_1,Z_2,...,Z_n$, порожденный множествами $\{a_1,b_1\},\{a_2,b_2\},...,\{a_n,b_n\}$, где $Z_i=\{(a_i,b_i)\}$ различны, $i\in J_n$. Здесь $\bar{C}_n^m$ — множество сочетаний с повторениями из n различных элементов по m [2], причем для любого $h=(g_{i_1},g_{i_2},...,g_{i_m})\in\bar{C}_n^m$ справедливо $g_{i_1}\le g_{i_2}\le...\le g_{i_m}$. Предположим, что кратности ρ_i элементов $g_1,g_2,...,g_{k_a}$ множества A равны $\rho_1=\rho_2=...=\rho_{k_a}=m$, а множество B имеет вид: $B=\{b_1,b_2,...,b_n\}=\left\{e_1^{\rho_1},e_2^{\rho_2},...,e_{k_a}^{\rho_{k_a}}\right\}$, $\rho_1=\rho_2=$ $=...=\rho_{k_a}=m$. При этом множества Z_i можно представить как $Z_1=Z_2=...=Z_m=\{(g_1,e_1)\}$, $Z_{m+1}=Z_{m+2}=$, $=...=Z_{2m}=\{(g_2,e_2)\}$, ..., $Z_{n-m+1}=Z_{n-m+2}=...=Z_n=$ $=\{(g_{k_a},e_{k_a})\}$. Этот k-образ множеств обозначим $\bar{C}I_n^m(Z_1,Z_2,...,Z_n)$ или $\bar{C}I_n^m$ и назовем множеством парных сочетаний с повторениями.

2. Постановка задачи

Пусть X-композиционный образ комбинаторных множеств [4], $X\in\left\{PI_{nk},AI_{nk}^m,\bar{C}I_n^m\right\}$. Рассмотрим задачу оптимизации

$$\Phi(h)\to\min_{h\in X},\qquad(1)$$

$$h\in H\subseteq X,$$

где $\Phi: X\to R^1$ — некоторый функционал.

В результате погружения f множества X в R^N [1, 2] сформулируем задачу оптимизации функции $\varphi(x)$, эквивалентную (1):

$$\varphi(x)\to\min,\qquad(2)$$
$$x\in E_f\subseteq E,$$

где $\varphi: E\to R^1$ — функция N переменных, определенная на множестве $E\subset R^N$; $E=f(X)$ — образ множества X в пространстве R^N; $\varphi(x)=\Phi(h)$ при $x=f(h)$, $\forall h\in H$.

3. Задачи с линейной целевой функцией без дополнительных ограничений

Рассмотрим случай, когда в задаче (2) функция $\varphi(x)$ является линейной, а в качестве множества E выступают образы множеств парных перестановок EI_{nk}, парных размещений EI_{nk}^m, парных сочетаний с повторениями $\bar{S}I_n^m$ в пространстве R^N:

$$\varphi(x)=\sum_{i=1}^{N}c_ix_i\to\min,\qquad(3)$$

где $x=(x_1,x_2,...,x_N)\in E\subset R^N$; $c_i\in R^1$; $i\in J_N$.

Отметим, что задачи минимизации линейных функций на евклидовых комбинаторных множествах исследовались ранее [7–9]. Распространим изложенный в этих работах подход на задачу оптимизации вида (3).

Пусть множество E в задаче (3) представляет собой множество парных перестановок EI_{nk}. Тогда задача оптимизации примет вид:

$$\varphi(x)=\sum_{i=1}^{w}c_ix_i\to\min,\qquad(4)$$

где $x=(x_1,x_2,...,x_w)\in EI_{nk}\subset R^w$; $w=2n$; $c=(c_1,c_2,...,c_w)\in R^w$.

Напомним, что координаты точек множества EI_{nk} принимают значения всевозможных перестановок пар (a_i,b_i), где $a_i\in A$, $b_i\in B$, $i\in J_n$. Зададим на множестве пар (a_i,b_i) следующее бинарное отношение:

$$((a_{i_j},b_{i_j})\prec_c(a_{i_k},b_{i_k}))\Leftrightarrow((c_{2j-1}-c_{2k-1})(a_{i_j}-a_{i_k})+$$
$$+(c_{2j}-c_{2k})(b_{i_j}-b_{i_k})\le0),\qquad(5)$$

где $c\in R^w$ — вектор коэффициентов функции цели задачи (4), $i_j,i_k\in J_n$, $j,k\in J_n$.

Отметим некоторые свойства введенного отношения.

1. Из структуры соотношения (5) и произвольности выбора пар $(a_{i_j},b_{i_j}),(a_{i_k},b_{i_k})$ следует, что любые две пары из множества $\{(a_1,b_1),(a_2,b_2),...,(a_n,b_n)\}$ сравнимы между собой по отношению $\prec_c$.

2. Отношение $\prec_c$ рефлексивно, так как для него выполняется условие $(a_{i_j},b_{i_j})\prec_c(a_{i_j},b_{i_j})$. Это следует из того, что

$$(c_{2j-1}-c_{2j-1})(a_{i_j}-a_{i_j})+(c_{2j}-c_{2j})(b_{i_j}-b_{i_j})=0.$$

3. Отношение $\prec_c$ антисимметрично, поскольку для него справедливо соотношение

$$((a_{i_j},b_{i_j})\prec_c(a_{i_k},b_{i_k}))\&((a_{i_k},b_{i_k})\prec_c(a_{i_j},b_{i_j}))\Rightarrow$$
$$\Rightarrow((a_{i_k},b_{i_k})=(a_{i_j},b_{i_j})).$$

Его справедливость вытекает из того, что неравенства

$$(c_{2j-1}-c_{2k-1})(a_{i_j}-a_{i_k})+(c_{2j}-c_{2k})(b_{i_j}-b_{i_k})\le0,$$
$$(c_{2k-1}-c_{2j-1})(a_{ik}-a_{i_j})+(c_{2k}-c_{2j})(b_{i_k}-b_{ij})\le0$$

могут одновременно выполняться только при $a_{i_j}=a_{i_k}$, $b_{i_j}=b_{i_k}$.

Выстроим пары (a_i,b_i) в соответствии с отношением (5). Пусть последовательность $\{i_1,i_2,...,i_n\}$ такова, что

$$(a_{i_1},b_{i_1}) \underset{c}{\prec} (a_{i_2},b_{i_2}) \underset{c}{\prec} ... \underset{c}{\prec} (a_{i_n},b_{i_n}). \quad (6)$$

Справедливо следующее утверждение.

Теорема 1. Минимум линейной функции $\varphi(x)$ задачи (4) на множестве EI_{nk}, порожденном множествами $Z_i=\{a_i,b_i\}$, $i\in J_n$, достигается в точке

$$x^*=(x_1^*,x_2^*,...,x_w^*)\in EI_{nk}, \quad (7)$$

где $x_{2j-1}^*=a_{i_j}$; $x_{2j}^*=b_{i_j}$; $i_j\neq i_t$ при $t\neq j$, $i_j\in J_n$, $j\in J_n$, а последовательность $\{i_1,i_2,...,i_n\}$ удовлетворяет (6).

Доказательство. Предположим противное, пусть минимум функции $\varphi(x)$ задачи (4) на множестве EI_{nk} достигается в точке $y\in EI_{nk}$, отличной от x^* вида (7). Это значит, что можно указать такое значение $p\in J_w$, что $x_j^*=y_j$, $j\in J_{p-1}$, $x_p^*\neq y_p$. Рассмотрим два случая.

1. $p\in I_w'=\{1,3,...,w-1\}$. Тогда существует такое $q\in I_w'$, $q>p$, что $y_q=x_p^*$, $y_{q+1}=x_{p+1}^*$. В векторе y выполним транспозиции значений координат y_p и y_q, y_{p+1} и y_{q+1}. Полученный вектор обозначим $\bar{y}$. Очевидно, $\bar{y}\in EI_{nk}$. Рассмотрим разность

$$\varphi(\bar{y})-\varphi(y)=c_py_q+c_{p+1}y_{q+1}+c_qy_p+c_{q+1}y_{p+1}-$$

$$-c_py_p-c_{p+1}y_{p+1}-c_qy_q-c_{q+1}y_{q+1}=$$

$$=(c_p-c_q)(y_q-y_p)+(c_{p+1}-c_{q+1})(y_{q+1}-y_{p+1})=$$

$$=(c_p-c_q)(x_p^*-y_p)+(c_{p+1}-c_{q+1})(x_{p+1}^*-y_{p+1}).$$

В соответствии с (6), (7) $x_p^*=a_{i_t}$, $x_{p+1}^*=b_{i_t}$, где $t=\frac{p+1}{2}$, $y_p=a_{i_d}$, $y_{p+1}=b_{i_d}$, причем $d>t$, $d,t\in J_n$. Тогда

$$\varphi(\bar{y})-\varphi(y)=(c_p-c_q)(a_{i_t}-a_{i_d})+(c_{p+1}-c_{q+1})(b_{i_t}-b_{i_d}).$$

Согласно (6), (7) $(a_{i_t},b_{i_t}) \underset{c}{\prec} (a_{i_d},b_{i_d})$, поэтому $\varphi(\bar{y})-\varphi(y)\le 0$. Это противоречит предположению о достижении в точке $y\in EI_{nk}$ минимума функции $\varphi(x)$ задачи (4).

2. $p\in I_w''=\{2,4,...,w\}$. В этом случае можно указать такое $q\in I_w''$, $q>p$, что $y_q=x_p^*$, $y_{q-1}=x_{p-1}^*$. Проводя аналогичные случаю 1 построения, получим противоречие.

На основании противоречий, полученных в случаях 1 и 2, придем к справедливости утверждения теоремы.

Результаты теоремы 1 можно применить при решении следующей задачи: $\psi(y)=\|y-d\|^2\to\min$, $y\in EI_{nk}$, где $d=(d_1,d_2,...,d_w)\in R^w$. Для $\psi(y)$ при $y\in EI_{nk}$ справедливо:

$$\psi(y)=\|y-d\|^2=\sum_{i=1}^{w}y_i^2+\sum_{i=1}^{w}d_i^2-2\sum_{i=1}^{w}d_iy_i=$$

$$=\sum_{i=1}^{n}(a_i^2+b_i^2)+\sum_{i=1}^{w}d_i^2-2\sum_{i=1}^{w}d_iy_i,$$

$$\min_{y\in EI_{nk}}\psi(y)=\sum_{i=1}^{n}(a_i^2+b_i^2)+\sum_{i=1}^{w}d_i^2+2\min_{y\in EI_{nk}}\sum_{i=1}^{w}d_i^*y_i,$$

где $d_i^*=-d_i$, $i\in J_w$.

В соответствии с теоремой 1 минимум в правой части достигается в точке $y^*\in EI_{nk}$, где $y_{2j-1}^*=a_{i_j}$, $y_{2j}^*=b_{i_j}$, $i_j\neq i_t$ при $t\neq j$, $i_j\in J_n$, $j\in J_n$, а последовательность $\{i_1,i_2,...,i_n\}$ удовлетворяет соотношению $(a_{i_1},b_{i_1}) \underset{d^*}{\prec} (a_{i_2},b_{i_2}) \underset{d^*}{\prec} ... \underset{d^*}{\prec} (a_{i_n},b_{i_n})$, то есть

$$\min_{y\in EI_{nk}}\|y-d\|^2=\sum_{i=1}^{n}(a_i^2+b_i^2)+\sum_{i=1}^{w}d_i^2+2\sum_{i=1}^{w}d_i^*y_i^*. \quad (8)$$

Рассмотрим задачу оптимизации вида $\psi(y)=\|y-d\|^2\to\max$, $d=(d_1,d_2,...,d_w)\in R^w$, $y\in EI_{nk}$.

По аналогии с выводом соотношения (8) получим:

$$\max_{y\in EI_{nk}}\|y-d\|^2=\sum_{i=1}^{n}(a_i^2+b_i^2)+\sum_{i=1}^{w}d_i^2+\max_{y\in EI_{nk}}(-2\sum_{i=1}^{w}d_iy_i)=$$

$$=\sum_{i=1}^{n}(a_i^2+b_i^2)+\sum_{i=1}^{w}d_i^2-\min_{y\in EI_{nk}}(2\sum_{i=1}^{w}d_iy_i).$$

На основании теоремы 1 минимум в правой части равенства достигается в точке $\tilde{y}\in EI_{nk}$, где $\tilde{y}_{2j-1}=a_{i_j}$, $\tilde{y}_{2j}=b_{i_j}$, $i_j\neq i_t$ при $t\neq j$, $i_j\in J_n$, $j\in J_n$, а последовательность $\{i_1,i_2,...,i_n\}$ удовлетворяет соотношению $(a_{i_1},b_{i_1}) \underset{d}{\prec} (a_{i_2},b_{i_2}) \underset{d}{\prec} ... \underset{d}{\prec} (a_{i_n},b_{i_n})$, то есть

$$\max_{y\in EI_{nk}}\|y-d\|^2=\sum_{i=1}^{n}(a_i^2+b_i^2)+\sum_{i=1}^{w}d_i^2-2\sum_{i=1}^{w}d_i\tilde{y}_i. \quad (9)$$

Предположим теперь, что $d\in R^w$ является элементом множества EI_{nk}. Тогда справедливо соотношение $\sum_{i=1}^{w}d_i^2=\sum_{i=1}^{n}(a_i^2+b_i^2)$. Значит,

$$\max_{y\in EI_{nk}}\|y-d\|^2=2\sum_{i=1}^{n}(a_i^2+b_i^2)-2\sum_{i=1}^{w}d_i\tilde{y}_i=$$

$$=C-2\sum_{i=1}^{w}d_i\tilde{y}_i. \quad (10)$$

Рассмотрим функцию $l_d(y)=\sum_{i=1}^{w}d_iy_i$, где $d\in EI_{nk}$. Покажем, что минимальное значение $l_d(y)$ на множестве EI_{nk} не зависит от выбора вектора коэффи-

циентов $d \in EI_{nk}$ и равно $Q = \sum_{i=1}^{w} d_i \tilde{y}_i$, где $\tilde{y} \in EI_{nk}$ удовлетворяет соотношению (9). С этой целью введем функцию $l_{d^1}(y) = \sum_{i=1}^{w} d_i^1 y_i$, где $d^1 \in EI_{nk}$, $d^1 \neq d$. Поскольку d^1 можно получить из d путем конечного числа транспозиций, изменим порядок суммирования слагаемых в выражении для $l_{d^1}(y)$ таким образом, чтобы выполнялось условие $l_{d^1}(y) = \sum_{j=1}^{w} d_{i_j}^1 y_j$, $d_{i_j}^1 = d_j$, $j \in J_w$. Но тогда, поскольку $y \in EI_{nk}$ принимает значения всевозможных перестановок пар $(a_1,b_1),(a_2,b_2),...,(a_n,b_n)$, то

$$\min_{y \in EI_{nk}} l_d(y) = \min_{y \in EI_{nk}} l_{d^1}(y) = Q,$$

что и требовалось показать. Это значит, что для любой точки $d \in EI_{nk}$ расстояние до наиболее удаленной от нее точки множества EI_{nk} является величиной постоянной, ее значение определяется соотношением (10) и равно $D = \sqrt{C - 2Q}$, где $C = 2\sum_{i=1}^{n}(a_i^2 + b_i^2)$, а $Q = \sum_{i=1}^{w} d_i \tilde{y}_i$. Приведенные рассуждения доказывают справедливость следующего утверждения.

Теорема 2. Диаметр множества EI_{nk} равен $diam\, EI_{nk} = \sqrt{2\sum_{i=1}^{n}(a_i^2 + b_i^2) - 2\sum_{i=1}^{w} y_i^0 \tilde{y}_i}$, где y^0 — произвольный элемент множества EI_{nk}, а $\tilde{y} \in EI_{nk}$ удовлетворяет условиям $\tilde{y}_{2j-1} = a_{i_j}$, $\tilde{y}_{2j} = b_{i_j}$, $i_j \neq i_t$ при $t \neq j$, $i_j \in J_n$, $j \in J_n$, последовательность $\{i_1, i_2, ..., i_n\}$ удовлетворяет соотношению $(a_{i_1}, b_{i_1}) \underset{y^0}{\prec} (a_{i_2}, b_{i_2}) \underset{y^0}{\prec} ... \underset{y^0}{\prec} (a_{i_n}, b_{i_n})$.

Пусть теперь множество E в задаче (3) представляет собой множество EI_{nk}^m парных размещений из n элементов, k из которых различны, по m. Тогда задача (3) примет следующий вид:

$$\varphi(x) = \sum_{i=1}^{v} c_i x_i \to \min, \qquad (11)$$

где $x = (x_1, x_2, ..., x_v) \in EI_{nk}^m \subset R^v$, $c = (c_1, c_2, ..., c_v) \in R^v$, $v = 2m$.

Воспользуемся следующей особенностью множества EI_{nk}^m. Из комбинаторных свойств базового комбинаторного множества AI_{nk}^m, порожденного действительными числами $\alpha_1, \alpha_2, ..., \alpha_n$, следует [9], что оно является объединением множеств перестановок $P_{mk_i}^{(i)}$. При этом, каждое множество $P_{mk_i}^{(i)}$ порождается одним неупорядоченным m-элементным набором $\{\alpha_{j_1}, \alpha_{j_2}, ..., \alpha_{j_m}\} \subset \{\alpha_1, \alpha_2, ..., \alpha_n\}$. Количество M множеств перестановок $P_{mk_i}^{(i)}$ равно числу всевозможных неупорядоченных наборов из n элементов, k из которых различны, по m. Число M не превосходит числа сочетаний с повторениями из n элементов по m:

$$M \le \overline{\binom{n}{m}} = \frac{(n+m-1)!}{m!(n-1)!}.$$

Распространим указанные свойства множества AI_{nk}^m на множество парных размещений EI_{nk}^m. Представим множество EI_{nk}^m в виде объединения множеств парных перестановок $EI_{mk_i}^{(i)}$, $i \in J_M$. При этом каждое множество $EI_{mk_i}^{(i)} \subset R^v$ порождается неупорядоченным множеством

$$G^{(i)} = \{(a_{j_1}, b_{j_1}), (a_{j_2}, b_{j_2}), ..., (a_{j_m}, b_{j_m})\} \subset$$

$$\subset \{(a_1,b_1),(a_2,b_2),...,(a_n,b_n)\}. \qquad (12)$$

Количество M множеств перестановок $EI_{mk_i}^{(i)}$ не превышает числа сочетаний с повторениями из n элементов по m и зависит от количества k_i различных пар в множестве $\{(a_1,b_1),(a_2,b_2),...,(a_n,b_n)\}$. Таким образом,

$$EI_{mk}^m = \bigcup_{i=1}^{M} EI_{mk_i}^{(i)}, \qquad (13)$$

$$\forall \{j_1, j_2, ..., j_m\} \subset J_n \quad EI_{mk_i}^{(i)} = EI_{mk_i}^{(i)}(G^{(i)}) =$$

$$= EI_{mk_i}^{(i)}\left(\{(a_{j_1}, b_{j_1}), (a_{j_2}, b_{j_2}), ..., (a_{j_m}, b_{j_m})\}\right).$$

Рассмотрим множество $G^{(i)}$ вида (12). Зададим на множестве пар $(a_{j_t}, b_{j_t}) \in G^{(i)}, t \in J_m$, отношение порядка, аналогичное (5):

$$((a_{j_t}, b_{j_t}) \underset{c}{\prec} (a_{j_s}, b_{j_s})) \Leftrightarrow ((c_{2t-1} - c_{2s-1})(a_{j_t} - a_{j_s}) +$$

$$+ (c_{2t} - c_{2s})(b_{j_t} - b_{j_s}) \le 0). \qquad (14)$$

Упорядочим элементы множества $G^{(i)}$ в соответствии с отношением порядка (14). Пусть последовательность $\{j_{t_1}, j_{t_2}, ..., j_{t_m}\} \subset J_n$ такова, что

$$(a_{j_{t_1}}, b_{j_{t_1}}) \underset{c}{\prec} (a_{j_{t_2}}, b_{j_{t_2}}) \underset{c}{\prec} ... \underset{c}{\prec} (a_{j_{t_m}}, b_{j_{t_m}}). \qquad (15)$$

Теорема 3. Минимум линейной функции $\varphi(x)$ задачи (11) на множестве EI_{nk}^m, порожденном множествами $Z_i = \{a_i, b_i\}$, $i \in J_n$, достигается в точке $x^* = (x_1^*, x_2^*, ..., x_w^*) \in EI_{nk}^m$, такой, что

$$x^* = \arg\min_{i \in J_M} \varphi(y^i), \quad y^{(i)} = \arg \min_{y \in EI_{mk_i}^{(i)}(G^{(i)})} \varphi(y),$$

где $G^{(i)}$ — множество вида (12); $y^i_{2r-1}=a_{j_{t_r}}$; $y^i_{2r}=b_{j_{t_r}}$; $j_{t_r}\neq j_{t_s}$ при $s\neq r$, $j_{t_r}\in J_n$, $t_r\in J_m$, $r\in J_m$, а $\{j_{t_1},j_{t_2},...,j_{t_m}\}\subset J_n$ удовлетворяет (15).

Доказательство. Воспользуемся декомпозицией множества EI^m_{nk} вида (13). Применяя к каждому множеству $EI^{(i)}_{mk_l}$ теорему 1 и выбирая среди точек y^i минимума $\varphi(y)$ на $EI^{(i)}_{mk_l}$ точку с минимальным значением $\varphi(y)$, приходим к справедливости утверждения.

Используя теорему 3, получим решение задачи

$$\psi(y)=\|y-d\|^2\to\min,\quad y\in EI^m_{nk},$$

где $d=(d_1,d_2,...,\ d_v)\in R^v$, $v=2m$.

Проводя построения, аналогичные выполненным при доказательстве теоремы 3 и опираясь на теорему 1, получим:

$$y^0=\arg\min_{i\in J_M}\left\|y^i-d\right\|^2,\ y^{(i)}=\arg\min_{y\in EI^{(i)}_{mk_l}(G^{(i)})}\|y-d\|^2,$$

где $y^i_{2r-1}=a_{j_{t_r}}$, $y^i_{2r}=b_{j_{t_r}}$, $j_{t_r}\neq j_{t_s}$ при $s\neq r$, $j_{t_r}\in J_n$, $t_r\in J_m$, $r\in J_m$, а последовательность $\{j_{t_1},j_{t_2},...,j_{t_m}\}\subset J_n$ удовлетворяет (15).

Точное решение задачи оптимизации (11) описанным выше способом при больших n и m требует достаточно высоких вычислительных затрат, что не всегда является приемлемым. Рассмотрим способ получения оценки минимума линейной функции на множестве EI^m_{nk}.

