

**ПЕРЕХОД ОТ ПРОДУКЦИЙ К ДВУДОЛЬНЫМ МИВАРНЫМ СЕТЯМ И ПРАКТИЧЕСКАЯ РЕАЛИЗАЦИЯ АВТОМАТИЧЕСКОГО КОНСТРУКТОРА АЛГОРИТМОВ, УПРАВЛЯЕМОГО ПОТОКОМ ВХОДНЫХ ДАННЫХ И ОБРАБАТЫВАЮЩЕГО БОЛЕЕ ТРЕХ МИЛЛИОНОВ ПРАВИЛ**

*Варламов О.О.*

*Московский автомобильно-дорожный государственный технический университет (МАДИ)*

*e-mail: ovar@narod.ru*

**Аннотация**

Показан теоретический переход от однодольных продукционных систем к двудольным миварным логико-вычислительным сетям. Приведены примеры реализации миварных сетей в формализмах матриц и графов. Теоретически обоснована линейная вычислительная сложность автоматического конструирования алгоритмов из переменных-объектов и правил-процедур миварных сетей. В качестве миварных правил могут быть использованы различные сервисы, модули и вычислительные процедуры. Автоматический конструктор алгоритмов может использоваться для поиска логического вывода в области создания экспертных систем.

На основе миварных сетей создан программный комплекс УДАВ, который обрабатывает более 1,17 млн переменных и более 3,5 млн правил на обычных компьютерах и ноутбуках. Приведены результаты практических расчетов и решений различных прикладных задач, которые на практике подтверждают линейную вычислительную сложность конструирования алгоритмов в формализме миварных сетей. Программный комплекс УДАВ используется как для решения логических, так и вычислительных задач. Приведены сведения о практической реализации нескольких миварных экспертных систем.

Миварные сети позволяют перейти к новому поколению экспертных систем и интеллектуальных пакетов прикладных программ. Миварный подход позволил на практике создать автоматические обучаемые эволюционные активные логически рассуждающие информационные системы. В перспективе на основе миварных сетей будет создана глобальная мультипредметная активная экспертная система под названием "Миварная активная энциклопедия".

**Ключевые слова:** мивар, миварные сети, логический вывод, вычислительная сложность, искусственный интеллект, интеллектуальные системы, экспертные системы.

**Введение**

Проблема создания интеллектуальных систем остается актуальной и практически значимой. Создание экспертных систем нового поколения позволит автоматизировать решение различных сложных интеллектуальных задач и повысит конкурентоспособность своих пользователей. Миварный подход позволил предложить новые модели и методы обработки информации и управления [1-22]. Миварные технологии накопления и обработки информации разрабатываются в России достаточно давно. Первые статьи были посвящены исследованию некоторых задач теории графов и разработке линейного матричного метода определения маршрута логического вывода на адаптивной сети правил [1-3]. Затем были

работы по созданию миварного информационного пространства и эволюционных баз данных и правил [4-5]. Наиболее строгое формализованное и теоретическое оформление мивары получили в работах [6-7]. Затем были рассмотрены вопросы развития миваров [8-10] и их применения для создания различных тренажеров и обучающих систем [11-22]. Наиболее полно обзор теории и последних достижений миваров приведен в работах [4, 6, 10, 15, 18].

Будем понимать под системами искусственного интеллекта активные самообучающиеся логически рассуждающие системы. В прошлом веке были разработаны технологии создания экспертных систем по отдельным узконаправленным предметным областям. Это было обусловлено сложностями формализованного описания требуемых предметных областей и тем, что системы логического вывода не могли обрабатывать более 20 объектов/правил. В то же время, получили развитие "интеллектуальные пакеты прикладных программ" (ИППП), которые позволяли решать в автоматизированном режиме задачи в разных областях, где требовались вычисления и конструирование алгоритмов решения задач. Технологии ИППП развиваются в миварах и сервисно-ориентированных архитектурах.

Миварный подход объединяет и развивает достижения в научных областях: баз данных, вычислительных задач, логической обработки и включает две основные технологии.

1) *Миварная технология накопления информации* - это способ создания глобальных эволюционных баз данных и правил (знаний) с изменяемой структурой на основе адаптивного дискретного миварного информационного пространства унифицированного представления данных и правил, базирующегося на трех основных понятиях "вещь, свойство, отношение".

2) *Миварная технология обработки информации* - это способ создания системы логического вывода или "автоматического конструирования алгоритмов из модулей, сервисов или процедур" на основе активной обучаемой миварной сети правил с линейной вычислительной сложностью.

Миварная технология накопления информации предназначена для хранения любой информации с возможным эволюционным изменением структуры и без ограничений по объему и формам представления.

Миварная технология обработки информации предназначена для обработки информации, включая логический вывод, вычислительные процедуры и "сервисы".

Фактически, миварные сети позволяют развить продукционный подход и создать автоматическую обучаемую логически рассуждающую систему. В наших работах показано, что миварный подход объединяет и развивает продукционные системы, онтологии, семантические сети, сервисно-ориентированные архитектуры, многоагентные системы и другие современные информационные технологии в целях создания интеллектуальных систем и систем ИИ.

В настоящее время "движок" УДАВ выполняет поиск логического вывода и автоматически конструирует алгоритмы решения задач из готовых модулей-сервисов, управляемые потоком входных данных. На обычном ноутбуке УДАВ обрабатывает более 1,17 млн переменных и 3,5 млн правил. Программная реализация наглядно доказывает на практике линейную сложность поиска логического вывода, эволюционность и активность работы миварных экспертных систем нового поколения.

## **Анализ существующих парадигм и моделей обработки данных**

Традиционно выделяют следующие парадигмы и модели обработки данных: исчисление высказываний, исчисление предикатов, продукции, семантические сети, онтологии и др. У продукционного подхода есть важные преимущества. Поспелов Д.А. писал, что знания о внешнем мире могут иметь двоякую природу:

1) могут содержать декларативное описание фактов и явлений внешнего мира, фиксирующее их наличие или отсутствие, а также основные связи и закономерности, в которые эти факты и явления входят;

2) могут содержать и процедурные описания того, как надо манипулировать с этими фактами и достигать целей, интересных для системы [23].

Продукции в общем виде записывают в форме "Если..., то...". Часть специалистов по интеллектуальным системам считает, что запись знаний в виде систем продукций носит универсальный характер – любые знания можно записать в такой форме. В системе продукций можно представлять самые разнообразные правила, процедуры, формулы или сервисы. К ним, по сути, сводятся все каузальные, т.е. причинно-следственные утверждения. Поспелов Д.А. делает совершенно обоснованный вывод: "Продукционные системы получили при представлении знаний в последнее время наибольшее распространение