Учтем, что в точках $x\in EI^m_{nk}$ координаты $x_i, i\in I'_v$, $I'_v=\{1,3,...,v-1\}$, принимают значения из множества $A=\{a_1,a_2,...,a_n\}$, а координаты x_i, $i\in I''_v$, $I''_v=\{2,4,...v\}$, – значения из множества $B=\{b_1,b_2,...,b_n\}$. Из способа построения множества EI^m_{nk} следует, что оно является подмножеством множества $EA^v_w\subset R^v$, которое определяется следующим образом. Для любого $x\in EA^v_w$ координаты $x_i, i\in I'_v$, принимают значения размещений из n элементов множества A по m, а координаты x_i, $i\in I''_v$ – значения размещений из n элементов множества B по m, то есть $EI^m_{nk}\subset EA^v_w, w=2n$, $v=2m$. Тогда выражение для $\varphi(x)$ можно записать в виде:

$$\varphi(x)=\sum_{i\in I'_v}c_ix_i+\sum_{i\in I''_v}c_ix_i=\sum_{i=1}^m\bar{c}_iy_i+\sum_{i=1}^m\tilde{c}_iz_i,$$

где $y\in R^m$, $z\in R^m$, $\bar{c}_i=c_{2i-1}$, $\tilde{c}_i=c_{2i}$, $i\in J_m$.

Поскольку $EI^m_{nk}\subset EA^v_w$, то

$$\min_{x\in EI^m_{nk}}\varphi(x)\geq\min_{x\in EA^v_w}\varphi(x)=\min_{y\in EI^m_{nk_a}(A)}\sum_{i=1}^m\bar{c}_iy_i+$$

$$+\min_{z\in EI^m_{nk_b}(B)}\sum_{i=1}^m\tilde{c}_iz_i,$$

где k_a,k_b — количества различных элементов во множествах A и B соответственно.

Решение задачи оптимизации линейной функции $\Phi(x)=\sum_{i=1}^m\alpha_ix_i$ на множестве E^m_{nk}, порожденном множеством $\beta=\{\beta_1,\beta_2,...,\beta_n\}$, приведено в [2,8]. Оно достигается в точке $x^*\in E^m_{nk}$, где $x^*_{p_i}=\beta_{l_i}$, $i\in J_s$, $x^*_{p_i}=\beta_{l_{n-r+i}}$, $i\in J_r$, $s+r=m$; $\beta_{l_1}\leq\beta_{l_2}\leq...\leq\beta_{l_n}$; $\alpha_{p_1}\geq\alpha_{p_2}\geq...\geq\alpha_{p_s}\geq0>\alpha_{p_{s+1}}\geq...\geq\alpha_{p_m}$.

Определяя таким образом минимумы в правой части неравенства, получим искомую оценку. Справедливо следующее утверждение:

Теорема 4. Пусть функция $\varphi(x)=\sum_{i=1}^v c_ix_i$ определена в точках множества $EI^m_{nk}\subset R^v$, порожденного множествами $Z_i=\{a_i,b_i\}$, $i\in J_n$. Тогда

$$\min_{x\in EI^m_{nk}}\varphi(x)\geq\sum_{i=1}^m\bar{c}_iy^*_i+\sum_{i=1}^m\tilde{c}_iz^*_i,\qquad(16)$$

где $\bar{c}_i=c_{2i-1}$; $\tilde{c}_i=c_{2i}$; $i\in J_m$; $y^*=(y^*_1,y^*_2,...,y^*_m)$; $y^*_{p_i}=a_{l_i}$, $i\in J_s$, $y^*_{p_i}=a_{l_{n-r+i}}$, $i\in J_r$, $s+r=m$, последовательности $\{l_1,l_2,...,l_n\}$ и $\{p_1,p_2,...,p_m\}$ таковы, что

$$a_{l_1}\leq a_{l_2}\leq...\leq a_{l_n},\qquad(17)$$

где $\bar{c}_{p_1}\geq\bar{c}_{p_2}\geq...\geq\bar{c}_{p_s}\geq0>\bar{c}_{p_{s+1}}\geq...\geq\bar{c}_{p_m}$, $z^*=(z^*_1,z^*_2,...,z^*_m)$, $z^*_{q_i}=b_{h_i}$, $i\in J_t$, $z^*_{q_i}=b_{h_{n-d+i}}$, $i\in J_d$, $t+d=m$. Последовательности $\{h_1,h_2,...,h_n\}$ и $\{q_1,q_2,...,q_m\}$ таковы, что

$$b_{h_1}\leq b_{h_2}\leq...\leq b_{h_n},\qquad(18)$$

$$\tilde{c}_{q_1}\geq\tilde{c}_{q_2}\geq...\geq\tilde{c}_{q_t}\geq0>\tilde{c}_{q_{t+1}}\geq...\geq\tilde{c}_{q_m}.$$

Используя оценку (16), получим оценку минимума $\psi(x)=\|x-d\|^2$, где $d\in R^v$, на множестве EI^m_{nk}. Имеем

$$\psi(x)=\|x-d\|^2=\sum_{i=1}^v x_i^2+\sum_{i=1}^v d_i^2-2\sum_{i=1}^v d_ix_i.$$

Поскольку для любого $x\in EI^m_{nk}$ координаты $x_i, i\in I'_v$, принимают значения из множества A, а координаты x_i, $i\in I''_v$, — значения из множества B, то справедливо неравенство:

$$\min_{x\in EI_{nk}^m}\sum_{i=1}^{v}x_i^2=\min_{x\in EI_{nk}^m}\left(\sum_{i\in I_v'}x_i^2+\sum_{i\in I_v''}x_i^2\right)\ge\sum_{i=1}^{m}a_{k_i}^2+\sum_{i=1}^{m}b_{s_i}^2,$$

где $\{k_1,k_2,...,k_n\}$ и $\{s_1,s_2,...,s_n\}$ таковы, что

$$|a_{k_1}|\le|a_{k_2}|\le...\le|a_{k_n}|,\ |b_{s_1}|\le|b_{s_2}|\le...\le|b_{s_n}|. \quad (19)$$

Тогда

$$\min_{x\in EI_{nk}^m}\psi(x)=\min_{x\in EI_{nk}^m}\|x-d\|^2\ge\sum_{i=1}^{m}\left(a_{k_i}^2+b_{s_i}^2\right)+$$

$$+\sum_{i=1}^{v}d_i^2+2\min_{x\in EI_{nk}^m}\sum_{i=1}^{v}d_i^*x_i,$$

где $d_i^*=-d_i$, $i\in J_v$.

Применяя результаты теоремы 4 в правой части неравенства, получаем следующую оценку:

$$\min_{x\in EI_{nk}^m}\|x-d\|^2\ge\sum_{i=1}^{m}\left(a_{k_i}^2+b_{s_i}^2\right)+\sum_{i=1}^{v}d_i^2+$$

$$+2\sum_{i=1}^{m}\left(\bar{d}_i^*y_i^0+\tilde{d}_i^*z_i^0\right), \quad (20)$$

где $\{k_1,k_2,...,k_n\}$ и $\{s_1,s_2,...,s_n\}$ удовлетворяют (19); $\bar{d}_i^*=d_{2i-1}^*$; $\tilde{d}_i^*=d_{2i}^*$; $i\in J_m$; $y^0=(y_1^0,y_2^0,...,y_m^0)$; $y_{p_i}^0=a_{l_i}$; $i\in J_s$, $y_{p_i}^0=a_{l_{n-r+i}}$; $i\in J_r$, $s+r=m$. Здесь последовательность $\{l_1,l_2,...,l_n\}$ удовлетворяет (17), а $\{p_1,p_2,...,p_m\}$ такова, что $\bar{d}_{p_1}^*\ge\bar{d}_{p_2}^*\ge...\ge\bar{d}_{p_s}^*\ge0>$ $>\bar{d}_{p_{s+1}}^*\ge...\ge\bar{d}_{p_m}^*$, $z^0=(z_1^0,z_2^0,...,z_m^0)$, $z_{q_i}^0=b_{h_i}$, $i\in J_t$, $z_{q_i}^0=b_{h_{n-d+i}}$, $i\in J_\theta$, $t+\theta=m$, где последовательность $\{h_1,h_2,...,h_n\}$ удовлетворяет (18), а $\{q_1,q_2,...,q_m\}$ такова, что $\tilde{d}_{q_1}^*\ge\tilde{d}_{q_2}^*\ge...\ge\tilde{d}_{q_s}^*\ge0>\tilde{d}_{q_{s+1}}^*\ge...\ge\tilde{d}_{q_m}^*$.

Рассмотрим теперь случай, когда множество E в задаче (3) представляет собой множество $\bar{S}I_n^m$ парных сочетаний с повторениями из n различных элементов по m. В этом случае задача (3) примет вид:

$$\varphi(x)=\sum_{i=1}^{v}c_ix_i\to\min, \quad (21)$$

где $x=(x_1,x_2,...,x_v)\in\bar{S}I_n^m\subset R^v$, $v=2m$,

$c=(c_1,c_2,...,c_v)\in R^v$.

Отметим, что координаты точек множества $\bar{S}I_n^m$ принимают значения всевозможных упорядоченных наборов пар (a_i,b_i), $a_i\in A$, $b_i\in B$, $i\in J_n$. При этом наборы отличаются друг от друга составом пар (a_i,b_i), а каждая пара (a_i,b_i), $i\in J_n$, может встречаться в наборе от нуля до m раз.

Представим функцию цели задачи (21) в следующем виде:

$$\varphi(x)=\sum_{i=1}^{2m}c_ix_i=\sum_{i=1}^{m}(c_{2i-1}x_{2i-1}+c_{2i}x_{2i})=$$

$$=\sum_{i=1}^{m}g_i(x_{2i-1},x_{2i}). \quad (22)$$

Поскольку функция $\varphi(x)$ вида (22) является сепарабельной, то для нее справедливо соотношение:

$$\min_{x\in\bar{S}I_n^m}\varphi(x)=\min_{x\in\bar{S}I_n^m}\sum_{i=1}^{m}g_i(x_{2i-1},x_{2i})\ge$$

$$\ge\sum_{i=1}^{m}\min_{\substack{(x_{2i-1},x_{2i})\in Z_j,\\ j\in J_n}}g_i(x_{2i-1},x_{2i}). \quad (23)$$

Нестрогое неравенство в соотношении (23) будет выполняться как равенство в случае, если для элементов множества $\bar{S}I_n^m$ допустить произвольный порядок следования значений (a_{i_j},b_{i_j}), $i_j\in J_n$, $j\in J_m$, координат x_{2i-1},x_{2i}, $i\in J_m$. Если для каждого элемента $x\in\bar{S}I_n^m$ потребовать выполнения бинарного отношения, определенного в (6), вида

$$x=(a_{i_1},b_{i_1},a_{i_2},b_{i_2},...,a_{i_m},b_{i_m})\in\bar{S}I_n^m,$$

$$(a_{i_1},b_{i_1})\prec(a_{i_2},b_{i_2})\prec...\prec(a_{i_m},b_{i_m}),$$

то в (23) справедливым будет неравенство $\ge$. Таким образом, справедливо утверждение:

Теорема 5. Пусть функция $\varphi(x)=\sum_{i=1}^{v}c_ix_i$ определена в точках множества $\bar{S}I_n^m\subset R^v$, $v=2m$, порожденного множествами $Z_i=\{a_i,b_i\}$, $i\in J_n$. Тогда

$$\min_{x\in\bar{S}I_n^m}\varphi(x)\ge\sum_{i=1}^{v}c_ix_i^0, \quad (24)$$

где $x_{2i-1}^0=a_i^0$, $x_{2i}^0=b_i^0$, $i\in J_m$,

$(a_i^0,b_i^0)=\arg\min_{(a_j,b_j)\in Z_j,j\in J_n}\left(c_{2i-1}a_j+c_{2i}b_j\right)$.

Построим теперь оценку минимума линейной функции $\varphi(x)$ задачи (21) на множестве $\bar{S}I_n^m$ на основе подхода, примененного при доказательстве теоремы 4. Учтем, что в точках $x\in\bar{S}I_n^m$ координаты $x_i, i\in I_v'$ принимают значения из множества A, а координаты x_i, $i\in I_v''$, – значения из множества B. Из комбинаторных свойств множества $\bar{S}I_n^m$ следует, что оно является подмножеством множества $E\bar{S}_w^v\subset R^v$, которое строится следующим образом. Для любого $x\in E\bar{S}_w^v$ координаты $x_i, i\in I_v'$, принимают значения сочетаний с повторениями из n элементов множества A по m, а координаты x_i, $i\in I_v''$ – значения сочетаний с повторениями из n элементов множества B по m, то есть $\bar{S}I_{nk}^m\subset E\bar{S}_w^v$, $w=2n$, $v=2m$.

Исходя из этого, запишем выражение для $\varphi(x)$ в виде:

$$\varphi(x)=\sum_{i\in I'_v} c_i x_i+\sum_{i\in I''_v} c_i x_i=\sum_{l=1}^{m}\overline{c}_l y_l+\sum_{l=1}^{m}\tilde{c}_l z_l,$$

где $y\in R^m$; $z\in R^m$; $\overline{c}_i=c_{2i-1}$; $\tilde{c}_i=c_{2l}$; $i\in J_m$. Поскольку $\overline{SI}^m_{nk}\subset E\overline{S}^v_w$, то

$$\min_{x\in\overline{SI}^m_{nk}}\varphi(x)\ge\min_{x\in E\overline{S}^v_w}\varphi(x)=\min_{y\in\overline{S}^m_n(A)}\sum_{i=1}^{m}\overline{c}_i y_i+\min_{z\in\overline{S}^m_n(B)}\sum_{i=1}^{m}\tilde{c}_i z_i.$$

Решение задачи оптимизации линейной функции $\Phi(x)=\sum_{i=1}^{m}\alpha_i x_i$ на множестве $\overline{S}^m_n$, порожденном множеством $\beta=\{\beta_1,\beta_2,...,\beta_n\}$, приведено в [2]. Оно достигается в точке $x^*=(x_1^*,x_2^*,...,x_m^*)\in\overline{S}^m_n$, где $x_i^*=\beta_{\min}$, $i\in J_s$, $x_i^*=\beta_{\max}$, $i\in J_m\setminus J_s$, $\beta_{\min},\beta_{\max}$ — соответственно минимальный и максимальный элементы множества β, а константа s определяется системой неравенств:

$$\sum_{j=1}^{t}\alpha_{s+1-j}\ge 0\quad\forall t\in J_s,\quad\sum_{j=1}^{t}\alpha_{s+j}\le 0\quad\forall t\in J_{k-s}.$$

Определяя на этой основе минимумы в правой части неравенства, получим искомую оценку. Таким образом, справедливо утверждение:

Теорема 6. Пусть функция $\varphi(x)=\sum_{i=1}^{v}c_i x_i$ определена в точках множества $\overline{SI}^m_n\subset R^v$, $v=2m$, порожденного множествами $Z_i=\{a_i,b_i\}$, $i\in J_n$. Тогда

$$\min_{x\in\overline{SI}^m_n}\varphi(x)\ge a_{\min}\sum_{i=1}^{s_1}\overline{c}_i+a_{\max}\sum_{i=s_1+1}^{m}\overline{c}_i+$$
$$+b_{\min}\sum_{i=1}^{s_2}\tilde{c}_i+b_{\max}\sum_{i=s_2+1}^{m}\tilde{c}_i,\qquad(25)$$

где $\overline{c}_i=c_{2i-1}$, $\tilde{c}_i=c_{2i}$, $i\in J_m$,

$$a_{\min}=\min_{i\in J_n}\{a_i\},\ a_{\max}=\max_{i\in J_n}\{a_i\},\ b_{\min}=\min_{i\in J_n}\{b_i\},$$
$$b_{\max}=\max_{i\in J_n}\{b_i\}.\qquad(26)$$

Константы s_1 и s_2 определяются системами неравенств:

$$\sum_{j=1}^{t}\overline{c}_{s_1+1-j}\ge 0\quad\forall t\in J_{s_1},\quad\sum_{j=1}^{t}\overline{c}_{s_1+j}\le 0\quad\forall t\in J_{k-s_1};$$

$$\sum_{j=1}^{t}\tilde{c}_{s_2+1-j}\ge 0\quad\forall t\in J_{s_2},\quad\sum_{j=1}^{t}\tilde{c}_{s_2+j}\le 0\quad\forall t\in J_{k-s_2}.$$

На основании результатов теорем 5 и 6 получим оценки минимума функции $\psi(x)=\|x-d\|^2$, где $d\in R^v$, на множестве $\overline{SI}^m_n$.

Поскольку для любого $x\in\overline{SI}^m_n$ координаты $x_i, i\in I'_v$, и x_i, $i\in I''_v$ принимают значения соответственно из множеств A и B, то в точках множества $\overline{SI}^m_n$ справедлива следующая оценка:

$$\min_{x\in\overline{SI}^m_n}\psi(x)=\min_{x\in\overline{SI}^m_n}\|x-d\|^2\ge m\left(a_0^2+b_0^2\right)+\sum_{i=1}^{v}d_i^2+$$
$$+2\min_{x\in\overline{SI}^m_n}\sum_{i=1}^{v}d_i^* x_i,\qquad(27)$$

где $a_0=\min_{i\in J_n}|a_i|$, $b_0=\min_{i\in J_n}|b_i|$, $d_i^*=-d_i$, $i\in J_v$.

Минимум в правой части неравенства можно оценить с помощью оценок (24) или (25). Таким образом, для функции $\psi(x)=\|x-d\|^2$ на множестве $\overline{SI}^m_n$ справедливы следующие оценки:

$$\min_{x\in\overline{SI}^m_n}\|x-d\|^2\ge m\left(a_0^2+b_0^2\right)+\sum_{i=1}^{v}d_i^2+2\sum_{i=1}^{v}d_i^* x_i^0,$$

где a_0,b_0 и d^* удовлетворяют (27), $x^0_{2i-1}=a_i^0$, $x^0_{2i}=b_i^0$, $(a_i^0,b_i^0)=\arg\min_{(a_j,b_j)\in Z_j,\,j\in J_n}\left(c_{2i-1}a_j+c_{2i}b_j\right)$, $i\in J_m$.

$$\min_{x\in\overline{SI}^m_n}\|x-d\|^2\ge m\left(a_0^2+b_0^2\right)+\sum_{i=1}^{v}d_i^2+a_{\min}\sum_{i=1}^{s_1}\overline{d}_i^*+$$
$$+a_{\max}\sum_{i=s_1+1}^{m}\overline{d}_i^*+b_{\min}\sum_{i=1}^{s_2}\tilde{d}_i^*+b_{\max}\sum_{i=s_2+1}^{m}\tilde{d}_i^*,$$

где a_0,b_0 и d^* удовлетворяют (27), $\overline{d}_i^*=d^*_{2i-1}$, $\tilde{d}_i^*=d^*_{2i}$, $i\in J_m$, $a_{\min}$, $a_{\max}$, $b_{\min}$, $b_{\max}$ определяются соотношением (26), константы s_1 и s_2 определяются системами неравенств:

$$\sum_{j=1}^{t}\overline{d}^*_{s_1+1-j}\ge 0\quad\forall t\in J_{s_1},\quad\sum_{j=1}^{t}\overline{d}^*_{s_1+j}\le 0\quad\forall t\in J_{k-s_1};$$

$$\sum_{j=1}^{t}\tilde{d}^*_{s_2+1-j}\ge 0\quad\forall t\in J_{s_2},\quad\sum_{j=1}^{t}\tilde{d}^*_{s_2+j}\le 0\quad\forall t\in J_{k-s_2}.$$

4. Применение результатов исследований при разработке интеллектуальной системы поддержки принятия решений

Полученные в статье результаты могут быть использованы при разработке интеллектуальной системы решения задач геометрического проектирования. Цель создания интеллектуальной системы – повышение эффективности решения задач геометрического проектирования на общей фундаментальной основе с использованием комплекса разработанных к настоящему времени математических моделей и методов. Функциями системы являются анализ постановки задачи геометрического проектирования, сформулированной на языке, близком к естественному, принятие решения о возможности построения (выбора) математической модели задачи и ее решения на основе оптимизационных методов геометрического проектирования.

В состав интеллектуальной системы могут быть включены следующие подсистемы:

1. Подсистема анализа исходной информации о задаче и принятия решения о возможности ее

формализации и выбора математической модели задачи в соответствующем классе моделей.

2. Подсистема построения конкретной реализации математической модели.

3. Подсистема оптимизации параметров модели.

4. Подсистема визуализации исходных данных и результатов решения задачи геометрического проектирования.

5. Банк математических моделей задач геометрического проектирования.

6. Банк методов оптимизации.

Проблемы анализа постановки задачи геометрического проектирования, сформулированной на языке, близком к естественному, рассматривались в [10]. Применение описанных там подходов позволяет формализовать постановку задачи геометрического проектирования. Теоретической основой такой формализации является основная задача геометрического проектирования, сформулированная в [1]. Формализация интервальных математических моделей задач этого класса может быть выполнена на базе основной задачи геометрического проектирования в интервальном виде [11].

Дальнейшая конкретизация математической модели задачи связана с моделированием ее области допустимых решений. Основу моделей областей допустимых решений задач геометрического проектирования с дискретными параметрами составляют классические комбинаторные множества [1, 2]. Математические модели, построенные на базе классических комбинаторных множеств, составят основу банка математических моделей задач геометрического проектирования. Исследованные свойства комбинаторных множеств и разработанные с их помощью оптимизационные методы позволяют сформировать банк методов решения экстремальных комбинаторных задач геометрического проектирования.

Многие задачи геометрического проектирования представляют собой экстремальные комбинаторные задачи на множествах перестановок, размещений и сочетаний [1, 2]. Учет погрешностей в моделях задач этого класса на основе методов интервального анализа приводит к необходимости описания и исследования интервальных комбинаторных множеств перестановок, размещений, сочетаний [12]. Отображение этих множеств в евклидово пространство, как один из способов их анализа [6], порождает необходимость описания и исследования комбинаторных множеств парных перестановок, парных размещений и парных сочетаний.

Результаты исследований, полученные в настоящей работе, могут быть использованы при формировании банка математических моделей задач геометрического проектирования, банка методов оптимизации, в подсистемах построения конкретной реализации математической модели и оптимизации параметров модели.

Результаты работы построенной таким образом интеллектуальной системы позволят получать решения новых классов задач геометрического проектирования с учетом погрешностей исходных данных.

Заключение

Приведенные в работе постановки и решения задач оптимизации могут быть использованы при разработке интеллектуальных систем поддержки принятия решений, ориентированных на задачи геометрического проектирования с учетом погрешностей исходных данных. В статье предложена и проанализирована концепция такой системы.

Результаты решения некоторых классов задач, сформулированные в виде теорем, можно применить при разработке методов оптимизации различных классов функций на композиционных образах комбинаторных множеств. При этом задачи оптимизации линейной функций и нормы разности можно рассматривать как вспомогательные задачи при реализации методов решения более сложных задач оптимизации на построенных композиционных образах комбинаторных множеств.

Список литературы: 1. *Стоян Ю.Г., Яковлев С.В.* Математические модели и оптимизационные методы геометрического проектирования. К.: Наук. думка, 1986. 268с. 2. *Стоян Ю.Г., Ємець О.О.* Теорія і методи евклідової комбінаторної оптимізації. К.:Інститут системних досліджень освіти, 1993. 188 с. 3. *Айгнер М.* Комбинаторная теория. М.: Мир, 1982. 558 с. 4. *Стоян Ю.Г., Гребенник И.В.* Специальные классы комбинаторных множеств в геометрическом проектировании // В кн.: Сб. тезисов докладов по материалам 10-й юбилейной междунар. конф. «Теория и техника передачи, приема и обработки информации». Харьков-Туапсе 2004. С. 253-254. 5. *Kaucher E.* Interval Analysis in the Extended Interval Space IR // Comp. Suppl. 1980. № 2. P. 33–49. 6. *Гребенник И.В., Романова Т.Е.* Отображение интервальных комбинаторных множеств в евклидово пространство // Проблемы машиностроения. 2002. 5, №2. С. 87–91. 7. *Стоян Ю.Г., Яковлев С.В.* Свойства выпуклых функций на перестановочном многограннике // ДАН УССР, Сер. А. 1988. №3. С. 238-240. 8. *Яковлев С.В., Гребенник И.В.* О некоторых классах задач оптимизации на множествах размещений и их свойствах// Математика. 1991. №11. С. 74-86.(Изв. высш. учеб. заведений). 9. *Стоян Ю.Г., Гребенник И.В., Емец О.А.* Комбинаторные множества размещений и их свойства. Х., 1990. 38с. (Препринт АН УССР/Ин-т пробл. машиностроения, 342). 10. *Ещенко В.Г., Романова Т.Е.* Математическое обеспечение САПР задач геометрического проектирования // Программные средства машиностроения: Сб. науч. тр. АН УССР. Ин-т кибернетики. К., 1990. 81 с. 11. *Гребенник И.В., Евсеева Л.Г., Романова Т.Е.* Основная оптимизационная задача геометрического проектирования в интервальном виде / / Радиоэлектроника. Информатика. Управление. – 2004, № 2. С. 68-72. 12. *Гребенник И.В.* Интервальные модели комбинаторной оптимизации квазилинейных функций в пространстве $I_s^n R$ // Доповіді НАН України. – 2004. № 9. С. 60-64.

Поступила в редколлегию 17.11.2004

УДК 519.87

Альхалайбех Айман, О.Ф. Михаль

МОДЕЛИРОВАНИЕ НА СЕТЯХ ПЕТРИ ИНФОРМАЦИОННОГО ОБМЕНА МЕЖДУ СИСТЕМАМИ С ОГРАНИЧЕННЫМ РЕСУРСОМ ГОТОВНОСТИ

Введение

В бионике предметом исследований являются объекты живой природы, рассматриваемые в аспекте поиска аналогов для разработки искусственных *технических систем* (ТС) с характеристиками, приближающимися к характеристикам живых систем [1]. Объектами живой природы являются отдельные организмы (особи) и их сообщества. При этом системные различия между особями и сообществами в ряде случаем бывают не слишком значительными (простейшие, грибы, коллективные насекомые). Аналогично, в ТС часто условной является разбивка на системы и подсистемы, блоки и субблоки. Вследствие этого аналогия между системами в живой природе и технике простирается на область *информационного обмена* (ИО), производимого с целью согласования действий совокупностей систем (подсистем) — *сообществ* в живой природе, *распределенных* ТС в технике. Согласованность действий является ключевым моментом, определяющим целевую функцию поведения (выживания, функционирования) систем. Но качество ИО (число прямых и обратных информационных связей [2] между двумя ТС или ТС и объектом живой природы — человеком-оператором) лимитировано возможностями каналов ИО между системами и *ограниченными ресурсами готовности* (ОРГ) систем по обработке информации.

Для проектирования ТС представляют интерес оценки роста сложности при увеличении числа связей ИО с учетом ОРГ. Целью настоящей работы является получение подобных оценок на моделях. В качестве инструмента моделирования выбран аппарат *сетей Петри* (СП) [3]. При этом решаются задачи апробации методов моделирования, разработки соответствующих классов моделей, сопоставления и интерпретации результатов.

1. Сетевая парадигма

В распределенных ТС отдельные объекты, связи и взаимодействия удобно представлять в рамках сетевой парадигмы. Система предстает в виде графа. Вершины графа интерпретируются как функционально завершенные блоки системы либо детерминированные состояния блоков (события); ребра графа — как связи между блоками либо переходы между состояниями (связи между событиями).

Аппарат СП интересен тем, что интегрирует в себе блочно-функциональное и событийное описания системы. Основным элементом СП является двудольный ориентированный граф G, содержащий два множества вершин P: $\{p_1, p_2, ..., p_n\}$ (вектор мест) и T:$\{t_1, t_2, ..., t_m\}$ (вектор переходов) и два множества (матрицы) ориентированных ребер, являющихся связями места-переходы M_{PT} и переходы-места M_{TP}.

Двудольный граф $G(P, T, M_{PT}, M_{TP})$ дополнен *начальной разметкой* вектора вершин P и *правилами смены разметки*, включающими определенную *дисциплину* рассмотрения мест и переходов и изменение состояния некоторых из них в соответствии с фиксированными *логическими правилами*. Варьированием дисциплины и (или) логических правил порождаются разные варианты правил смены разметки. Варьированием правил смены разметки и наложением дополнительных ограничительных условий на параметры P, T, M_{PT} и M_{TP} порождаются разные подвиды СП. Далее для моделирования ИО между системами используется подвид СП *с приоритетами* [4], предполагающий рассмотрение переходов (их зарядку и разрядку) в фиксированное порядке.

2. Моделирование распределенных систем

Распределенные системы обработки информации получили широкое распространение в различных технических областях. ИО в системах предполагает совместное использование ограниченных ресурсов ИО и многоуровневую адаптивную организацию протоколов обмена, что требует затраты ресурсов.

Целесообразно выделить два вида ресурсов ИО: *ресурс каналов обмена* (РКО) и *ресурс готовности систем* (РГС) к обмену. РКО определяется количеством и качеством каналов, по которым системы могут обмениваться информацией. РГС обусловлен производительностью системы, способностью перерабатывать поступающую информацию и генерировать её для передачи. В технической реализации оба этих ресурса являются ограниченными в соответствии с номинальными условиями функционирования системы. При низкой активности ИО ресурсы простаивают, при повышенной (сверхпороговой) — их не хватает. В связи с этим разработка распределенных ТС с избыточными ресурсами ИО бывает экономически нецелесообразна, а при недостатке ресурсов система может не обеспечить надлежащего функционирования. В подобной ситуации целесообразен принцип адаптивности, предполагающий (в рассматриваемом варианте) единообразность (внутрисистемную стандартизованность) РГС и РКО и возможность маневра РГС

и РКО в соответствии с текущим состоянием загруженности системы.

При рассмотрении подобных вопросов целесообразно моделирование. Его назначение – демонстрация (апробация) структурных (схемных) особенностей реализации и исследование роста сложности системы по мере увеличения уровня адаптивности (интеллектуальности) обеспечиваемого информационного обмена. На уровне концепции модель рассмотрена ранее [2]. В данной работе представлена реализация модели средствами СП. Рассмотрен *симплексный* вариант обмена: система B является источником информации, система A – приемником: $(B \to A)$. В *дуплексном* варианте при симметричном прозрачном использовании каналов связи в двух направлениях (упрощенный случай) модель зеркально симметрично удваивается: $(B \leftrightarrow A) \equiv (B \to A)$ И $(A \to B)$

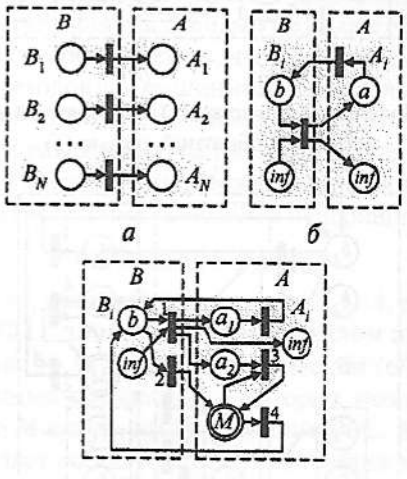


Рис. 1. Модели ИО без обратных связей (*а*) (показано N каналов), с одной (*б*) и с двумя (*в*) (показан один канал) обратными связями

На рис. 1 и 2 представлено четыре последовательно усложняющихся варианта моделей взаимодействия систем A и B. В варианте, показанном на рис. 1 *а*, учитывается только РКО: информация из B поступает в A без проверки готовности к приему. Варианты, показанные на рис. 1 *б*, *в* и рис. 2, включают проверку наличия РГС. Рассмотрим модели более подробно.

В 1-м варианте (рис. 1 *а*)при наличии метки в месте B_i связанный с ним переход срабатывает (заряжается и разряжается), и метка переходит в место A_i. Если ранее в месте A_i уже находилась метка, что соответствует ситуации отсутствия РГС, модель будет реагировать в соответствии с принятой дисциплиной работы СП. В частности, если используется дисциплина с поглощением избыточных меток [4], в месте A_i будет находиться снова одна метка, что интерпретируется как утрата (пропуск) информации.

Во 2-м варианте (рис. 1 *б*) дополнительно введены места b и a. Назначение обратной связи $a \to b$ — сигнализация о наличии РГС. Если в b имеется метка, то при появлении метки в *inf* переход в B_i срабатывает, и метки поступают в места a и *inf* в A_i. Далее срабатывает переход в A_i и метка поступает в b. Одновременно метка из места *inf* в A_i поступает на обработку в последующую часть системы (не показана). При этом высвобождается РГС.

В 1-м и 2-м вариантах активной является передающая сторона B_i; РГС на приемной стороне A_i не централизован. В 3-м варианте (рис. 1 *в*) РГС централизован и активность по получению информации проявляет приемная сторона A_i. Централизация РГС осуществляется местом M, общим для N информационных каналов (условно показан только один), по которым система B выставляет информацию для передачи. Система A обладает MРГС для приема информации (метки в месте M). Представляет интерес случай $N > M$. При этом A выделяет ресурс $(M \to b)$ каналу по запросу $(b \to M)$. Для моделирования применена СП с приоритетами. Переходы 1 и 3 имеют более высокий приоритет, чем переходы 2 и 4. При наличии информации *inf* на передающей стороне срабатывает переход 1, и информация передается $(inf \to inf)$. При этом метки передаются также в места a_1 и a_2. Далее готовность канала восстанавливается: высылается квитанция $(a_1 \to b)$, а метка в a_2 используется для блокировки повторного выделения РГС. Таким образом, данной схемой реализуется два механизма регулирования передачи информации: выделение РГС и захват канала продолжительным потоком информации. Если принимающая система имеет свободный РГС, она выделяет его i-й передающей системе B_i. Если информация для передачи отсутствует, ресурс возвращается $(b \to M)$. При наличии информации передающая система захватывает канал и удерживает его посредством механизма квитирования до тех пор, пока вся информация будет передана. После этого РГС возвращается системе A.

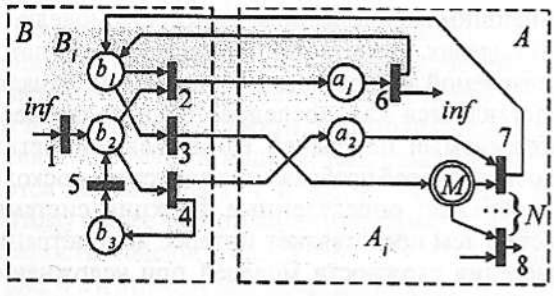


Рис. 2. Модель ИО с выделением РГС по запросу

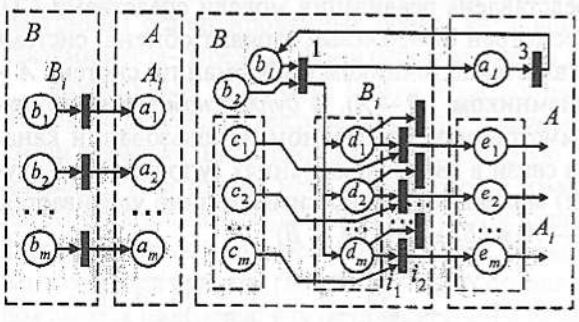
Схема ИО, представленная на рис. 2 (4-й вариант), напоминает вариант на рис. 1 *в*. Различие состоит в распределении активности при реализации процедуры ИО. Согласно схеме (рис. 2) РГС выделяется по запросам передающей стороны. При этом, как и в 3-м варианте (рис. 1 *в*), канал удерживается до завершения передачи всей информации.

Переходы 2, 3 и 4 приоритетно упорядочены. Если канал ранее не был выделен (метка в b_1 отсутствует), то при появлении информации в передающей системе (место b_2 заряжается через переход 1) срабатывает переход 4 и метка поступает в a_2. Это интерпретируется как запрос на выделение канала. Кроме того, через 2 цикла работы СП состояние b_2 восстанавливается ($4 \to b_3 \to 5 \to b_2$), и запрос повторяется. Состояние a_2 (наличие запроса) не изменяется до тех пор, пока в M появится метка (свободный РГС). В исходном состоянии работы системы (при запуске системы) в M имеется M меток. Если $N>M$, через некоторое время возможна ситуация занятости M каналов из N (отсутствие свободного РГС). При появлении метки в M (высвобождении РГС), срабатывает переход 7 и в b_1 поступает метка, обозначающая выделение канала. При этом запрос реализован и порция информации проходит по каналу ($b_2 \to 2 \to a_1 \to 6$). Переданная информация квитируется ($6 \to b_1$), что подтверждает готовность канала (выделенность РГС). Передается следующая порция информации. Если информация передана вся (метка в b_2 отсутствует), РГС возвращается ($b_1 \to 3 \to M$). Таким образом, представленной схемой также реализуется централизация РГС (на рис. 2 переходами 7 и 8, показано подключение места M к N информационным каналам) и захват канала до полной передачи информации по нему.

Для 3-го и 4-го вариантов (рис. 1 *в* и 2) конкуренция между каналами при $N>M$ регулируется приоритетностью, устанавливаемой порядком опроса каналов. Соответствующие механизмы на рисунках не показаны. Метки в модели не различаются, хотя и несут разную смысловую нагрузку в зависимости от контекста. Таким образом, при данном уровне детализации отображается только аспект функционирования системы, что является ограничением модели. Ограничением представленных схем является также отсутствие процедуры прерывания менее приоритетного канала более приоритетным. Обеспечение прерываний достижимо дальнейшим наращиванием сложности модели.

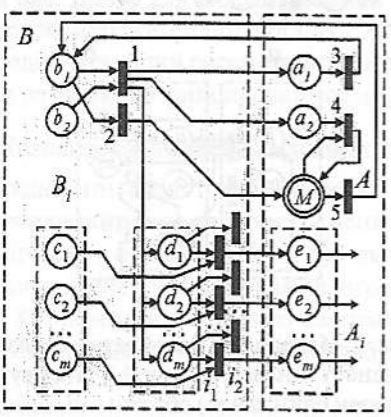
В моделях, представленных на рис. 1 и 2, поток передаваемой информации в каждом из каналов представляется как последовательный процесс, изображаемый передачей единственной метки. Подобный способ изображения допустим, поскольку отображает определенные функции системы. Вместе с тем представляет интерес демонстрация нарастания сложности моделей при усложнении процесса передачи информации, то есть при увеличении числа обратных связей. На рис. 3 и 4 представлена линия более сложных моделей: с m-шинным каналом передачи информации. Модели на рис. 3 *а*, *б* и рис. 4 *а*, *б* являются аналогами моделей на рис. 1 *а*, *б*, *в* и рис. 2 соответственно. Проведем детальное сопоставление.

Модель ИО без обратных связей в m-шинном варианте (рис. 3 *а*) структурно полностью аналогична одношинному варианту (рис. 1 *а*). В остальных моделях имеется следующее структурное отличие: введены три линейки мест (c_1-c_m, d_1-d_m и e_1-e_m) и две линейки переходов (i_1, i_2) для управления процессом передачи информации. В остальном преемственность моделей второй линии сохранена, в том числе и в системе обозначений. Работа моделей второй линии аналогична работе моделей первой линии. Детально рассмотрим работу на примере самой сложной из моделей (рис. 4*б*).

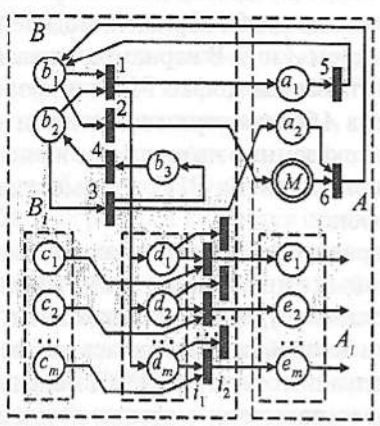


а *б*

Рис. 3. Модели m-шинного ИО без обратной связи (*а*) и с одной обратной связью (*б*)



а



б

Рис. 4. Модели m-шинного ИО с двумя обратными связями и с инициацией обмена приемной (*а*) и передающей (*б*) сторонами

На вход канала ИО подается передаваемая информация (линейка мест c_1-c_m) и управляющий

(сигнальный) бит b_2. Наличие метки в месте b_2 является признаком наличия информации в регистре c_1-c_m. Комбинация меток в c_1-c_m является передаваемой информацией, которая может интерпретироваться, в частности, как число в двоичной системе счисления. Работа подсети СП, включающей места b_1, b_2, b_3, a_1, a_2 и переходы 1-6, аналогична показанной на рис. 2. При наличии метки в b_1 (РГС выделен) срабатывает переход 1. При этом выдается метка на a_1 (для выдачи квитанции) и на линейку мест d_1-d_m. Линейка переходов i_1 имеет более высокий приоритет, чем i_2, поэтому сначала срабатывают те из переходов в i_1, которым соответствуют метки в c_1-c_m, а затем оставшиеся в d_1-d_m метки сбрасываются в переходы i_2. В итоге содержимое c_1-c_m поступит в e_1-e_m, а линейка d_1-d_m будет подготовлена (очищена) для последующей работы. Дальнейшее прохождение информации из e_1-e_m, как и предшествующее поступление информации в c_1-c_m, рассматриваемыми схемами не иллюстрируется.

Следует особо отметить, что с введением линеек мест и переходов для m-шинного параллельного перемещения информации число тактов функционирования управляющих частей моделей не изменилась. Этим косвенно подтверждается функциональная репрезентативность моделей первой линии.

3. Стековая структура

Модели, представленные на рис. 1–4, реализованы на СП с приоритетами и правилом поглощения меток [4], за исключением мест M (обозначены двойными кружками), в которых может находиться до M меток, изображающих РГС. Место M представляет собой пул для накопления и хранения текущих освободившихся ресурсов РГС. Для простоты в нашей системе обозначений число меток, изображающих РГС, совпадает с обозначением пула для их хранения.

Для сравнения оценок сложности моделей было бы удобно, чтобы вся модель была построена из *одинаковых* элементов СП. Включение элементов, реализующих иные принципы, осложняет оценку сложности и сравнение оценок. В связи с этим представляет интерес реализация пула M также средствами СП с приоритетами и правилом поглощения меток. На рис. 5 показан один из возможных вариантов. Пул M представлен в виде стека при значениях параметров $N=3$, $M=2$. Места q_1, q_2, q_3 и переходы 1, 2, 3 являются входами стека, места l_1, l_2, l_3, k_1, k_2, k_3 и переходы 5, 6, 7 — его выходами. Элементы, реализующие входы и выходы стека, тождественны соответствующим элементам модели, предназначенным для подачи либо изъятия меток из пула M. Например для модели, показанной на рис. 4 *а*, справедливо следующее: $q_1 \equiv b_1$; переход 1 (рис. 5) тождественен переходу 2 (рис. 4 *а*); $l_1 \equiv a_2$; переход 5 (рис. 5) тождественен переходу 4 (рис. 4 *а*); k_1 отсутствует, его функции выполняет непосредственно пул M. Собственно стеком является звено, выделенное на рис. 5 пунктирной рамкой (места g_1, g_2, h_1, h_2 и переход 4). Число входов и выходов стека наращивается параллельным подключением мест q, l, k и соответствующих переходов. Длина стека наращивается парами g, h с подключением каждой пары через один переход, аналогично выделенному звену (рис. 5).

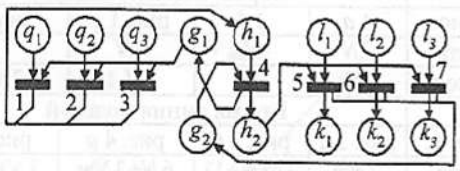


Рис. 5. Стековая структура взамен места M в моделях ИО

Работа стека. Метки поступают в вершину стека h_1 через входы и выдаются из основания стека h_2 через выходы. Стек, в силу своей структуры, стремится к *нормализованному состоянию*. Это значит, что, находясь внутри стека, метки перемещаются, стремясь занять положения, ближайшие к основанию. Для этого места стека q и h имеют самый высокий приоритет в модели. Вследствие приоритета стека всякий раз при поступлении метки в вершину либо удалении её из основания, сначала происходит его нормализация. В стеке места упорядочены попарно (q, h) для отображения состояний наличия-отсутствия РГС в i-м слое стека, согласно схеме:

Комбинации ($q_i=1$, $h_i=1$) и ($q_i=0$, $h_i=0$) являются недопустимыми и при корректной начальной разметке модели исключены логикой работы стека.

$q_i=1$; $h_i=0$; — ресурса нет;
$q_i=0$; $h_i=1$; — ресурс есть.

4. Оценка сложности систем

Все рассмотренные модели ИО систем (рис. 1–4) с пулом РГО, реализованным в варианте стека, показанном на рис. 5, и верифицированы с использованием программного обеспечения, аналогичного описанному в [4].

На основе представленных моделей сделаем оценки роста сложности реализации систем с повышением уровня их организации. Оценку проводим по числу ячеек памяти, необходимых для хранения информации модели. Как отмечалось, требуются два вектора P, T и две матрицы M_{PT}, M_{TP}. Поэтому общее число элементов, характеризующее сложность модели, будет $S=P+T+2PT$. Модели на рис. 4 *а* и *б* имеют максимальную ресурсоемкость реализации при максимальном объеме стека, то есть при $M=N$. Этот случай является наиболее целесообразным для проведения оценок. С его учетом исходные данные для оценивания представлены в табл. 1.

Ресурсоемкость моделей S (объемы памяти, требуемые для реализации) растет по *каждому* из параметров N и m по квадратичному закону. Как результат, для моделей первого ряда имеет место квад-

ратичный рост сложности, а для моделей второго ряда при совместном росте двух параметров – рост в соответствии с полиномом 4-й степени.

Таблица 1

Ресурсоемкость реализации моделей ИО. Исходные данные

	Первая линия моделей			
Рисунок	1 *а*	1 *б*	рис. 1 *в*	рис. 2
Места	$2N$	$4N$	$7N$	$7N$
Переходы	N	$2N$	$6N$-1	$7N$-1
	Вторая линия моделей			
Рисунок	рис. 3 *а*	рис. 3 *б*	рис. 4 *а*	рис. 4 *б*
Места	$2Nm$	$3N(m+1)$	$6N+3Nm$	$7N+3Nm$
Переходы	Nm	$2N(m+1)$	$6N+3Nm$-1	$7N+3Nm$-1

Иначе выглядит рост S с повышением сложности структурной организации. На основе построенных моделей эту зависимость можно проследить только по четырем точкам. Дальнейшее исследование требует построения новых моделей для структурно более сложных случаев. На графиках (рис. 6) представлены некоторые результаты. По оси абсцисс в условных (нечетко обусловленных) единицах представлена структурная сложность K модели, аналогично рассмотренному в [2]: 1 — отсутствие обратных связей; 2 — одна обратная связь: подтверждение готовности; 3 — две обратные связи: наличие информации и выделение РГС; 4 — прямая и обратная связь: запрос РГС и выдача РГС. Ось ординат отображает ресурсоемкость S реализации модели. Из четырёх семейств графиков первое (рис. 6 *а*) относится к моделям первой линии, остальные — к моделям второй линии. В каждом из семейств графики построены для множества значений N: {2, 4, 6, 8, 10). Упорядоченность графиков совпадает с возрастанием параметра N; график при $N = 10$ максимально удален от оси абсцисс. Графики на рис. 6 *б*, *в* и *г* построены для множества значений m:{ 8, 16, 32} соответственно.

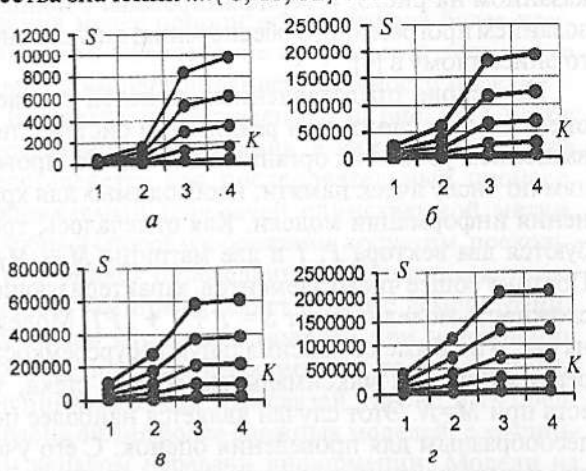


Рис. 6. Рост ресурсоемкости модели с повышением структурной сложности системы

Основываясь на полученных результатах, можно сделать ряд выводов качественного характера. Для обеих линий моделей наблюдается качественное соответствие зависимостей. Вместе с тем, с ростом N в случае, показанном на рис. 6 *а*, график просто масштабируется; в остальных случаях — изменяется более сложным образом: «полка» (3, 4) масштабируется «замедленно» в сравнении с участком (1, 2, 3). Данный эффект закономерен. В моделях второй линии с ростом m «полка» (3, 4) «линеаризуется» за счет преобладающего расходования ресурсов модели на создание шин.

С ростом m темп нарастания, вызванный N, «сглаживается». Графики на рис. 6 *г* расположены с «более равномерным шагом», чем на рис. 6 *а*. Последнее закономерно, поскольку в пределах одного канала рост ресурсов модели с увеличением числа шин m – линейный. При соизмеримых значениях N и m (рис. 6 *а* и *б*, верхние графики) общие тенденции поведения графиков (в пределах пучка) — «сближаются».

Наконец, наиболее общий вывод. Участки (1, 2) и (2, 3) соответствуют увеличению количества структурных связей; участок (3, 4) – изменению характера и функций связей без изменения их количества. Таким образом, ресурсоемкость модели, а следовательно, и сложность моделируемой системы, радикально нарастает с ростом числа структурных связей. Функциональная реструктуризация при сохранении прежнего числа структурных связей существенно менее сказывается на ресурсоемкости системы. Последний тезис возможно требует более широкого осмысления. Материал может представить интерес не только в связи с разработкой ТС, но и в контексте реструктуризации экономических и административно-управленческих систем.

Заключение

Исследован рост сложности систем, реализующих информационный обмен при ограниченном ресурсе готовности. Исследование проведено на моделях с использованием аппарата сетей Петри. Разработанные принципы моделирования и иерархические линии моделей отличаются простотой и прозрачностью в сравнении с моделированием на языках высокого уровня. Практическая ценность полученных результатов предположительно выходит за рамки чисто технических приложений. Результаты могут быть переосмыслены в контексте реструктуризации экономических и административных управленческих систем.

Список литературы: 1. *Литинецкий И.Б.* Бионика. В кн. Энциклопедия кибернетики. Т. 1. К.: УРЭ, 1973. С. 166 - 168. 2. *Альхалайбех Айман, Михаль О.Ф., Руденко О.Г.* 3-вено "техническая система – человек оператор", как модель информационного взаимодействия интеллектуальных подсистем с разной производительностью // Восточно-Европейский журнал передовых технологий. № 6 (6). 2003. С. 18-21. 3. *Котов В.Е.* Сети Петри. М.: Наука, 1984. 158 с. 4. *Михаль О.Ф. Руденко О.Г.* Моделирование на сетях Петри виртуального вычислительного устройства для исследования эффективности локально-параллельных алгоритмов // Управляющие системы и машины. 2003. №3. С. 18-28.

Поступила в редколлегию 28.10.2004

УДК 681.513

Н.А. Тесленко

АЛГОРИТМ ОБУЧЕНИЯ АВТОАССОЦИАТИВНОЙ ИСКУССТВЕННОЙ МНОГОСЛОЙНОЙ НЕЙРОННОЙ СЕТИ

Введение

В настоящее время искусственные нейронные сети получили широкое распространение в задачах интеллектуального анализа данных, благодаря своим универсальным аппроксимирующим свойствам [1, 2]. Наряду с задачами, так или иначе связанными с проблемой аппроксимации (эмуляция, управление, распознавание образов, прогнозирование и т. п.), достаточно часто приходится сталкиваться с проблемами сжатия информации (сокращение размерности входного пространства признаков), постоянно возникающими при обработке больших массивов информации.

Постановка задачи

Для исследования и сжатия массивов многомерных данных, характеризующихся линейной зависимостью признаков исходного многомерного пространства, наиболее часто применяется техника факторного анализа (главных компонент), позволяющая путем выявления доминирующих латентных признаков понизить размерность исходной задачи. Для решения той же проблемы в существенно нелинейных ситуациях в [3] было предложено использовать архитектуры типа многослойного персептрона специального вида с сигмоидальными активационными функциями, обучаемые с помощью процедур обратного распространения ошибок. Основным недостатком этого подхода является низкая скорость обучения, присущая всем многослойным нейронным сетям с нелинейными активационными функциями типа логистической кривой или гиперболического тангенса.

Для решения задачи понижения размерности исходного пространства признаков (сжатия информации) в данной работе предлагается использовать многослойную автоассоциативную искусственную нейронную сеть прямого распространения (bottleneck), построенную на формальных нейронах с полиномиальной функцией активации вида [4]:

$$\psi_j(u_j) = 1{,}5\left(u_j(k) - \frac{u_j^3}{3}(k)\right), \qquad (1)$$

где u_j — внутренний сигнал активации j-го формального нейрона; $k = 1, 2, 3, \ldots$ — текущее дискретное время; $-1 < \psi_j(u_j) < 1$; $-1 < u_j < 1$. Вид функции (1) приведен на рис. 1.

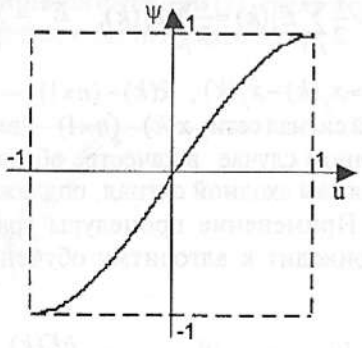


Рис. 1. Активационная функция формального нейрона

Данная активационная функция обладает стандартными свойствами сигмоидальных функций, то есть является монотонной, возрастающей, ограниченной и имеет отличную от нуля производную на всей области определения.

Как было показано в [4–6], алгоритм обучения формального нейрона, построенный на основе активационной функции вида (1), позволяет существенно повысить скорость настройки синаптических весов формального нейрона, схема которого приведена на рис. 2.

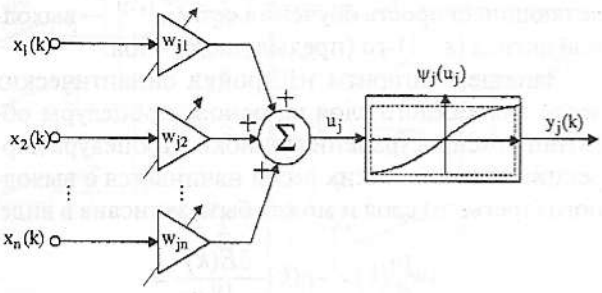


Рис. 2. Схема формального нейрона с полиномиальной функцией активации

Предлагаемая нейронная сеть состоит из трех слоев, образованных нейронами с полиномиальными функциями активации. Число нейронов первого скрытого и выходного слоев совпадает с размерностью входного пространства и принимается равным n. Число нейронов второго скрытого слоя принимается m<n и определяется степенью сжатия информации. Выходной сигнал каждого слоя является входным сигналом для последующего слоя.

Требуется разработать алгоритм обучения данной архитектуры.

Архитектура сети и алгоритм обучения

Процесс обучения нейронной сети сводится к настройке матриц синаптических весов $W^{[s]}$, $s = 1, 2, 3$, путем минимизации принятой целевой функции, в качестве которой используется выражение:

$$E(k) = \frac{1}{2}\sum_{j=1}^{n} E_j(k) = \frac{1}{2}\sum_{j=1}^{n} e_j^2(k), \quad E^k = \sum_k E(k),$$

где $e_j(k) = x_j(k) - \hat{x}_j(k)$, $\hat{x}(k) - (n\times 1)$ — выходной векторный сигнал сети, $x(k) - (n\times 1)$ — входной вектор. В данном случае в качестве обучающего используется сам входной сигнал, подлежащий компрессии. Применение процедуры градиентного спуска приводит к алгоритму обучения общего вида:

$$w_{ji}^{[s]}(k+1) = w_{ji}^{[s]}(k) - \eta(k)\frac{\partial E(k)}{\partial w_{ji}^{[s]}(k)} =$$

$$= w_{ji}^{[s]}(k) - \eta(k)\frac{\partial E(k)}{\partial u_j^{[s]}(k)}\frac{\partial u_j^{[s]}(k)}{\partial w_{ji}^{s}(k)} =$$

$$= w_{ji}^{[s]}(k) + \eta(k)\delta_j^{[s]}(k)o_i^{[s-1]}(k),$$

$$\Delta w_{ji}^{[s]}(k) = -\eta(k)\frac{\partial E(k)}{\partial w_{ji}^{[s]}(k)},$$

где $\delta_j^{[s]}(k) = -\dfrac{\partial E(k)}{\partial u_j^{[s]}(k)}$ — локальная ошибка j-го нейрона s-го слоя; $\eta(k)$ — скалярный параметр, определяющий скорость обучения сети; $o_i^{[s-1]}$ — выходной сигнал $(s-1)$-го (предыдущего) слоя.

Запишем алгоритм настройки синаптических весов для каждого слоя на основе процедуры обратного распространения ошибки. Процедура коррекции синаптических весов начинается с выходного (третьего) слоя и может быть записана в виде:

$$\Delta w_{ji}^{[3]}(k) = -\eta(k)\frac{\partial E(k)}{\partial w_{ji}^{[3]}(k)} =$$

$$= -\eta(k)\frac{\partial E(k)}{\partial u_j^{[3]}(k)}\frac{\partial u_j^{[3]}(k)}{\partial w_{ji}^{[3]}(k)} = \eta(k)\delta_j^{[3]}(k)o_i^{[2]}(k),$$

при этом локальная ошибка третьего выходного слоя с учетом (1) определяется следующим образом:

$$\delta_j^{[3]}(k) = \frac{\partial E(k)}{\partial u_j^{[3]}(k)} = \frac{\partial E(k)}{\partial e(k)}\frac{\partial e(k)}{\partial u_j^{[3]}(k)} =$$

$$= \psi_j^{[3]\prime}(u_j^{[3]}(k))e(k) = 1{,}5(1 - u_j^{[3]2}(k))e(k),$$

а, следовательно, процедура настройки весов выходного слоя окончательно может быть записана в виде:

$$\Delta w_{ji}^{[3]}(k) = 1{,}5\eta(k)(1 - u_j^{[3]2}(k))e_j(k)o_i^{[2]}(k),$$

где $e_j(k) = (e_1(k), e_2(k), \ldots, e_n(k))^T - (n\times 1)$ — вектор ошибок обучения в момент времени k.

Алгоритм настройки синаптических весов второго скрытого слоя имеет вид:

$$\Delta w_{ji}^{[2]}(k) = \eta(k)\delta_j^{[2]}(k)o_i^{[1]}(k),$$

а его локальная ошибка определяется соотношениями:

$$\delta_j^{[2]}(k) = \frac{\partial E(k)}{\partial u_j^{[2]}(k)} = \psi_j^{[2]\prime}(u_j^{[2]}(k))\frac{\partial E(k)}{\partial o_i^{[2]}(k)} =$$

$$= 1{,}5(1 - u_j^{[2]2}(k))\frac{\partial E(k)}{\partial o_i^{[2]}(k)}, \qquad (2)$$

$$\frac{\partial E(k)}{\partial o_i^{[2]}(k)} = \sum_{j=1}^{n}\frac{\partial E(k)}{\partial u_j^{[3]}(k)}\frac{\partial u_j^{[3]}(k)}{\partial o_i^{[2]}(k)} = \sum_{j=1}^{n}\delta_j^{[3]}(k)w_{ji}^{[3]}(k), \qquad (3)$$

$$\delta_j^{2}(k) = 1{,}5(1 - u_j^{[2]2}(k))\sum_{l=1}^{n}\delta_l^{[3]}(k)w_{li}^{[2]}(k). \qquad (4)$$

Таким образом, алгоритм настройки синаптических весов второго скрытого слоя можно записать в виде:

$$\Delta w_{ji}^{[2]}(k) = 1{,}5\eta(k)(1 - u_j^{[2]2}(k))o_i^{[1]}\sum_{l=1}^{n}\delta_l^{[3]}(k)w_{li}^{[3]}(k).$$

Синаптические веса первого скрытого слоя настраиваются с помощью процедуры:

$$\Delta w_{ji}^{[1]}(k) = \eta(k)\delta_j^{[1]}(k)o_i^{[0]}(k) = \eta(k)\delta_j^{[1]}(k)x(k),$$

а локальная ошибка первого скрытого слоя вычисляется аналогично соотношениям (2–4) и имеет следующий вид:

$$\delta_j^{[1]}(k) = 1{,}5(1 - u_j^{[1]2}(k))\sum_{l=1}^{m}\delta_l^{[2]}(k)w_{li}^{[2]}(k). \qquad (5)$$

Окончательно процедура настройки синаптических весов первого скрытого слоя с учетом (5) описывается выражением:

$$\Delta w_{ji}^{[1]}(k) = 1{,}5\eta(k)(1 - u_j^{[1]2}(k))x_i(k)\sum_{l=1}^{m}\delta_l^{[2]}(k)w_{li}^{[2]}(k).$$

Таким образом, алгоритм обучения данной сети представляет собой процедуру настройки трех матриц синаптических весов, начиная с выходного (третьего скрытого) слоя и имеет вид:

$$\begin{cases} \Delta w_{ji}^{[3]}(k) = 1{,}5\eta(k)(1 - u_j^{[3]2}(k))e(k)o_i^{[2]}(k), \\ \Delta w_{ji}^{[2]}(k) = 1{,}5\eta(k)(1 - u_j^{[2]2}(k))o_i^{[1]}\sum_{l=1}^{n}\delta_l^{[3]}(k)w_{li}^{[3]}(k), \\ \Delta w_{ji}^{[1]}(k) = 1{,}5\eta(k)(1 - u_j^{[1]2}(k))x_i(k)\sum_{l=1}^{m}\delta_l^{[2]}(k)w_{li}^{[2]}(k). \end{cases} \qquad (6)$$

На рис. 3 приведена схема обучения трехслойной автоассоциативной сети для сжатия информации на основе (6).

Численный эксперимент

Рассмотрим задачу компрессии информации на примере данных ирисов Фишера [7] с помощью описанной выше нейронной сети. Рассматривается три вида цветков ириса: Setosa, Versicolor и Virginica. Всего имеется по 50 экземпляров каждого вида, и для каждого из них измерены четыре величины: длина и ширина чашелистика, длина и ширина лепестка. Размерность сжатых данных равна 2. Для данной нейронной сети в качестве активационных используются функции:

$$\psi_j(u_j) = 1{,}5u_j(k) - 0{,}5u_j^3(k), \qquad (7)$$

$$\psi_j(u_j) = \sin\left(\frac{\pi u_j(k)}{2}\right), \qquad (8)$$

$$\psi_j(u_j) = \tanh(u_j(k)). \qquad (9)$$

Алгоритм компрессии информации с активационной функцией (7) позволяет достичь ошибки порядка 4,5 за 8-9 эпох или за среднее время 7с. при использовании комплекса программно-технических средств на базе ПК (Mb: KT266A, CPU: Duron 800Mhz, RAM: 1Gb). Алгоритм компрессии информации с активационной функцией (8) достигает порядка 4,5 за 55 эпох или за среднее время 40с. Алгоритм компрессии информации с активационной функцией (9) при равных начальных условиях достигает ошибки только порядка 13.

На рис. 4 представлены данные ирисов Фишера (3 класса), сжатые алгоритмом с использованием активационной функции (7), среди которых при сжатии линейно отделим класс Setosa.

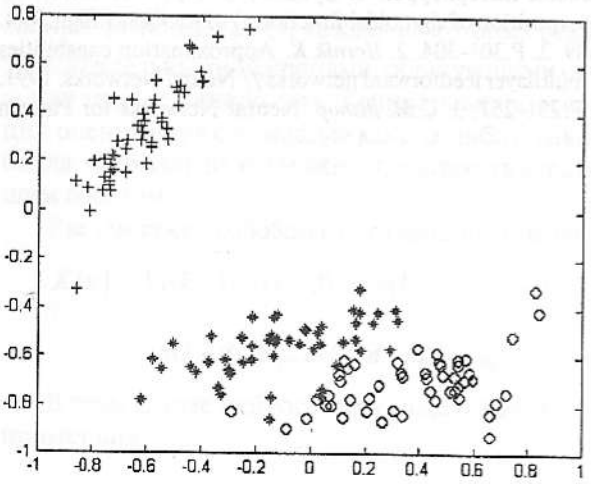


Рис. 4. Графическое изображение сжатых данных ирисов Фишера

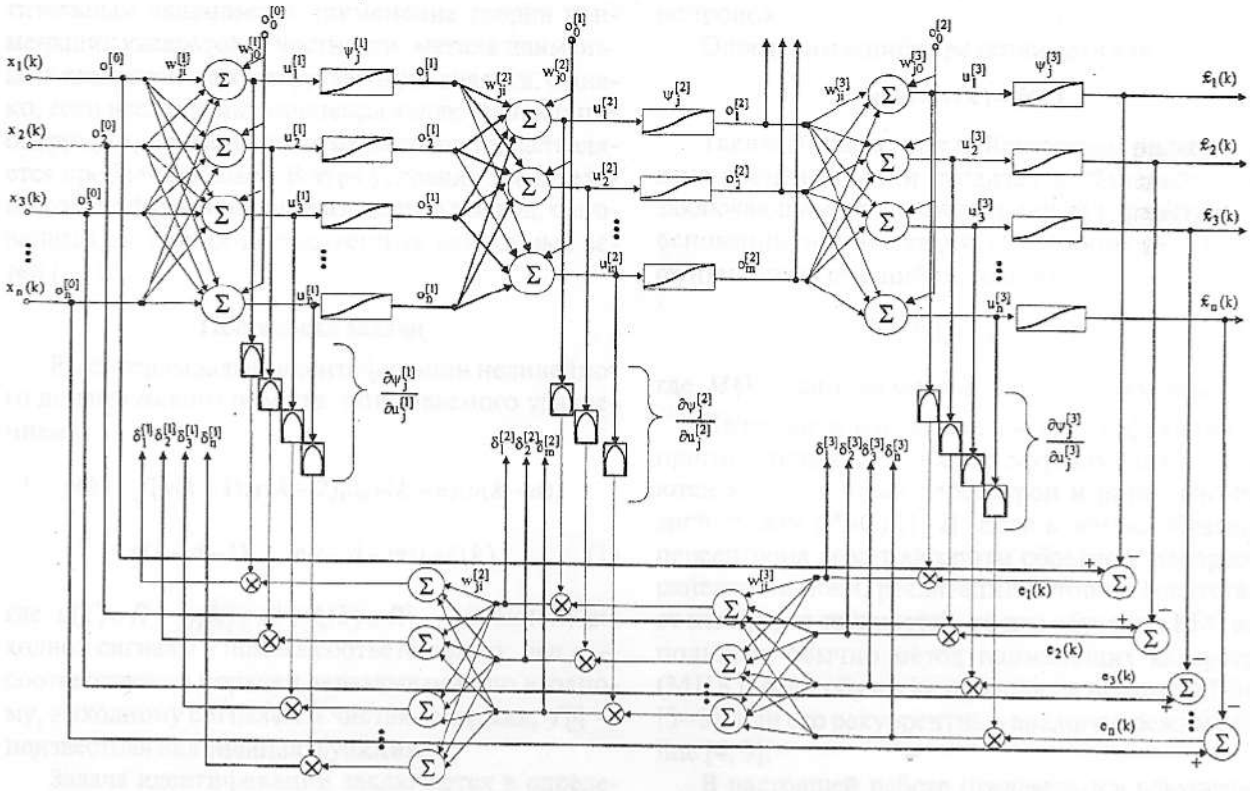


Рис. 3. Схема обучения автоассоциативной искусственной нейронной сети прямого распространения

Заключение

Для задачи сжатия информации предложен алгоритм обучения нейронной сети типа трехслойного персептрона архитектуры bottle-neck, который позволяет упростить численную реализацию процедуры обратного распространения ошибки, что, в свою очередь, позволяет сократить время обучения. Данная нейронная сеть с предложенным алгоритмом обучения может использоваться для сжатия и восстановления данных в реальном времени.

Список литературы: 1. *Cybenko G.* Approximation by superposition of sigmoidal function //Math.Contr.Sign.Syst. 1989. 2. P.303-304. 2. *Hornik K.* Approximation capabilities of multilayer feedforward networks // Neural Networks. 1991. 4. P.251-257. 3. *C.M. Bishop.* Neural Networks for Pattern Recognition. Clarendon Press – Oxford. 1995. 4. *Бодянский Е.В., Полушкина (Тесленко) Н.А.* Алгоритм обучения формального нейрона с полиномиальной активационной функцией// Радиоэлектроника и информатика. 2003. №4. С.45-48. 5. *Полушкина (Тесленко) Н.А., Штанько В.И.* Алгоритм обучения нейронной сети с кубическими функциями активации// Междунар. науч. конф. «Теория и техника передачи, приема и обработки информации»: Тез. докл. Харьков: ХНУРЭ. 2003. С.397-398. 6. *Бодянский Е.В., Слипченко А.Н., Тесленко Н.А.* Прогнозирующая нейронная метасеть с нетрадиционными функциями активации//Автомобильный транспорт: Сб. науч. трудов. Харьков: ХНАДУ. 2003. Вып. 13. С.273-275. 7. *Нейронные сети. STATISTICA Neural Networks*: Пер. с англ. М.: Горячая линия – Телеком. 2001. 182 с.

Поступила в редколлегию 05.11.2004

УДК 519.71

С.А. Ляшенко

ОБ ОДНОМ АЛГОРИТМЕ ОБУЧЕНИЯ НЕЙРОСЕТЕВОЙ МОДЕЛИ НЕЛИНЕЙНОГО ДИНАМИЧЕСКОГО ОБЪЕКТА

Центральными вопросами математического прогнозирования являются выбор и обоснование математической модели прогнозируемого процесса или явления. В свою очередь построение такой модели представляет собой задачу идентификации, успех решения которой зависит от соотношения двух факторов – объёма априорной информации о структуре и параметрах процесса и объёме апостериорной (измеряемой) информации. Априорная информация позволяет определить структуру модели, то есть решить задачу структурной идентификации. На основании же измеряемой или апостериорной информации определяют параметры модели выбранной структуры, то есть решают задачу параметрической идентификации.

Качество решения задач идентификации и прогнозирования во многом определяется характером изучаемого процесса. Так как обычно во всех реальных ситуациях присутствует элемент случайности, для решения данных задач применяют статистические методы. В связи с тем, что априорная информация о свойствах сигналов и помех, как правило, отсутствует или является неполной, предпочтительным оказывается применение теории наименьших квадратов, в частности, метода наименьших квадратов и его рекуррентного аналога. Однако, если исследуемые процессы являются нестационарными, применение этих методов представляется проблематичным. В этих условиях значительно более эффективным оказывается подход, основанный на теории искусственных нейронных сетей [1–4].

Постановка задачи

Рассмотрим задачу идентификации нелинейного динамического объекта, описываемого уравнением

$$y(k)=f[y(k-1),y(k-2),..,y(k-n),u(k-d),$$
$$u(k-d-1),...,u(k-d-m)]+\xi(k), \qquad (1)$$

где $u(k)\in R^1$, $y(k)\in R^1$, $\xi(k)\in R^1$ — входной, выходной сигналы и помеха соответственно; m,u,d — соответственно порядки запаздывания по входному, выходному сигналам и чистая задержка; $f[\cdot]$ — неизвестная нелинейная функция.

Задача идентификации заключается в определении функции $f[\cdot]$ по измеряемым входным и выходным переменным $\{u(k),\ y(k)\}$.

При использовании нейросетевого подхода осуществляется аппроксимация нелинейной функции $f[\cdot]$ некоторой системой заранее выбранных базисных функций $\{\phi_i[\cdot]\}$, каждая из которых реализуется отдельными нейронами, образующими одно – или многослойную сеть. Точность аппроксимации оценивается с помощью какого-либо функционала. Формально это можно представить следующим образом.

Введём вектор обобщённого входного сигнала

$$X(\kappa)=[y(k-1),y(k-2),...,y(k-n),.u(k-d),$$
$$u(k-d-1),...u(k-d-m)]. \qquad (2)$$

В этом случае нейросетевая модель объекта (1) примет вид

$$\hat{y}(k)=\hat{f}[x(k)]=\sum_{i=0}^{p}w_i(k)\Phi_i[x(k)]=\Phi^T[x(k)]w(k), \qquad (3)$$

где $w_i(k)$ — подлежащие определению весовые параметры сети; p — количество используемых в сети нейронов.

Определим ошибку реакции сети как

$$e(k)=y(k)-\hat{y}(k)\ . \qquad (4)$$

Таким образом, при нейросетевом подходе задача идентификации сводится к обучению сети, заключающемуся в определении её параметров на основании минимизации некоторого выпуклого функционала от ошибки, например,

$$J=M\{[y(k)-\hat{y}(k)]^2\}\ , \qquad (5)$$

где $M\{\cdot\}$ — символ математического ожидания.

Применительно к задачам идентификации и прогнозирования наиболее эффективными являются многослойный персептрон и радиально-базисные сети (РБС) [1, 2]. Если в основе обучения персептрона лежит алгоритм обратного распространения ошибки, реализации которого сопутствует целый ряд трудностей, то для обучения РБС используют обычно метод наименьших квадратов (МНК), если обучение проходит в режиме off-line [5–8], или его рекуррентные аналоги в режиме on-line [4, 9].

В настоящей работе предлагаются рекуррентные алгоритмы обучения РБС, представляющие собой модификацию МНК.

Обучение сети

В РБС в качестве базисных функций $\phi_i[x]$ могут быть взяты, например, мультиквадратичная, обратная мультиквадратичная, сплайн-функция и получившая наиболее широкое распространение гауссова функция

$$\Phi_i[x] = \exp\left\{-\frac{\|x-\mu_i\|^2}{\sigma_i^2}\right\}. \quad (6)$$

Здесь μ_i, σ_i^2 — центры и радиусы базисных функций соответственно; $\|\cdot\|$ — евклидова норма.

Иногда вместо функций (6) в нейронной модели (3) применяют нормализированные базисные функции

$$\tilde{\Phi}_i[x(k)] = \frac{\Phi_i[x(k)]}{\sum_{j=1}^{p}\Phi_j[x(k)]}, \quad (7)$$

обладающие свойством $\sum_{i=1}^{p}\tilde{\Phi}_i[x(k)] = 1$, что является весьма существенным для многих приложений.

Если в качестве базисных выбраны функции вида (6) или (7), то в процессе обучения сети должны быть определены все её параметры: w_i, μ_i и σ_i^2. При этом возможны следующие варианты обучения [2]:

1. Задаются центры μ_i и радиусы σ_i^2 и настраиваются веса w_i.

2. Центры μ_i и радиусы σ_i^2 определяются в процессе самообучения, а веса вычисляются путём минимизации функционала (5).

3. Все параметры сети определяются путём минимизации функционала (5).

Первый вариант обучения является наиболее простым, но зачастую наименее эффективным. Он используется в РБС с жестко заданными радиусами σ_i^2. В этом случае оправдывает себя применение нормализованных базисных функций (7), обеспечивающих в отличие от (6) равномерное покрытие всех точек входного пространства и делающих сеть менее чувствительной к неудачному заданию центров μ_i. Однако, как следует из (7), каждая нормализованная базисная функция зависит от всех других базисных функций, и введение в сеть нового нейрона приводит к изменению всех базисных функций. Поэтому при необходимости обучения сети в режиме on-line применение функций вида (7) нецелесообразно. Наиболее эффективным, хотя и наиболее трудоемким, является третий вариант обучения, при котором определяются все компоненты вектора

$$\theta(k) = (w_0(k), w_1(k), \mu_1^T(k), \sigma_1(k), \ldots w_p(k), \mu_p^T(k), \sigma_p(k))^T.$$

Используемый при этом алгоритм обучения рекуррентного МНК имеет вид [1, 9]

$$\theta(k) = \theta(k-1) + \frac{P(k-1)\nabla f(k)}{\lambda + \nabla f^T(k)P(k-1)\nabla f(k)}\cdot e(k), \quad (8)$$

$$P(k) = \frac{1}{\lambda}\cdot\left[P(k-1) - \frac{P(k-1)\nabla f(k)\nabla f^T(k)P(k-1)}{\lambda + \nabla f^T(k)P(k-1)\nabla f(k)}\right], \quad (9)$$

где $\nabla f(k) = \left[1, \Phi_1[x(k)]\right.$,

$$2\Phi_1[x(k)]w_1\sigma_1^{-2}\times(x(k)-\mu_1)^T,$$

$$\left.2\Phi_1[x(k)]w_1\sigma_1^{-3}\times\|x(k)-\mu_1\|^2, \ldots, \Phi_p[x(k)]\right.,$$

$$2\Phi_p[x(k)]w_p\sigma_p^{-2}\times(x(k)-\mu_p)^T,$$

$$\left.2\Phi_p[x(k)]w_p\sigma_p^{-3}\|x(k)-\mu_p\|^2\right]^T;$$

$$e(k) = y(k) - y(\hat{k});\ \lambda \in (0, 1].$$

Как отмечается в [4], с ростом размерности исследуемого объекта и количества обучающих пар $\{u(k),\ y(k)\}$ вычислительные затраты, необходимые для реализации алгоритма (8), (9), существенно возрастают. Поэтому в указанной работе был предложен алгоритм обучения, в котором используется некоторое фиксированное число обучающих пар $S < N$, где N — общее количество настраиваемых параметров. Такой алгоритм обладает лучшими динамическими и худшими по сравнению с (8),(9) фильтрующими свойствами.

Улучшение фильтрующих свойств может быть достигнуто путём увеличения числа обучающих пар, то есть путём выбора $S > N$. Подобный рекуррентный алгоритм может быть построен по аналогии с алгоритмом текущего регрессионного анализа [10]. Использование блочного представления векторов и матриц позволяет получить следующие рекуррентные соотношения, представляющие собой такой алгоритм обучения:

$$\theta_{S+1}(k) = \theta_S(k-1) + \frac{P_S(k-1)\nabla f(k)}{\lambda + \nabla f^T(k)P_S(k-1)\nabla f(k)}\times$$

$$\times(y(k) - \theta_S^T(k-1)\nabla f[x(k)]); \quad (10)$$

$$P_{S+1}(k) = \frac{1}{\lambda}\cdot\left[P_S(k-1) - \frac{P_S(k-1)\nabla f(k)\nabla f^T(k)P_S(k-1)}{\lambda + \nabla f^T(k)P_S(k-1)\nabla f(k)}\right]; \quad (11)$$

$$\theta_S(k) = \theta_{S+1}(k) - \frac{\lambda^S P_{S+1}(k)\nabla f(k-S+1)}{1-\lambda^S + \nabla f^T(k-S+1)P_{S+1}(k)\nabla f(k-S+1)}\times$$

$$\times(y(k-S+1) - \theta_{S+1}^T(k)\nabla f[(k-S+1)]); \quad (12)$$

$$P_S(k) = P_{S+1}(k) + \frac{\lambda^S P_{S+1}(k)\nabla f(k-S+1)\nabla f^T(k-S+1)P_{S+1}(k)}{1-\lambda^S + \nabla f^T(k-S+1)P_{S+1}(k)\nabla f(k-S+1)}, \quad (13)$$

где $\lambda \in (0,1]$.

Соотношения (10), (11) описывают коррекцию вектора оценок θ и информационной матрицы P при поступлении новой информации, а (12), (13) — при удалении устаревшей. Алгоритм (10)–(13) является более динамичным по сравнению с (8), (9), приближаясь ростом S к нему по фильтрующим свойствам.

Если же при обучении сети сначала происходит удаление устаревшей информации, а потом учитывается вновь поступившая, то алгоритм примет следующий вид:

$$\theta_{S-1}(k) = \theta_S(k-1) - \frac{\lambda^{S-1}P_S(k-1)\nabla f(k-S+1)}{1-\lambda^{S-1} + \nabla f^T(k-S+1)P_S(k-1)\nabla f(k-S+1)} \times$$

$$\times (y(k-S+1) - \theta_S^T(k-1)\nabla f(k-S+1)); \quad (14)$$

$$P_{S-1}(k) = P_S(k-1) + \frac{\lambda^{S-1}P_S(k-1)\nabla f(k-S+1)\nabla f^T(k-S+1)P_S(k-1)}{1-\lambda^{S-1} + \nabla f^T(k-S+1)P_S(k)\nabla f(k-S+1)}; \quad (15)$$

$$\theta_S(k) = \theta_{S-1}(k) + \frac{P_{S-1}(k)\nabla f(k)}{\lambda + \nabla f^T(k)P_{S-1}(k)\nabla f(k)} \times$$

$$\times (y(k) - \theta_{S-1}^T(k)\nabla f(k)); \quad (16)$$

$$P_S(k) = \frac{1}{\lambda}\cdot\left[P_{S-1}(k-1) - \frac{P_{S-1}(k-1)\nabla f(k)\nabla f^T(k)P_{S-1}(k-1)}{\lambda + \nabla f^T(k)P_{S-1}(k-1)\nabla f(k)}\right]. \quad (17)$$

В принципе оценки, получаемые с помощью алгоритмов (10)–(13) и (14)–(17), совпадают. Однако применение алгоритма (14)–(17) оказывается более предпочтительным с точки зрения удобства вычислений (например удаление устаревшей информации в (16) приводит к меньшему количеству арифметических операций при вычислении (17)).

Моделирование

Изучалась эффективность работы алгоритма обучения(10)–(13) при построении нейронных моделей различных нелинейных динамических объектов. В частности, на рис. 1–3 представлены результаты моделирования объекта, описываемого уравнением [4, 11]

$$y(k) = 0{,}725\beta \cdot \left(\frac{16u(k-1) + 8y(k-1)}{\beta(3+4u(k-1)) + 4y^2(k-1)}\right) + 0{,}2u(k-1) + 0{,}2y(k-1), \quad (18)$$

при отсутствии и наличии помехи измерений $\xi(k)$ (1).

Условия эксперимента совпадали с теми, которые описаны в [4], то есть входной сигнал $u(k)$ и помеха $\xi(k)$, получаемые с помощью датчика случайных чисел, представляли собой стационарные p-равномерно распределенные в интервале [–1; 1] и [–0,25; 0,25] соответственно случайные последовательности. Построение модели осуществлялось на основании 5000 обучаемых пар. Все кривые отражают зависимость $y(k)$ от $u(k-1)$ при $y(k-1) = 0$.

На рис. 1, 2 сплошные линии соответствуют выходному сигналу объекта $y(k)$, линии с крестиками и кружками — выходному синтезу модели, обучаемой по алгоритму (10)–(13) при выборе $\lambda = 1$ и $\lambda = 0{,}995$ соответственно.

На рис. 1 приведены результаты идентификации объекта (18) при различных значениях параметров β и отсутствии помех измерений.

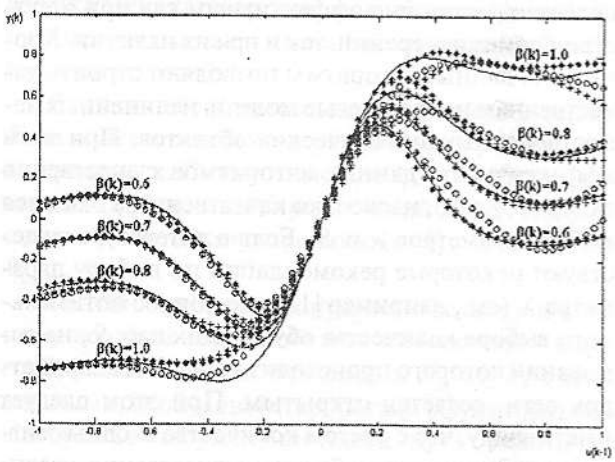


Рис. 1

На рис. 2, 3 показаны результаты идентификации при наличии помех измерения $\xi(k)$. Рис. 2 отражает работу алгоритма обучения (10)–(13), а рис. 3 — алгоритма (14)–(17).

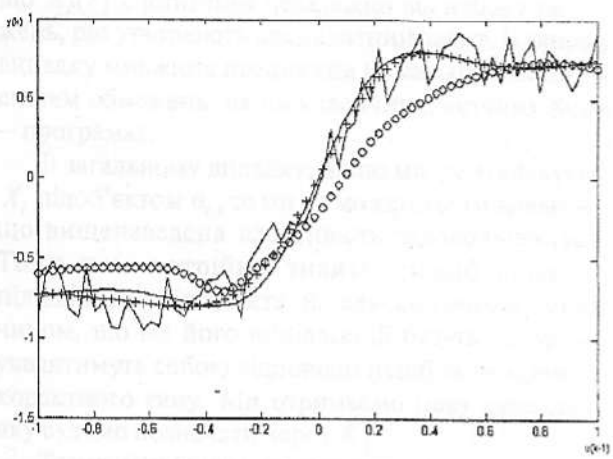


Рис. 2

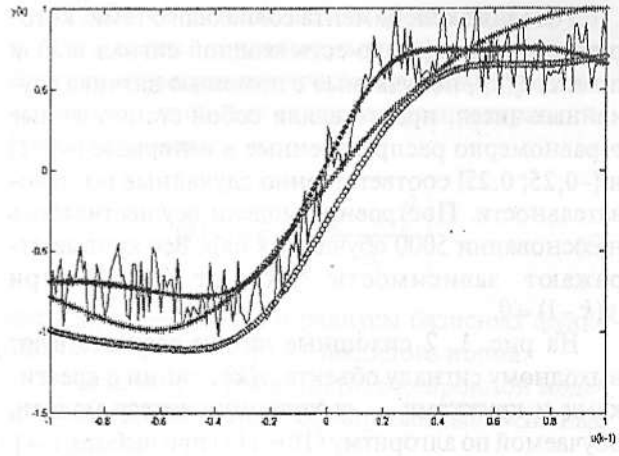


Рис. 3

Выводы

Как следует из приведенных результатов, использование алгоритмов обучения (10)–(13) и (14)–(17) является достаточно эффективным как при отсутствии помех измерений, так и при их наличии. Кроме того, данные алгоритмы позволяют строить качественные нейросетевые модели нелинейных нестационарных динамических объектов. При этом эффективность данных алгоритмов существенно зависит от того, насколько качественным оказался выбор параметров λ и S. Если в литературе существуют некоторые рекомендации по выбору параметра λ (см., например [12]), то вопрос оптимального выбора количества обучающих пар S, на основании которого происходит коррекция параметров сети, остаётся открытым. При этом следует иметь ввиду, что с ростом количества входных сигналов исследуемого объекта, существенно увеличивается число нейронов сети, приводящее к резкому возрастанию вычислительной сложности реализации нейросетевой модели. Поэтому алгоритмы типа (10)–(13) и (14)–(17) являются наиболее эффективными при построении моделей нестационарных нелинейных объектов с небольшим количеством входных сигналов.

Список литературы: 1. *Nelles O.* Nonlinear system identification. Berlin: Springer, 2001. 785p. 2. *Руденко О.Г., Бодянский Е.В.* Основы теории искусственных нейронных сетей. Харьков: Телетех, 2002. 317с. 3. *Sohilling R.J., Carroll J.J., Al-Ajlouni A.F.* Approximation of nonlinear systems with radial basis function neural networks //IEEE Trans. on Neural Networks. 2001. V.12 - №1. P. 1-15. 4. *Руденко О.Г., Бессонов А.А.* Идентификация нелинейных нестационарных объектов в реальном времени с помощью радиально-базисных сетей //Кибернетика и системный анализ. 2003. №6. С.177-185. 5. *Chen S., Billings S.A.* Neural networks for nonlinear dynamic system modeling and identification // Int. J. Control. 1992. V.56. №2. P. 319-346. 6. *Zhu S., Billings S.A.* Fast orthogonal identification of nonlinear stochastic models and radial basis function neural networks // Int. J. Control. 1996. V.64. №5. P. 871-886. 7. *Chen S., Chng E.S., Alkadhimi K.A.* Regularized orthogonal least squares algorithm for constructing radial basis function networks // Int. J. Control. 1996. V.64. №5. P. 829-837. 8. *Chng E.S., Chen S., Mulgrew B.* Gradient radial basis function networks for nonlinear and nonstationary time series prediction // IEEE Trans. on Neural Networks. 1996. V.7 №1. P. 190-194. 9. *Fung Ch.F., Billings S.A., Luo W.* On-line supervised adaptiv training using radial basis function networks // Neural Networks. 1996. V.9 №3. P. 1597-1617. 10. *Руденко О.Г., Теренковский И.Д., Штефан А., Ода Г.А.* Модернизированный алгоритм текущего регрессивного анализа в задачах идентификации и прогнозирования // Радиоэлектроника и информатика. 1998. №4. С. 58-61. 11. *Nerendra K.S., Parthasarathy K.* Identification and control of dynamical system using neural networks // IEEE Trans. on Neural Networks. 1990. V.1 №1. P. 4-26. 12. *Куник Е.Г., Коваленко А.Н., Ляшенко С.А.,* Аль Сади Ф.М. О выборе параметра взвешивания информации при построении нестационарных регрессионных моделей // АСУ и приборы автоматики. 2002. Вып.116. С. 109-114.

Поступила в редколлегию 19.11.2004

УДК 681. 3:622. 276

В.І. Шекета

ПОБУДОВА МОДЕЛІ МОДИФІКАЦІЙНИХ ПРЕДИКАТНИХ ЗАПИТІВ НА ОСНОВІ СИНТАКСИЧНОЇ КАТЕГОРІЙНОЇ СТРАТЕГІЇ

Категорійні підходи до логічного програмування з'явилися разом із категорійним підходом до процедури уніфікації [1...9]. Основним результатом стало введення категоріальної формалізації для синтаксису логіки тверджень Хорна, і її розширення на основі семантики теоретичних топосів. В [10] розвиваючи деякі базові ідеї, сформульовані в [11], виконано категоріальний аналіз логічних програм і побудову відповідних моделей на основі використання індексованих моноїдних категорій.

Всі ці підходи зосереджені на побудові суто теоретико – операційних моделей. В той же час мало уваги приділяється застосуванню денотаційних семантик до побудови операторів на зразок оператора безпосереднього слідування, який є суто важливим із точки зору побудови логічних програм і дослідження їх семантик [12]. Більшість досліджень семантик логічних програм зосереджено на побудові формальних конструкцій на основі теорії фіксованих значень. Тому саме з цих причин, доцільним є подальше дослідження застосувань категорійного апарату, який включає в себе семантики на основі фіксованих значень. Першою роботою даного напрямку була робота [13] в якій було введено поняття категоріального синтаксису над множиною скінченних категорій. Це послужило вихідним пунктом для введення як поняття категорійної дедукції, так і денотаційних семантик, що є відповідниками семантик коректних рішень для логічних Хорн – програм. Такі семантики можуть бути обчислені на основі конструкцій для фіксованих значень, що не виходять за рамки категоріальної дедукції. Однією із переваг такого підходу є те, що категорія термів не обов'язково повинна співпадати з відповідною алгебраїчною категорією для заданої множини функціональних символів.

Всі рішення в нафтогазовій предметній області приймаються на основі аналізу висновків експертів, спеціалістів з великим досвідом роботи. В роботі [14] база знань інформаційної системи розглядається як набір інформаційних сутностей атомарних предикатів з деякого скінченного інформаційного простору B. Всі зміни, що відбуваються в базі знань, розглядаються , як наслідок модифікаційних предикатних запитів Q_m. Основою самих запитів є набір модифікаційних предикатних правил:

$$Q_M \longleftrightarrow (K_B)^{<<} \left\| \begin{matrix} K_{B-}(o) \\ K_{B+}(o) \end{matrix} \right. <<,$$

де $o, o_n, p_n \in \Re$. $K_{B+}(o)$ означає, що атомарний предикат o має бути включений у базу знань K_B, K_{B-} означає, що o має бути виключений з бази знань; $(K_B)^{<<}$ – означає модифікацію бази знань на рівні логічної зв'язаності предикатних правил, як наслідок виконання операцій додавання і вилучення правил; << – дескриптор модифікації, який розглядається, як категорійна стрілка. **Недослідженим** залишається питання категорійної інтерпретації самих модифікаційних предикатних запитів.

Таким чином, метою даної статті є введення і дослідження категорійної моделі модифікаційних предикатних запитів на основі синтаксичної категорійної стратегії.

Для заданої скінченної категорії добутків термів K ми знаємо, що монострілки можна розглядати, як предикати. Припустимо, що ми хочемо побудувати модифікаційний предикатний запит, використовуючи множину предикатів X_1, ... X_k типів θ_1, ... θ_k.

Основна ідея, яку ми прагнемо досягнути в даному дослідженні, є побудова синтаксичної категорії, в якій значення предикатів X_1, ... X_k є явно заданими, і побудова інтерпретації (функтора), що відображає одержану синтаксичну і семантичну категорію предикатних запитів способом, який є сумісним із твердженнями, що утворюють сам запит. Коли ми говоримо про явний спосіб задання значень предикатів X_i в синтаксичній категорії, то ми маємо на увазі, що для кожного терма $\ell : r >> \theta_i$ зворотний зсув X_i вздовж ℓ є відповідним підоб'єктом для r. Іншими словами, це буде означати, що $X_i(\ell)$ є істинним незалежно від набору тверджень, що утворюють предикатний запит. В даному випадку множина предикатів не вимагає введення систем обмежень, як і в класичних логічних Хорн – програмах.

В загальному випадку, якщо ми ідентифікуємо X_i підоб'єктом θ_i, то ми не можемо бути впевнені, що вищенаведена властивість задовольняється. Тому нам потрібно знайти спосіб вільного під'єднання підоб'єкта θ_i для кожного X_i таким чином, що всі його ініціалізації будуть існувати і уявлятимуть собою відповідні підоб'єкти заданого коректного типу. Ми отримаємо нову категорію, яку будемо позначати через $K[X_1, ..., X_k]$.

Тепер можна ввести поняття категорійної дедукції. Нехай ціль задано у вигляді послідовності

атомарних цілей і твердження є парою, утвореною із цілі Ch і атомарної цілі (заголовка) $X_i(\ell)$. Використовуватимемо запис $X_i(\ell) >> Ch$. В першому наближенні будемо розглядати категорійну дедукцію як послідовність кроків в транзитивній системі із мітками. Позначатимемо даний факт через ~>. Таким чином, якщо Z_1 і Z_2 є цілями, φ — підстановкою і t — тверженням виду $X_\ell(\iota_1) >> Ch$, тоді

$$Z_1 \overset{\varphi, t}{\sim>} Z_2,$$

де $Z_1 = X_{J_1}(\varsigma_1), \ldots, X_{J_k}(\varsigma_k), \ldots, X_{J_l}(\varsigma_l)$ для деякої стрілки

$$\varsigma : Y_1 >> \theta_k;$$

$$Z_2 = \sigma\tilde{}\ X_{J_1}(\varsigma_1), \ldots, \varphi\tilde{}\ Ch, \ldots, \sigma\tilde{}\ X_{J_1}(\varsigma_\ell).$$

В даному випадку пара σ, φ є уніфікатором для ς і ς_1, і маємо пару стрілок замість одної, оскільки ми виконуємо уніфікацію стрілок із різних джерел (що відповідає операції попереднього переіменування термів).

Таким чином, категорійну дедукцію будемо розглядати як дедукцію в транзитивній системі ~>. Категорійним спрощенням будемо вважати категорійну дедукцію, що закінчується порожньою ціллю. Для заданої дедукції

$$kd = Z_1 \overset{\sigma_1, t_1}{—\sim>} \ldots \overset{\sigma_k, t_k}{—\sim>} Z_k,$$

обчислюваний розв'язок для kd означимо через композицію $\sigma_k; \ldots; \sigma_1$.

Інтерпретацією в даному випадку буде функтор, що зберігає скінченні добутки

$$[F] : K[X_1, \ldots, X_k] >> Set^{K^{\varnothing}}$$

та розширює Y — вбудування (тобто таке, що $[\theta] = H(F, \theta)$ для кожного $\theta \in O_k$) і виконує прив'язку підоб'єкта $H(F, \theta_i)$ до X_i. Можна показати, що для заданої прив'язки підоб'єктів для X_l — їх існує тільки одна інтерпретація, що розширює дану прив'язку. Більше того, множина таких інтерпретацій утворює повну структуру.

Тепер введемо оператор на множині інтерпретації R_Q, параметризація якого задана по відношенню до запиту Q

$$R_Q([F])(X_i) = \bigcup_{X_i(\iota) >> Ch \in Q} \mathrm{Im}_{[\iota]}([Ch]),$$

де $\mathrm{Im}_f(X)$ є образом монострілки X вздовж стрілки f.

Тому замість розгляду цілей як монострілок в категорії K, використаємо індексовану категорію над K. Об'єкт на шарі $\theta \in O_K$ буде категорійним відповідником цілі типу θ. Тобто, ми не виходитимемо за рамки стандартної категорійної інтерпретації логіки першого порядку.

$\wedge\Pi$ — категорійною стратегією будемо вважати індексовану категорію Ξ_1 над базовою категорією K. Для кожного $\theta \in O_k$ об'єкти і стрілки в $\Xi_1\theta$ будемо називати формулами і доведеннями (абстрактного типу θ) відповідно. Будемо використовувати термін ціль як синонім до формула. Для заданої цілі Z абстрактного типу θ і $f : r >> \theta$ в K $f : Z = Z(f)$ є ініціалізацією для Z.

Будемо записувати $Z : \theta$ і $f : \theta$ як скорочені позначення для $Z \in O_{Q_\theta}$ і $f \in M_{r_{Q_\theta}}$. Для заданої $\wedge\Pi$ категорійної стратегії твердженням (абстрактного типу θ) є об'єкт tr із відповідною парою (Ch, Zh) цілей абстрактного типу θ. Позначимо даний факт як $Zh \overset{tr}{>>} Ch$.

Модифікаційним предикатним запитом будемо вважати пару (Q, Ξ_1), де Ξ_1 є $\wedge\Pi$-категорійною стратегією, і Q є множиною тверджень. Будемо говорити, що Q є запитом над Ξ_1.

Модифікаційний запит можна розглядати також як індексовану категорію Q над O_k, таку, що $Q(\theta)$ є категорією об'єктів абстрактного типу θ стрілок $tr : Zh >> Ch$ тверджень типу θ.

Основна ідея, що лежить в основі пропонованого підходу полягає в тому, що базова категорія представляє універсум всіх можливих станів, до яких може привести виконання модифікаційного запиту. Для кожного стану відповідний шар представляє множину дедукцій, які можуть бути виконані. Твердження модифікаційного запиту в даному випадку є новими дедукціями, які ми можемо розглядати в доповнення до доведень, що містяться в шарах індексованої категорії.

Нехай задано сигнатуру першого порядку M_{F_1}, утворену із множини F функціональних символів і множин Π-предикатних символів відповідної розмірності. Побудуємо категорію $T_{M_{F_1}}$, як алгебраїчну категорію на основі F. Об'єктами $T_{M_{F_1}}$ є натуральні числа, стрілками із k до $l \in l$ — кортежі із термів, побудовані на основі множини змінних $\{w_1, \ldots, w_k\}$.

$$O_{T_{M_{F_1}}} = N,\ T_{M_{F_1}}(k, l) = S_{M_{F_1}}(\{w_1, \ldots, w_k\})^l.$$

Тепер виконаємо побудову синтаксичної категорії $\Xi^1_{M_{F_1}}$ (синтаксичної категорійної стратегії).

1. Для кожного $k \in N$ $\Xi^1_{M_{F_1}}(k)$ є дискретною категорією атомарних цілей, утвореною із змінних $w_1, \ldots, w_k$.

2. Для $\iota = <\iota_1, \ldots, \iota_l> | k >> l$ $\Xi^1_{M_{F_1}}(\iota)$ є функтором, що задає відображення атомарної цілі Z в $Z[w_1/\iota_1, \ldots, w_l/\iota_l]$.

Нехай Π — сигнатура предикатів над K, тобто фактично множина предикатних символів відповідного типу в O_K. Будемо записувати $\pi : \theta$, якщо π є предикатним символом типу θ. Тоді ми можемо оголосити індексовану категорію Ξ^1_Π над K таку, що:

1. Ξ^1_Π (θ) дискретна категорія, об'єктами якої є пари $<\pi, f>$ такі, що $\pi: r \in \prod$ і $f: \theta >> r$ є стрілкою в K. Будемо записувати $\pi(f)$ замість $<\pi, f>$.

2. Ξ^2_Π (θ), де $F: r >> \theta$ є функтором, що задає відображення $\pi(\iota) \in O_{\Xi^1_\Pi(\theta)}$ в $O(f, \iota)$.

Для заданого Q над категорійною стратегією Ξ_1 *моделлю модифікаційного предикатного запиту* Q буде пара $([F], \nu)$, де $[F]: \Xi_1 >> \Xi_2$ є інтерпретацією, і ν є функцією, що виконує відображення твердження $Zh \overset{tr}{<<} Ch \in Q$ в стрілку $[Zh] \overset{\nu(tr)}{<<} [Ch]$.

Якщо ми розглянемо запит як індексовану категорію, то тоді ν є функтором із Q в $T(\Xi_2)$, де $T: IKt >> IKt$ є функтором, який задає відображення індексованої категорії над K в індексовану категорію над O_K, опускаючи всі стрілки в базовій категорії. Формально кажучи, якщо $\Xi_2 : K >> Kt$, то ми маємо, що $T(\Xi_2): O_K >> Kt$ таке, що $T(\Xi_2)(\theta) = \Xi_2(\theta)$.

Нехай ми маємо модифікаційний предикатний запит (Q, Ξ^1). Вище ми ввели поняття категорійної дедукції. Починаючи із інтерпретації $[F]: \Xi^1 \to \Xi^2$, нові версії оператора R_Q даватимуть в якості результату нову інтерпретацію $[F]': \Xi^1 \to \Xi^2$, яка може бути розширена до рівня моделі Q і є більш ефективною за модель $([F], \nu)$.

Модифікаційний предикатний запит (Ξ^1, Q) будемо вважати *типізованим*, якщо існує множина $\{X_1 : \theta_1, \ldots, X_k : \theta_k\}$ абстрактних цілей із наступними властивостями:

1) Ξ^1 одержується із категорійної стратегії $\overline{\Xi^1}$ шляхом приєднання абстрактних цілей до відповідних шарів $\overline{\Xi^1}$;

2) немає тверджень, направлених на задоволення цілей в $\overline{\Xi^1}$.

Ініціалізацію абстрактної цілі будемо називати динамічною ціллю. Так, якщо $\iota: r >> \theta_i$, то будемо використовувати запис для $\iota \tilde{} X_i$ також і в формі $X_i(\iota)$. Метою введення динамічних цілей є виконання покрочної модифікації конструкцій для фіксованих значень, в той час, коли всі цілі в $\overline{\Xi^1}$ мають фіксоване значення. Для заданого $[F]: \Xi^1 \to \Xi^2$ інтерпретація усіх динамічних цілей залежить тільки від інтерпретації абстрактних цілей.

Нехай Q є модифікаційним предикатним запитом над синтаксичною категорійною стратегією Ξ^1_Π. Ми можемо позначити $\overline{\Xi^1}: K^\varnothing \to Kt$ таким чином, що:

1) для кожного $\theta \in O_\kappa$, $\overline{\Xi^1}(\theta) = \varnothing$;

2) для кожного $\iota \in M_{r_K}$, $\overline{\Xi^1}(\iota) = id_\varnothing$.

Тобто Ξ^1_Π може бути одержана через приєднання до $\overline{Q}$ цілі $\pi(id_\theta)$ для кожного $\pi : \theta \in \Pi$. Те ж саме матиме місце і для $\Xi^1_{M_F}$.

Для того, щоб означити оператор визначення фіксованого значення із ефективними властивостями, нам потрібна більш комплексна категорійна структура в стратегії Ξ^2, ніж в Ξ^1. Для цього введемо поняття семантичної стратегії.

Семантичною категорійною стратегією Ξ^2 будемо вважати категорійну стратегію, де

1) шари містять ко-добутки і канонічні ко-границі для υ-ланцюгів;

2) кожний переіндексований функтор $\Xi^2\iota$ має лівий приєднаний елемент $\Delta_\iota^{\Xi^2}$ і зберігає канонічні ко-границі для υ-ланцюгів.

Домовимось опускати верхній індекс Ξ^2 із $\Delta_\iota^{\Xi^2}$, коли це є зрозумілим із контексту. Якщо ми працюємо тільки із скінченними запитами, то буде достатнім, щоб шари містили тільки скінченні ко-добутки.

Для заданої категорії скінченних добутків K розглянемо індексовану категорію Ξ^2. В даному випадку є можливим виконання трансформації Ξ^2 в семантичну категорійну стратегію, оскільки кожний шар є новою структурою, і тому він містить ко-добутки, задані через операцію перетину, і канонічні ко-добутки для υ-ланцюгів, задані через операцію об'єднання. Далі ми можемо оголосити $\Delta_f^{\Xi^2}$, де $f : r >> \theta$, як функцію, що задає відображення $X \subseteq H_K(1, r)$ в $\{\iota \circ f \mid \iota \in X\}$, яка є підмножиною для $H_K(1, \theta)$. Дана функція є монотонною і тому може бути перетворена в функтор тривіальним способом :

– $\Delta_\iota^{\Xi^2}$ є адитивним, і зберігає ко-границі;

– $\Delta_\iota^{\Xi^2}$ є лівим елементом приєднання для $\Xi^2(\iota)$, оскільки

$$\left(\Xi^2(\iota) \circ \Delta_\iota^{\Xi^2}\right)(X) = \left\{ f \in X \,\middle|\, \begin{matrix} f \text{ задає} \\ \text{факторизацію для } \iota \end{matrix} \right\} \subseteq X.$$

$$\left(\Delta_\iota^{\Xi^2} \circ \Xi^2(\iota)\right)(X) = \left\{\Delta_{f2} \in X, f \circ \iota \circ f_2 \circ \iota\right\} \subseteq X.$$

В наступних дослідженнях ми будемо використовувати введену семантичну категорійну стратегію для побудови семантик фіксованих значень для модифікаційних предикатних запитів в Ξ^1_Π.

Нехай тепер маємо інтерпретацію $[F] = (F, \iota)$ із Ξ^1 в Ξ^2, де Ξ^2 є семантичною категорійною стратегією. Ми хочемо побудувати крок за кроком модифіковану інтерпретацію, яка також буде моделлю для Q. На першому кроці ми перейдемо від $[F]$ до $R_Q([F]) = (F, \iota')$, де

$$\ell'_{\theta_i}(X_i) = [X_i] \vee \bigcup_{X_i(\iota) << Ch \in Q} \Delta_{F\iota}[Ch]$$

$$\ell_\theta\left(X_i(\iota)\right) = \iota^{-1}\left(\ell'_{\theta_i}(X_i)\right),$$

при умові, що $\ell' = \ell$, що обмежує тільки $\overline{\Xi^1}$. В загальному випадку потрібно було би оголосити ℓ'

на множині стрілок, але оскільки між динамічними цілями ми розглядаємо тільки відношення ідентичності, то результат є суть очевидним.

Згідно з означенням $(F,\ \ell')$ є індексованим функтором. В той же час, якщо ми хочемо, щоб R_Q був функтором на множині інтерпретацій, то ми повинні спершу означити його поведінку на множині індексованих природніх перетворень. Якщо θ_1 є стрілкою між інтерпретаціями, то ми матимемо, що $R_Q(\theta_1) = \theta_1'$, де

$$(\theta_1)'_{\theta_i X_i} = (\theta_1)_{\theta_i,\ X_i} \vee \bigcup_{X_i(\iota:\ r >> \theta_i) << Ch \in Q} \Delta_{F_\iota}(\theta_1)_{r,\ Ch}$$

$$(\theta_1)'_{\theta_i X_i(\iota)} = \iota^{-1}\left((\theta_1)'_{\theta_i,\ X_i}\right), (\theta_1)_{\theta_i Z}\ (\theta_1)_{\theta_i Z},$$

якщо $Z \in O_{\overline{\Xi^1}}$. Оскільки $\overline{\Xi^1}$ містить тільки нетривіальні стрілки і $(\theta_1)_\theta$ є природнім перетворенням для кожного θ, то те саме має місце і для $(\theta_1)'_\theta$.

Твердження 1. Інтерпретація $[F]$ для модифікаційного предикатного запиту Q може бути розширена до рівня моделі Q тоді і тільки тоді, коли вона може бути розширена до алгебри для R_Q.

Доведення. Припустимо, що $[F]$ може бути розширена до алгебри для R_Q, тобто існує стрілка $\tau : R_Q([F]) \to [F]$. Для заданого твердження $X_i(\iota) << Ch$ існує стрілка

$$[Ch] \xrightarrow{L} \iota^{-1}\ \Delta_{F_\iota}\ [\ Ch] \xrightarrow{inj} \iota^{-1}$$

$$\left([X_i] \vee \bigcup_{X_i(S) << Z \in Q} \Delta_{F_\iota}[Z]\right) =$$

$$= \iota^{-1}\ R_Q\ ([F])(X_i) = R_Q([F])\ldots(X_i(\iota)) \xrightarrow{\tau_{X_i(\iota)}} [X_i(\iota)]$$

І навпаки, припустимо, що $([F],\ \nu)$ є моделлю для Q. Виконавши фіксацію абстрактної цілі X_i для кожного твердження $tr = X_i(\iota) << Ch$, ми матимемо стрілку

$$St_{tr} = \Delta_{F_\iota}\ [\ Ch] \xrightarrow{\Delta_{F_\iota}\nu(tr)} \Delta_{F_\iota}\ [\ X_i(\iota)] \xrightarrow{L_\iota} [\ X_i]$$

Використовуючи властивості ко-добутків, ми можемо означити

$$\tau_{\theta_i,\ X_i} : R_Q([F])\ (X_i) \xrightarrow{[id, \{St_{tr}\}_{tr}]} [\ X_i],$$

$$\tau_{\theta_i,\ X_i(\iota)} : \iota^{-1}\ \tau_{\theta_i,\ X_i}\ \text{і}\ \tau_{\theta_i,\ Z} = id_Z,$$

якщо $Z \in O_{\overline{\Xi^1}}$, а це власне і є індексоване природнє перетворення $\tau : R_Q([F]) \to [F]$. Звідки маємо, що $([F],\ \tau)$ є алгеброю для R_Q.

Крім того, слід відзначити також, що в даному випадку існує канонічне перетворення η між $[F]$ і $R_Q([F])$, задане наступним чином:

$$\eta_{\theta_i,\ X_i} = [X_i]_{\theta_i} \xrightarrow{inj} R([F])_{\theta_i}(X_i),$$

$$\eta_{\theta_i,\ X_i(\iota)} = \iota^{-1}(\nu_{\eta_i,\ X_i}),\ \eta_{\theta_i,\ Z} = id_Z,\ \text{якщо}\ Z \in O_{\overline{\Xi^1}}$$

де *inj* – є ін'єктивним відображенням. Це в свою чергу означає, що $([F],\ \eta)$ є ко-алгеброю для R_Q.

Твердження 2. Оператор безпосереднього слідування R_Q зберігає ко-границі для υ-ланцюгів в $IF_{n_F}(\Xi^1,\ \Xi^2)$.

Доведення. Нехай D – діаграма в підкатегорії $IF_{n_F}(\Xi^1,\ \Xi^2)$ індексованих природніх перетворень форми L. Припустимо, що $([F],\ \{\tau_i\}_{i \in O_L})$ є канонічною ко-границею для D. Ми хочемо довести, що $\left(R_Q([F],\ \{R_Q(\tau_i)\}_{i \in O_L})\right)$ є ко-границею для $D \circ R_Q$.

Для цього нам достатньо довести, що для кожного $\theta \in O_\kappa$

$$Z \in O_{\Xi^1\theta}\left(R_Q([F])_\theta(Z),\ \left\{\left(R_Q(\tau_i)_{\theta_i Z}\right)\right\}_{i \in O_L}\right) =$$

$$= colim\ (D \circ R_Q \circ Fb_\theta \circ OF_Z),$$

де OF – множина формул(тверджень) першого порядку; Fb – множина вільних змінних.

Крім того, ми знаємо, що

$$\left([Z]_\theta,\ \left\{(\tau_i)_{\theta,\ Z}\right\}_{i \in O_L}\right) = colim\ (D \circ Fb_\theta \circ OF_Z).$$

Якщо Z є абстрактною ціллю X_n, і оскільки Δ_{F_ι} є лівим елементом приєднання, що зберігає ко-границі, то матимемо

$$\left(\Delta_{F_\iota}[Ch]_\theta, \left\{\Delta_{F_\iota}(\tau_i)_{\theta,Ch}\right\}_{i \in O_L}\right) = colim(D \circ Fb_\theta \circ OF_{Ch} \circ \Delta_{F_\iota})$$

для кожного твердження $X_n(\iota) << Ch$, де $\iota : \theta >> \theta_n$. Відповідно до твердження 1, в кінцевому підсумку одержимо, що:

$$\left(R_Q([F])_{\theta_n, X_n}, \left\{\left(R_Q(\tau_i)_{\theta_n, X_n}\right)\right\}_{i \in O_L}\right) =$$

$$= \left([X_n] \vee \bigcup_{X_n(\iota) << Ch} \left\{\Delta_{F_\iota}(\tau_i)_{\theta,Ch}\right\}_{i \in O_L}\right) =$$

$$= colim \left(\begin{matrix} D \circ Fb_{\theta_n} \circ OF_{X_n} \vee \\ \bigcup_{X_n(\iota) << Ch} (D \circ Fb_\theta \circ OF_{Ch} \circ \Delta_{F_\iota}) \end{matrix}\right).$$

Оскільки остання діаграма є рівною до $\left(D \circ R_Q \circ Fb_\theta \circ OF_{X_n}\right)$, то теорема має місце для абстрактних цілей. У випадку динамічних цілей виду $X_n(\iota)$ доведення безпосередньо слідує із того фак-

ту, що переіндексовані функтори в Ξ^2 зберігають ко-границі для υ – ланцюгів.

Тепер ми можемо отримати фіксоване значення $\left(R_Q^{\upsilon}([F]),\ \theta_1\right)$, а значить і модель для Q, яка є найменшою моделлю, що розширює $([F],\ \nu)$.

Таким чином, для заданого модифікаційного предикатного запиту $(Q,\ \Xi^2)$ R_Q має найменше фіксоване значення, яке є моделлю для Q в Ξ^2.

Для спрощення запису будемо використовувати позначення $[F]^k$ для $R_Q^k([F])$, що матиме місце і для випадку $k = \upsilon$.

Якщо ми тепер запишемо означення для R_Q із дотриманням вимог введеної синтаксичної категорійної стратегії, то отримаємо, що

$$R_Q([F])_{\theta_i}(X_i) = [X_i] \vee \bigcup_{X_i(\iota) << X_j(S)} \left\{ f \circ \iota \middle| f \circ s \in [X_j] \right\}$$

$$R_Q([F])_{\theta_i}(X_i(\iota)) = \left\{ f \middle| f \circ \iota \in [Ch] \right\}.$$

Якщо ми працюємо із K, означеним на основі алгебраїчної категорії для сігнатури M_{F_1}, то тоді $R_Q([F])$ стає еквівалентним до стандартної R_Q – семантики модифікаційних предикатних запитів[15,16].

Висновки та перспективи подальших досліджень

В даній статті запропоновано категоріальну модель модифікаційних предикатних запитів на основі початкової інтерпретації для синтаксичної категорійної стратегії в рамках денотаційної семантики шляхом введення динамічних цілей і виконання покрочної модифікації конструкцій для фіксованих значень. Виконано дослідження її властивостей шляхом введення категоріальної дедукції. Введено поняття семантичної категорійної стратегії для одержання оператора визначення фіксованого значення із ефективними властивостями і оператора безпосереднього слідування, що розглядається як функтор на множині інтерпретацій.

Подальші дослідження даного напрямку будуть зосереджені на розширенні одержаної категорійної моделі модифікаційних предикатних запитів та побудови її коректних імплементацій.

Список літератури: 1. *Burhans D., Shapiro S.* Expanding the notion of answer in rule-based systems / Technical Report 99- 07 // Department of Computer Science and Engineering, SUNY Buffalo. November 1999. 155p. 2. *Comini M., Levi G., Meo M., Vitiello G.* Abstract diagnosis // Journal of Logic Programming, №39 (1 – 3). 1999. P.43 – 93. 3. *Comini M., Meo M.* Compositionality properties of SLD-derivations // Theoretical Computer Science. №211(1&2). 1999. P.275 – 309. 4. *Cousot P., Cousot R.* Temporal abstract interpretation // In Conference Record of the 27 Annual ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages. Boston, USA. January 2000. ACM Press, New York, NY. P. 12 – 25 5. *Gabbrielli M., Levi G., Meo M.* Resultants semantics for PROLOG // Journal of Logic and Computation. №6(4). 1996.P. 491 – 521. 6. *Jacobs B.* Categorical Logic and Type Theory // Studies in Logic and the Foundations of Mathematics. North Holland, Elsevier. 1999. 325p. 7. *Lipton J., McGrail R.* Encapsulating data in logic programming via categorical constraints / In Palamidessi C., Glaser H., Meinke K. Editors // Principles of Declarative Programming. Volume 1490 of Lecture Notes in Computer Science. Springer Verlag, Berlin. 1998. P.391 – 410. 8. *Maleusieux F., Ridoux O., Boizumault P.* Abstract compilation of Prolog / In Jaar J. Editor // Joint International Conference and Symposium on Logic Programming. Manchester, United Kingdom. June 1998. MIT Press. P.130 – 144. 9. *Power J., Robinson E.* Premonoidal categories and notions of computation // Mathematical Structures in Computer Science. № 7(5). October 1997. P. 453 – 468. 10. *Corradini A., Asperti A.* A categorical model for logic programs: Indexed monoidal categories / In Proceedings REX Workshop '92 // Springer Lectures Notes in Computer Science. 1992. P. 5 – 36. 11. *Corradini A., Montanari U.* An algebraic semantics for structured transition systems and its application to logic programs // Theoretical Computer Science. №103(1). August 1992. P.51 – 106. 12. *Barbuti R., Giacobazzi R., Levi G.* A General Framework for Semantics-based Bottom-up Abstract Interpretation of Logic Programs / / ACM Transactions on Programming Languages and Systems. № 15(1). 1993. P. 133 – 181. 13. *Finkelstein S., Freyd P., Lipton J.* Logic programming in tau categories // In Computer Science Logic '94. Vol. 933 of Lecture Notes in Computer Science. Springer Verlag, Berlin. 1995. P. 249 – 263. 14. *Шекета В.І.* Модифікаційні предикатні запити / Науковий журнал «Проблеми програмування» Інституту Програмних Систем НАН України. 2004. №2 – 3. С.339 – 343 // Спеціальний випуск за матеріалами 4-ї МНПК "УкрПрог'2004", 1 – 3 червня 2004. Київ, Кібернетичний центр НАН України. 15. *Шекета В.І.* Ініціалізація еластичних семантик над простором Гербранда для модифікаційних предикатних запитів // Міжнародний науково-технічний журнал «Вимірювальна та обчислювальна техніка в технологічних процесах". Хмельницький. 2003 № 2(22). С.13 – 18. 16. *Шекета В.І.* Аналіз семантики шаблонів виклику модифікаційних предикатних запитів для інформаційних систем на основі баз даних і знань // Комп'ютерна інженерія та інформаційні технології. Вісник нац. університету "Львівська політехніка". Львів. 2003. № 496. С.217 – 228.

Надійшла до редколегії 24.11.2004

АВТОРЫ ВЫПУСКА

Альхалайбех Айман	– аспирант ХНУРЭ
Безрук Валерий Михайлович	– д-р техн. наук, доц. каф. сети связи, ХНУРЭ
Бессонов Александр Александрович	– стаж.-иссл. каф. ЭВМ, ХНУРЭ
Бобух Всеволод Анатольевич	– аспирант каф. ЭВМ, ХНУРЭ
Бондаренко Михаил Федорович	– д-р техн. наук, проф., ректор ХНУРЭ
Бурцев Валерий Николаевич	– канд. техн. наук, зам. директора по научной работе НПФ «Цветодинамика».
Ваде Гриби	– аспирант каф. автоматизации производства ВТ, ХНУРЭ
Вечирская Ирина Дмитриевна	– аспирантка каф. программного обеспечения ЭВМ, ХНУРЭ
Горшков Евгений Витальевич	– аспирант каф. искусственного интеллекта, ХНУРЭ
Гребенник Игорь Валериевич	– канд. физ.-мат. наук, доц. каф. системотехники, ХНУРЭ
Дударь Зоя Владимировна	– канд. техн. наук, проф., и. о. зав. каф. программного обеспечения ЭВМ, ХНУРЭ
Елисеев Владимир Васильевич	– канд. техн. наук, генер. директор НИИ УВМ, г. Северодонецк
Ерохин Андрей Леонидович	– канд. техн. наук, доц. каф. программного обеспечения ЭВМ, ХНУРЭ
Иванилов Артем Александрович	– аспирант каф. программного обеспечения ЭВМ, ХНУРЭ
Киричук Елена Павловна	– аспирантка каф. программного обеспечения компьютерных систем, НАУ им. Н.Е. Жуковского «ХАИ»
Коваленко Новомир Петрович	– младш. научн. сотрудник каф. сети связи, ХНУРЭ
Колодяжный Виталий Владимирович	– канд. техн. наук, ст. научн. сотрудник ПНИЛ АСУ ХНУРЭ
Лещинский Владимир Александрович	– аспирант каф. программного обеспечения ЭВМ, ХНУРЭ
Лысенко Виктор Андреевич	– вед. инженер Научно-исслед. института радиотехн. измерений, г. Харьков
Ляшенко Сергей Алексеевич	– канд. техн. наук, доц. каф. безопасности жизнедеятельности, ХНТУ сельского хозяйства
Михаль Олег Филиппович	– канд. техн. наук, доц. каф. ЭВМ, ХНУРЭ
Петров Константин Эдуардович	– канд. техн. наук, доц. каф. прикладной математики, Национальный университет внутренних дел МВД Украины
Плисс Ирина Павловна	– канд. техн. наук, вед. научн. сотрудник ПНИЛ АСУ ХНУРЭ
Полунин Роман Алексеевич	– аспирант, асс. каф программного обеспечения ЭВМ, ХНУРЭ
Руденко Олег Григорьевич	– д-р техн. наук, проф., зав. каф. ЭВМ, ХНУРЭ
Сироджа Игорь Борисович	– д-р техн. наук, проф., зав. каф. программного обеспечения компьютерных систем, НАУ им. Н.Е. Жуковского «ХАИ»
Сотников Платон Олегович	– аспирант каф. ЭВМ, ХНУРЭ
Тесленко Наталья Александровна	– соискатель ПНИЛ АСУ ХНУРЭ, ст. лаб. каф. философии, ХНУРЭ
Хассан Ктейман	– аспирант каф. автоматизации производства ВТ, ХНУРЭ
Хаханов Владимир Иванович	– д-р техн. наук, проф. каф. автоматизации производства ВТ, ХНУРЭ
Хаханова Ирина Витальевна	– канд. техн. наук, доц. каф. автоматизации производства ВТ, ХНУРЭ
Четвериков Григорий Григорьевич	– канд. техн. наук, доц. каф. программного обеспечения ЭВМ, ХНУРЭ
Чикина Валентина Алексеевна	– канд. техн. наук, вед. научн. сотр. каф. программного обеспечения ЭВМ ХНУРЭ
Шабанов-Кушнаренко Юрий Петрович	– д-р техн. наук, проф. каф. программного обеспечения ЭВМ, ХНУРЭ
Шекета Василий Иванович	– канд. техн. наук, доц. каф. программного обеспечения, Ивано-Франковский национальный технический университет нефти и газа

УДК 519.7

О бионике интеллекта / М.Ф. Бондаренко, Ю.П. Шабанов-Кушнаренко // Бионика интеллекта: научн.-техн. журнал. 2004. № 61/1. С. 3–14.

Общая характеристика бионики интеллекта как научного направления.

Ил. 13. Библиогр.: 6 назв.

УДК 519.7

Про біоніку інтелекту / М.Ф. Бондаренко, Ю.П. Шабанов-Кушнаренко // Біоніка інтелекту: наук.-техн. журнал. 2004. № 61/1. С. 3–14.

Загальна характеристика біоніки інтелекту як наукового напрямку.

Іл. 13. Бібліогр.: 6 назв.

UDC 519.7

About Bionics of Intelligence /M.F. Bondarenko, Yu.P. Shabanov-Kushnarenko // Bionica Intellecta: Sci. Mag. 2004. № 61/1. P. 3–14.

Bionics of intelligence general characteristic as scientific direction.

Fig. 13. Ref.: 6 items.

УДК 519.7

Об алгебре предикатов / М.Ф. Бондаренко, Ю.П. Шабанов-Кушнаренко // Бионика интеллекта: научн.-техн. журнал. 2004. № 61/1. С. 15–26.

Дана характеристика алгебры предикатов как эффективного средства формального описания объектов бионики интеллекта.

Табл. 12. Ил. 4.

УДК 519.7

Про алгебру предикатів / М.Ф. Бондаренко, Ю. П. Шабанов-Кушнаренко // Біоніка інтелекту: наук.-техн. журнал. 2004. № 61/1. С. 15–26.

Дано характеристику алгебри предикатів як ефективного засобу формального опису об'єктів біоніки інтелекту.

Табл. 12. Іл. 4.

UDC 519.7

About Algebra of Predicates / M.F. Bondarenko, Yu. P. Shabanov-Kushnarenko // Bionica Intellecta: Sci. Mag. 2004. № 61/1. P. 15–26.

The characteristic of algebra of predicates as effective means of bionics of intelligence objects formal description is given.

Tab. 12. Fig. 4.

УДК 519.7

Модели языка /М.Ф. Бондаренко, В.А. Чикина, Ю.П. Шабанов-Кушнаренко // Бионика интеллекта: научн.-техн. журнал. 2004. № 61/1. С. 27–37.

Введение в моделирование механизмов естественного языка средствами и методами бионики интеллекта.

Табл.14. Ил.18. Библиогр.: 6 назв.

УДК 519.7

Моделі мови /М.Ф. Бондаренко, В.О. Чікіна, Ю.П. Шабанов-Кушнаренко // Біоніка інтелекту: наук.-техн. журнал. 2004. № 61/1. С. 27–37.

Уведення в моделювання механізмів природної мови засобами та методами біоніки інтелекту.

Табл.14. Іл.18. Бібліогр.: 6 назв.

UDC 519.7

Models of Language /M.F. Bondarenko, V.A. Chikina, Ju.P. Shabanov-Kushnarenko // Bionica Intellecta: Sci. Mag. 2004. № 61/1. P. 27–37.

Introduction to the mehanismes natural language by means and methods of the bionics intelligence.

Tab.14. Fig.18. Ref.: 6 items.

УДК 519.7

Линейные логические операторы в виде схем и графов / И.Д. Вечирская, З.В. Дударь, А.А. Иванилов, В.А. Лещинский // Бионика интеллекта: научн.-техн. журнал. 2004. № 61/1. С. 38–41.

В данной работе рассматривается идея представления линейных логических преобразований в виде схем и графов. Приведен пример реализации бинарного отношения переключательной схемой. Исследован подход к переходу на линейные логические преобразования.

Ил. 5. Библиогр.: 8 назв.

УДК 519.7

Лінійні логічні оператори у вигляді схем та графів / І.Д. Вечірська, З.В. Дудар, А.О. Іванілов, В.О. Лещинський // Біоніка інтелекту: наук.-техн. журнал. 2004. № 61/1. С. 38–41.

В даній роботі розглянута ідея подання лінійних логічних перетворень у вигляді схем та графів. Наведено приклад реалізації бінарного відношення схемою перемикання. Досліджено підхід до переходу на лінійні логічні перетворення.

Іл. 5. Бібліогр.: 8 назв.

УДК 519.7

Linear Logical Operators As Circuits And Graphs / I.D. Vechirska, Z.V. Dudar, A.A. Ivanilov, V.A. Leshynskiy // Bionica Intellecta: Sci. Mag. 2004. № 61/1. P. 38–41.

The idea of representation of linear logical transformations as circuits and graphs is considered in this paper. An example of binary relation realization with switching circuits is given. The transition to linear logical transformations is studied.

Fig. 5. Ref.: 8 items.

УДК 519.62

Алгебро-логічні засоби моделювання природної мови (Повідомлення 1) / Г.Г. Четвериков // Біоніка інтелекту: наук.-техн.журнал. 2004. № 61/1. С. 42–45.

У статті розглядаються перспективні напрямки розвитку сучасних цифрових пристроїв, мереж та систем. Стверджується, що розвиток засобів обчислювальної техніки є базою автоматизації розумової діяльності людини. Проте, успіхи в галузі інтелектуалізації обчислювальної техніки незначні. Наведено, що технічна реалізація концепції штучного інтелекту вимагає значного підвищення продуктивності ЕОМ, а саме: розробки паралельних (просторових) принципів та методів обробки інформації на рівні інтерфейсу для сприйняття і видачі інформації на природній мові.

Бібліогр.: 5 назв.

УДК 519.62

Алгебро-логические средства моделирования естественного языка (Сообщение1) / Г.Г. Четвериков // Бионика интеллекта: наук.-техн.журнал. 2004. № 61/1. С. 42–45.

В статье рассматриваются перспективные направления развития современных цифровых устройств, сетей и систем. Проведенный анализ показывает, что развитие средств вычислительной техники является основой автоматизации умственной деятельности человека. Однако успехи в области интеллектуализации вычислительной техники невелики. Утверждается и обосновывается, что техническая реализация концепции искусственного интеллекта требует существенного повышения производительности ЭВМ, а именно: разработки параллельных (пространственных) принципов и методов обработки информации на уровне интерфейса для восприятия и выдачи информации на естественном языке.

Библиогр.: 5 назв..

UDK 519.62

Algebro-Logical Tools of Natural Language Modeling (Report 1) / G.G. Chetverikov // Bionica Intellecta: Sci. Mag. 2004. № 61/1. P. 42–45.

Perspective directions of modern digital devices, networks and systems development are considered in this article. The conducted analysis shows that computer facilities development is an automation basis of human intellectual activity. However, the achievements in the computer facilities intellectualization field are insignificant. Technical realization of the artificial intelligence concept is claimed and substantiated to require essential increase in computer productivity, namely: the creation of parallel (spatial) principles and methods of the information processing at the interface level for information perception and delivery in natural language.

Ref.: 5 items.

УДК 621.397.331

Об одном алгоритме синтеза перспективных трансформаций телевизионных изображений на основе аффинных преобразований /О.Г. Руденко, П.О. Сотников // Бионика интеллекта: научн.-техн. журнал. 2004. № 61/1. С. 46–48.

Предлагается алгоритм синтеза перспективных проекций телевизионных изображений. Отличительной особенностью алгоритма является то, что он основан на аффинных преобразованиях. Кроме того, алгоритм является аппаратно-ориентированным и позволяет осуществлять трансформацию телевизионных изображений в реальном времени.

Ил. 2. Библиогр.: 2 назв.

УДК 621.397.331

Про один алгоритм синтезу перспективних трансформацій телевізійних зображень на основі аффінних перетворень / О.Г. Руденко, П.О. Сотников // Біоніка інтелекту: наук.-техн. журнал. 2004. № 61/1. С. 46–48.

Пропонується алгоритм синтезу перспективних проекцій телевізійних зображень. Відмінною рисою алгоритму є те, що він заснований на афінних перетвореннях. Крім того, алгоритм є апаратно-орієнтованим і дозволяє здійснювати трансформацію телевізійних зображень у реальному часі.

Іл. 2. Бібліогр.: 2 назв.

UDC 621.397.331

About An Algorithm of Television Image Perspective Transformation Synthesizing Based On Affine Transformations / O.G. Rudenko, P.O. Sotnikov // Bionica Intellecta: Sci. Mag. 2004. № 61/1. P. 46–48.

The algorithm of television image perspective projections synthesizing is proposed. The distinctive feature of this algorithm is that it is based on affine transformations. Moreover, the algorithm is device-directed and allows executing television image transformations in real time.

Fig. 2. Ref.: 2 items.

УДК 519.71

Применение нейронной сети CMAC при решении задач обработки изображений и ее аппаратная реализация / А.А. Бессонов, В.А. Бобух, О.Г. Руденко // Бионика интеллекта: научн.-техн. журнал. 2004. № 61/1. С. 49–55.

Рассматривается традиционная и новая схемы нейронной сети CMAC. Исследуются вопросы эффективного выбора ее параметров. Для решения задачи фильтрации изображений предлагается иерархическая схема CMAC. Эффективность данной нейронной сети демонстрируется путем имитационного моделирования.

Ил. 6. Библиогр.: 8 назв.

УДК 519.71

Застосування нейронної мережі CMAC при вирішенні задач обробки зображень та її апаратна реалізація / О.О. Безсонов, В. А. Бобух, О.Г. Руденко // Біоніка інтелекту: наук.-техн. журнал. 2004. № 61/1. С. 49–55.

Розглядається традиційна та нова схеми нейронной мережі CMAC. Вивчаються питання ефективного вибору її параметрів. Для вирішення задачі фільтрації зображень пропонується ієрархічна схема CMAC. Ефективність даної нейронної мережі демонструється шляхом імітаційного моделювання.

Іл. 6. Бібліогр.: 8 назв.

UDC 519.71

Application of An Neural Network CMAC At Problem Solving of Image Processing And Its Hardware Realization / A.A. Bessonov, V.A. Bobukh, O.G. Rudenko // Bionica Intellecta: Sci. Mag. 2004. № 61/1. P. 49–55.

The traditional and new schemes of CMAC neural network are presented. The adjustment of network parameters is discussed. A hierarchical CMAC for an image filtering is proposed. The utility of the CMAC neural networks in image and signal filtering is illustrated with some simulated examples.

Fig. 6. Ref.: 8 items.

УДК 004.8

Интеллектуальная система поддержки принятия технологических решений при выборе оборудования в технологической подготовке авиационного производства / Е.П. Киричук, И.Б. Сироджа // Бионика интеллекта: научн.-техн. журнал. 2004. № 61/1. С. 56–60.

В настоящее время остро стоит проблема обеспечения качественной и оперативной технологической подготовки авиационного производства. Выбор оборудования характеризуется как плохо формализованная задача. Поэтому создание интеллектуальной системы поддержки принятия технологических решений по выбору оборудования, в основе которой лежит квантовый подход к представлению знаний, является актуальной задачей.

Ил. 7. Библиогр.: 6 назв.

УДК 004.8

Інтелектуальна система підтримки прийняття технологічних рішень з вибору обладнання під час технологічної підготовки авіаційного виробництва / О.П. Киричук, І.Б. Сіроджа // Біоніка інтелекту: наук.-техн. журнал. 2004. № 61/1. С. 56–60.

Наразі гостро стоїть проблема забезпечення якісної і оперативної технологічної підготовки авіаційного виробництва. Вибір обладнання характеризується як погано формалізована задача. Тому створення інтелектуальної системи підтримки прийняття технологічних рішень з вибору обладнання, в основі якої лежить квантовий підхід до представлення знань, є актуальною задачею.

Іл. 7. Бібліогр.: 6 назв.

UDC 004.8

Intellectual Technological Decision-Making Support System For Equipment Choosing In Technological Preparing of Aircraft Production / E.P. Kirichuk, I.B. Sirodzha // Bionica Intellecta: Sci. Mag. 2004. № 61/1. P. 56–60.

Nowadays the problem of high-quality and effective technological preparation of aircraft production exists. Equipment selection is characterized as poor formalized task. Therefore creation of intellectual technological decision-making support system for equipment selection based on quantum approach for knowledge presentation is an urgent task.

Fig. 7. Ref.: 6 items.

УДК 519.713:681.326

Анализ неисправностей при проектировании дискретных логических сетей / Ваде Гриби, В.В. Елисеев, Хассан Ктейман, В.И. Хаханов, И.В. Хаханова // Бионика интеллекта: научн.-техн. журнал. 2004. № 61/1. С. 61–67.

Предложен быстродействующий дедуктивный метод обратного моделирования неисправностей, ориентированный на обработку цифровых проектов вентильного и регистрового уровней описания. Быстродействие программной реализации на порядок выше, чем у методов прямого транспортирования списков дефектов.

Ил. 9. Библиогр.: 12 назв.

УДК 519.713:681.326

Аналіз несправностей при проектуванні дискретних логічних мереж / Ваде Грібі, В.В. Єлисеєв, Хасан Ктейман, В.І. Хаханов, І.В. Хаханова // Біоніка інтелекту: наук.-техн. журнал. 2004. № 61/1. С. 61–67.

Запропоновано швидкодіючий дедуктивний метод зворотного моделювання несправностей, орієнтований на обробку цифрових проектів вентильного та реєстрового рівнів опису. Швидкодія програмної реалізації на порядок вище, ніж у методів прямого транспортування списків дефектів.

Іл. 9. Бібліогр.: 12 назв.

UDC 519.713:681.326

Fault Analysis At Discret Logical Network Designing / Wade Ghribi, V.V. Yeliseyev, Hassan Ktiaman, V.I. Hahanov, I.V. Hahanova // Bionica Intellecta: Sci. Mag. 2004. № 61/1. P. 61–67.

Fast deductive method of fault reverse modeling oriented for processing digital projects of gate or RTL description level is offered. Processing speed of program realization is 10 times higher than in methods with forward propagation algorithms.

Fig. 9. Ref.: 12 items.

УДК 004.89

Об интеллектуализации процесса поддержки жизненного цикла программных систем / А.Л. Ерохин, Р.А. Полунин / / Бионика интеллекта: научн.-техн. журнал. 2004. № 61/1.С. 68–73.

Обсуждаются проблемы информационного обеспечения процессов поддержки жизненного цикла программных систем. Предложен новый подход к интеллектуализации задач информационного обеспечения. Рассмотрена база знаний информационного обеспечения как отдельного процесса, изучаются ее структура, принципы наполнения, сопровождения, использования.

Ил. 3. Библиогр.: 6 назв.

УДК 004.89

Про інтелектуалізацію процесу підтримки життєвого циклу програмних систем / А.Л. Єрохін, Р.О. Полунін // Біоніка інтелекту: наук.-техн. журнал. 2004. № 61/1. С. 68–73.

Обговорюються проблеми інформаційного забезпечення процесів підтримки життєвого циклу програмних систем. Запропоновано новий підхід до інтелектуалізації задач інформаційного забезпечення. Розглянуто базу знань інформаційного забезпечення як окремого процесу, досліджуються її структура, принципи наповнення, супроводження, використання.

Іл. 3. Бібліогр.: 6 назв.

UDC 004.89

About Intelligent Automation of Software Systems' Lifecycle Activities / A.L. Yerokhin, R.A. Polunin // Bionica Intellecta: Sci. Mag. 2004. № 61/1. P. 68–73.

Some problems of information-related activities for software systems' lifecycle are considered in the article. A new approach for intelligent automation of information-related activities is presented. A knowledge base for information-related activities as a separate process is presented; its structure, filling principles, maintenance and application are discussed.

Fig. 3. Ref.: 6 items.

УДК 681.518:004.912

Формализация сложноорганизованных систем и распознавание аварийных ситуаций. Сообщение 1 / В.Н. Бурцев, А.Л. Ерохин // Бионика интеллекта: научн.-техн. журнал. 2004. № 61/1. С. 74–77.

Рассмотрен первый этап решения задачи построения способа отображения множества параметров сложноорганизованной системы, позволяющий повысить эффективность идентификации аварийных ситуаций. Предложена модель взаимодействия метрических и нормированных параметров системы со стохастическими возмущениями.

Іл.2. Библиогр.: 12 назв.

УДК 681.518:004.912

Формалізація складноорганізованих систем і розпізнавання аварійних ситуацій. Повідомлення 1 / В.М.Бурцев, А.Л.Єрохін // Біоніка інтелекту: наук.-техн. журнал. 2004. № 61/1.С. 74–77.

Розглядається перший етап розв'язання задачі побудови способу відображення множини параметрів складноорганізованої системи, який дозволяє підвищити ефективність ідентифікації аварійних ситуацій. Запропоновано модель взаємодії метричних та нормованих параметрів системи зі стохастичними впливами.

Іл.2. Бібліогр.: 12 назв.

UDC 681.518:004.912

Formalization of Complex-Organized Systems And Emergency Recognition. Report 1 / V.N. Burtsev, A.L. Yerokhin // Bionica Intellecta: Sci. Mag. 2004. № 61/1. P. 74–77.

The first stage in a task solution of building a method of parameters set display in complex-organized systems is considered; this allows increasing in emergencies identification efficiency. The interaction model of metric and normalized parameters of a system with stochastic indignations is offered.

Fig. 2. Ref.: 12 items.

УДК 681.513

Адаптивный алгоритм обучения нео-фаззи нейрона и нейро-фаззи сети на основе полиномиальных функций принадлежности / Е.В. Горшков, В.В. Колодяжный, И.П. Плисс // Бионика интеллекта: научн.-техн. журнал. 2004. № 61/1. С. 78–81.

В работе рассмотрены архитектуры нео-фаззи нейрона и нейро-фаззи сети Ванга-Менделя с использованием в качестве функций принадлежности полиномов 4-й степени. Предложен обобщенный алгоритм обучения рассматриваемых архитектур, настраивающий синаптические веса и параметры функций принадлежности. Алгоритм обладает вычислительной простотой, высокой скоростью сходимости, а также следящими и сглаживающими свойствами.

Ил. 3. Библиогр.: 8 назв.

УДК 681.513

Адаптивний алгоритм навчання нео-фаззі нейрона та нейро-фаззі мережі на основі поліноміальних функцій приналежності / Є.В. Горшков, В.В. Колодяжний, І.П. Плісс // Біоніка інтелекту: наук.-техн. журнал. 2004. № 61/1. С. 78–81.

В роботі розглянуто архітектури нео-фаззі нейрона та нейро-фаззі мережі Ванга-Менделя з використанням в якості функцій приналежності поліномів 4-го ступеня. Запропоновано узагальнений алгоритм навчання розглянутих архітектур, який настроює синаптичні ваги та параметри функцій приналежності. Алгоритму властиві обчислювальна простота, висока швидкість збіжності, а також слідкуючими та згладжуючими властивостями.

Іл. 3. Бібліогр.: 8 назв.

UDC 681.513

Adaptive Learning Algorithm For A Neo-Fuzzy Neuron And Neuro-Fuzzy Network Based On A Polynomial Membership Functions / Ye.V. Gorshkov, V.V. Kolodyaznhiy, I.P. Pliss // Bionica Intellecta: Sci. Mag. 2004. № 61/1. P. 78–81.

A neo-fuzzy neuron and Wang-Mendel neuro-fuzzy network architectures with polynomial activation functions of 4-th order are considered. A new generalized learning algorithm for the considered networks which adjusts both synaptic weights and membership functions is proposed. The learning algorithm possesses computational simplicity, high rate of convergence, tracking, and smoothing properties.

Fig. 3. Ref.: 8 items.

УДК 621.396.961.1

Об одном методе автоматизированного распознавания стадий сна по электроэнцефалограммам / В.М. Безрук, Н.П. Коваленко, В.А. Лысенко // Бионика интеллекта: научн-техн. журнал. 2004. № 61/1. С. 82–85.

Предложен и исследован новый метод автоматизированного распознавания заданных стадий сна по ЭЭГ при наличии класса неизвестных наблюдений. Метод основан на описании ЭЭГ математической моделью в виде авторегрессионных процессов.

Табл. 1. Ил. 4. Библиогр.: 6 назв.

УДК 621.396.961.1

Про один метод автоматизованого розпізнавання стадій сну за електроенцефалограмами / В.М. Безрук, Н.П. Коваленко, В.О. Лисенко // Біоніка інтелекту: научн-техн. журнал. 2004. № 61/1. С. 82–85.

Запропоновано та досліджено новий метод автоматизованого розпізнавання заданих стадій сну за ЕЕГ при наявності класу невідомих спостережень. Метод основано на описі ЕЕГ математичною моделлю у вигляді авторегресійних процесів.

Табл. 1. Іл. 4. Бібліогр.: 6 назв.

UDC 621.396.961.1

About One Method of Automated Recognition of Set Stages of Sleep On Electroenchefalogram's / V.M. Besruk, N.P. Kovalenko, V.A. Lisenko // Bionica Intellecta: Sci. Mag. 2004. № 61/1. P. 82–85.

A new method of automated recognition of sleep set stages with EEG at presence of class of unknown supervisions is offered and explored. A method is based on description of EEG by mathematical model as autoregression processes.

Tab. 1. Fig. 4. Ref.: 6 items.

УДК 519.08

Проблема формализации интеллектуальной деятельности человека / К.Э. Петров // Бионика интеллекта: научн.-техн. журнал. 2004. № 61/1. С. 86–89.

Предложены подходы к структурной и параметрической идентификации моделей интеллектуальной деятельности, основанные на теории компараторной идентификации. Рассмотрены основные трудности, возникающие в ходе решения этой проблемы. Показано, что решение данной задачи связано с определением модели оптимальной сложности в классе полинома Колмогорова-Габора.

Библиогр.: 6 назв.

УДК 519.08

Проблема формалізації інтелектуальної діяльності людини / К.Е. Петров // Біоніка інтелекту: наук.-техн. журнал. 2004. № 61/1. С. 86–89.

Запропоновано підходи до структурної та параметричної ідентифікації моделей інтелектуальної діяльності, яки засновані на теорії компараторной ідентифікації. Розглянуто основні труднощі, що виникають в ході вирішення цієї проблеми. Показано, що розв'язання даної задачі пов'язано з визначенням моделі оптимальної складності в класі полінома Колмогорова-Габора.

Бібліогр.: 6 назв.

UDC 519.08

The Task of Human Intellectual Activity Formalization / K.E. Petrov // Bionica Intellecta: Sci. Mag. 2004. № 61/1. P. 86–89.

Approaches to structural and parametrical identification of intellectual activity models based on the comparative identification theory are offered. The basic obstacles that appear during the problem solution are considered. The given problem solution is shown to be connected with definition of optimum complexity model in a class of Kolmogorov-Gabors polynom.

Ref.: 6 items.

УДК 519.6:514.1

Модели оптимизации на композиционных образах комбинаторных множеств в системах поддержки принятия решений / И.В. Гребенник // Бионика интеллекта: научн.-техн. журнал. 2004. № 61/1. С. 90–97.

Исследуются модели экстремальных задач на специальных классах комбинаторных множеств. Приводится постановка и решение ряда задач комбинаторной оптимизации на образах интервальных комбинаторных множеств в евклидовом пространстве. Описывается применение полученных результатов при разработке интеллектуальных систем поддержки принятия решений в геометрическом проектировании.

Библиогр.: 12 назв.

УДК 519.6:514.1

Моделі оптимізації на композиційних образах комбінаторних множин в системах підтримки прийняття рішень / І.В.Гребеннік // Біоніка інтелекту: наук.-техн. журнал. 2004. № 61/1. С. 90–97.

Досліджуються моделі екстремальних задач на спеціальних класах комбінаторних множин. Наводиться постановка та розв'язання ряду задач комбінаторної оптимізації на образах інтервальних комбінаторних множин в евклідовому просторі. Описується застосування отриманих результатів при розробці інтелектуальних систем підтримки прийняття рішень в геометричному проектуванні.

Бібліогр.: 12 назв.

UDC 519.6:514.1

Optimization Models On Composition Images of Combinatorial Sets In Decision Making Systems / I.V. Grebennik // Bionica Intellecta: Sci. Mag. 2004. № 61/1. P. 90–97.

Models of extreme tasks on special classes of combinatorial sets are studied. The formulation and decision of a number of combinatorial optimization tasks on images of interval combinatorial sets in Euclidean space are given. Application of obtained results at creation of intellectual decision making systems in geometric design is described.

Ref.: 12 items.

УДК 519.87

Моделирование на сетях Петри информационного обмена между системами с ограниченным ресурсом готовности / Альхалайбех Айман, О.Ф. Михаль // Бионика интеллекта: научн.-техн. журнал. 2004. № 61/1. С. 98–102.

При проектировании технических систем представляют интерес оценки роста сложности систем при увеличении числа информационных связей с учетом ограниченного ресурса готовности обработки поступающей информации. С помощью моделей на сетях Петри исследован рост сложности систем при изменении числа каналов, информационных шин и обратных связей. Разработаны соответствующие принципы моделирования и иерархические линии моделей. Ценность полученных результатов выходит за рамки технических приложений. Результаты применимы также к экономическим и административным системам.

Табл. 1. Ил. 6. Библиогр.: 4 назв.

УДК 519.87

Моделювання на мережах Петрі інформаційного обміну між системами з обмеженим ресурсом готовності / Альхалайбех Айман, О.П. Міхаль // Біоніка інтелекту: наук.-техн. журнал. 2004. № 61/1. С. 98–102.

При проектуванні технічних систем цікаво мати оцінки зростання складності систем при збільшенні кількості інформаційних зв'язків з урахуванням обмеженого ресурсу готовності обробки інформації, що надходить. За допомогою моделей на мережах Петрі досліджено зростання складності систем при зміні кількості каналів, інформаційних шин та зворотніх зв'язків. Розроблено відповідні принципи моделювання та ієрархічні лінії моделей. Цінність здобутих результатів виходить за межі технічних застосувань. Результати можуть бути застосовані також до економічних та адміністративних систем.

Табл. 1. Іл. 6. Бібліогр.: 4 назв.

UDC 519.87

Modeling of Information Exchange Between Systems With Limited Resource of Processing on Petri Network / Alhalaybeh Ayman, O.Ph. Mikhal // Bionica Intellecta: Sci. Mag. 2004. № 61/1. P. 98–102.

At technical systems design the estimations of increase in systems complexity at increase in number of information relationships with limited resources of entered information processing readiness are of interest. With models on Petri network an increase in systems complexity at alteration of channels, information buses and feedbacks number is studied. Appropriate modeling principles and model hierarchical lines are designed. Results may be applicable in economic and administrative systems as well.

Tab. 1. Fig. 6 Ref.: 4 items.

УДК 681.513

Алгоритм обучения автоассоциативной искусственной многослойной нейронной сети /Н.А. Тесленко // Бионика интеллекта: научн.-техн. журнал. 2004. № 61/1. С. 103–106.

Предложен алгоритм обучения многослойной нейронной сети на основе полиномиальной функции активации для задач компрессии информации. Данный алгоритм отличается численной простотой реализации и обеспечивает более высокую скорость сходимости.

Ил. 4. Библиогр.: 7 назв.

УДК 681.513

Алгоритм навчання автоасоціативної штучної багатошарової нейтронної мережі / Н.О. Тесленко // Біоніка інтелекту: наук.-техн. журнал. 2004. № 61/1. С. 103–106.

Запропоновано алгоритм навчання багатошарової нейронної мережі на базі поліноміальної функції активації для задач компресії інформації. Цей алгоритм відрізняється чисельною простотою реалізації та забезпечує підвищену швидкість збіжності.

Іл. 4. Бібліогр.: 7 назв.

UDC 681.513

Learning Algorithm of Autoassociative Artificial Multilayer Neural Network / N.A. Teslenko // Bionica Intellecta: Sci. Mag. 2004. № 61/1. P. 103–106.

In this paper the learning algorithm for multilayer neural network based on polynomial activation function for information compression problems is proposed. Given algorithm has numerical simplicity of realization and provides higher rate of convergence.

Fig. 4. Ref.: 7 items.

УДК 519.71

Об одном алгоритме обучения нейросетевой модели нелинейного динамического объекта / С.А. Ляшенко // Бионика интеллекта: научн.-техн. журнал. 2004. № 61/1. С. 107–110.

Предлагается многошаговый рекуррентный алгоритм обучения нестационарных параметров нейросетевой модели нелинейного динамического объекта. Разработан адаптивный алгоритм, использующий для коррекции оценок ограниченное число измерений. Эффективность процедуры обучения демонстрируется путем имитационного моделирования.

Ил.3. Библиогр.: 12 назв.

УДК 519.71

Про один алгоритм навчання нейромережевої моделі нелінійного динамічного об'єкта / С.О. Ляшенко // Біоніка інтелекту: научн.-техн. журнал. 2004. № 61/1. С. 107–110.

Пропонується багатокроковий рекурентний алгоритм навчання нестаціонарних параметрів нейромережевої моделі нелінійного об'єкта. Розроблено адаптивний алгоритм, який використовує для оцінок обмежену кількість вимірювань. Ефективність процедури навчання демонструється шляхом імітаційного моделювання.

Іл.3. Бібліогр.: 12 назв.

UDC 519.71

About A Learning Algorithm of Neural Network Model of Nonlinear Dynamic System / S.A. Lyashenko // Bionica Intellecta: Sci. Mag. 2004. № 61/1. P. 107–110.

A new multistep recursive learning algorithm for nonstationary parameters of neural network model of nonlinear dynamic systems is proposed. Adaptive algorithm that uses limited member of measurement for error correction is developed. Learning efficiency are demonstrated with simulated examples.

Fig. 3. Ref.: 12 items.

УДК 681. 3:622. 276

Построение модели модификационных предикатных запросов на основе синтаксической категорийной стратегии / В.И.Шекета // Бионика интеллекта: научн.-техн. журнал. 2004. № 61/1. С. 111–115.

Предложена категориальная модель модификационных предикатных запросов на основе начальной интерпретации для синтаксической категорийной стратегии, каждый слой которой представляет множество категорийных дедукций, которые могут быть выполнены. В рамках денотационной семантики введено множество динамических целей для выполнения модификации конструкций для фиксированных значений. Построены функторные интерпретации, отображающие полученные синтаксические и семантические категории. Введено понятие семантической категорийной стратегии для получения оператора определения фиксированного значения с эффективными свойствами и оператора непосредственного следования, который рассматривается как функтор на множестве интерпретаций.

Библиогр.: 16 назв.

УДК 681. 3:622. 276

Побудова моделі модифікаційних предикатних запитів на основі синтаксичної категорійної стратегії / В.І.Шекета // Біоніка інтелекту: наук.-техн. журнал. 2004. № 61/1. С. 111–115.

Запропоновано категоріальну модель модифікаційних предикатних запитів на основі початкової інтерпретації для синтаксичної категорійної стратегії, кожний шар якої представляє множину категорійних дедукцій, які можуть бути виконані. В рамках денотаційної семантики введено множину динамічних цілей для виконання покрочної модифікації конструкцій для фіксованих значень. Побудовано функторні інтерпретації, що відображають одержані синтаксичні і семантичні категорії. Введено поняття семантичної категорійної стратегії для одержання оператора визначення фіксованого значення із ефективними властивостями і оператора безпосереднього слідування, що розглядається як функтор на множині інтерпретацій.

Бібліогр.: 16 назв.

UDC 681. 3:622. 276

Construction of A Modification Predicate Queries Model On The Basis of Syntactic Categorical Strategy / V.I.Sheketa // Bionica Intellecta: Sci. Mag. 2004. № 61/1. P. 111–115.

The categorical model for modifications predicate queries on the basis of initial interpretation for syntactic categorical strategy is offered; its every layer represents the set of categorical deductions that can be executed. Within the framework of denotational semantics the set of dynamic goals is introduced for implementation of modifications for fixed values constructions. Functor interpretations which display the received syntactic and semantic categories are built. The semantic categorical strategy concept for obtaining an operator of fixed value determination with effective properties and immediate consequence operator as functor on a set of interpretations are introduced.

Ref.: 16 items.

ПРАВИЛА

оформления рукописей для авторов научно-технического журнала

"БИОНИКА ИНТЕЛЛЕКТА"

Статьи оформляются в редакторе Microsoft Word. Формат страницы – А4 (210х297 мм), поля: верхнее – 25мм, нижнее – 20 мм, левое, правое – 17 мм. Количество колонок – 2, интервал между ними – 5 мм, основной шрифт Times New Roman, кегль основного текста – 10 пунктов, междустрочный интервал – множитель (1,1), абзацный отступ – 6 мм. Объем рукописи – от 4 до 12 листов (языки: русский, украинский, английский).

Согласно решению редакционной коллегии, основанному на Постановлении Президиума ВАК Украины от 15.01.2003 №7-05/1 (Бюллетень ВАК, №1 2003, с.2) статья должна быть структурирована и содержать следующие необходимые элементы: **введение** (постановка проблемы в общем виде и ее связь с важными научными или практическими заданиями, актуальность); **суть** (постановка задачи, изложение основного материала исследования с обоснованием полученных научных результатов); **выводы** и перспективы дальнейших исследований в данном направлении.

Структура рукописи: УДК, инициалы и фамилии авторов, заголовок, основной текст, список литературы.

Заголовок статьи - кегль 12.

ОБРАЗЕЦ ОФОРМЛЕНИЯ

УДК 621.396.961.1

И.О. Фамилия, И.О. Фамилия

НАЗВАНИЕ РУКОПИСИ

Основной текст следует разделять на подразделы с заголовками, выделенными жирным шрифтом, пронумерованными арабскими цифрами, отступы: сверху – 6 пт, снизу – 3 пт.

1. Название раздела

Определения выделяются в тексте курсивом.

Рисунки и *таблицы* (черно-белые, контрастные) помещаются в текст после первой ссылки в виде встроенных объектов и раздельно нумеруются арабскими цифрами сквозной нумерацией, при наличии более одного объекта. Все надписи в рисунках и таблицах должны быть выполнены шрифтом Times New Roman, кегль в рисунках – 10, в таблицах – 9.

Рисунок содержит подрисуночную центрированную подпись (вне рисунка), шрифт 9, по центру, отступы сверху и снизу по 6 пт., как показано на рис. 1. Ширина рисунка должна соответствовать ширине колонки (или ширине страницы).

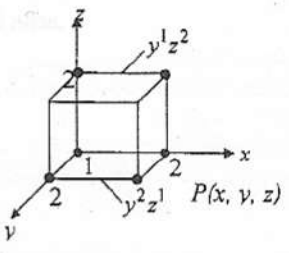


Рис. 1. График трехместного бинарного предиката

Табличный заголовок располагается справа над таблицей (вне таблицы, шрифт 9), что иллюстрируется табл. 1.

Таблица 1

x	3	2	1	0	2
y	0	1	2	3	2
$P(x,y)$	1	1	1	1	1

Формулы, символы, переменные, должны быть набраны в редакторе формул MathType (Equation). Формулы располагаются по центру и нумеруются при наличии ссылок на них в рукописи. Шрифт – Times New Roman. Высота переменной — 10 пунктов, индексов и подиндексов – 8 пт, основной мат. символ -12 (10) пт. Курсив в стиле формул включают только для текста и переменных. Сокращения русских слов выполняются прямым начертанием. Переменные, содержащиеся в тексте, также набираются в редакторе формул.

$$F(X_1,...,X_n)=\vee F(\delta_1,...,\delta_n)x_1^{\delta_1}..x_n^{\delta_n}. \qquad (1)$$

Список литературы включает опубликованные источники, на которые имеются ссылки в тексте, заключенные в квадратные скобки [1], печатается без абзацного отступа, кегль 9 пунктов, отступ сверху 6 пт.

Список литературы: 1. *Фамилия И.О.* Название книги. Город: Издательство, 1900. 000 с. 2. *Фамилия И.О.* Название статьи // Название журнала. Название серии. 1997. Т. 00, № 00. С. 00-00.

Поступила в редколлегию 00.00.2005

Рефераты представляются на трех языках: русский, украинский и английский (Times New Roman, кегль — 9 пунктов).

УДК 000.000.00

Название статьи /И.О.Фамилия, И.О.Фамилия // Бионика интеллекта: научн.-техн.журнал. 2005. № 00.С. 00-00.

Текст реферата.(3-4 предложения).

Табл.00. Ил.00. Библиогр.: 00 назв.

УДК 000.000.00

Назва статті /І.Б.Прізвище, І.Б.Прізвище // Біоніка інтелекту: наук.-техн. журнал. 2005. № 00. С. 00-00.

Текст реферату.

Табл. 00. Іл. 00. Бібліогр.: 00 назв.

UDC 000.000.00

Title of paper / Initials. Surname // Bionica intellecta. 2005. N 00. P. 000-000.

Text.

Tab. 00. Fig. 00. Ref.: 00 items.

Представление материалов: рукопись, рефераты, сведения об авторах (*фамилия, имя, отчество*, ученая степень, звание, должность, место работы, адрес, контактные телефоны) – на новой дискете 3,5 дюйма. Твердая копия материалов в одном экземпляре: рукопись, рефераты, сведения об авторах, рецензия, подписанная доктором наук, акт экспертизы, заявление на имя главного редактора со сведениями об авторах. Материалы, не соответствующие требованиям, редколлегией не рассматриваются.

Адрес редакции: Украина, 61166, Харьков, пр. Ленина, 14, ХНУРЭ, к. 437, тел. 702-13-97, факс 702-15-15, e-mail: innov@kture.kharkov.ua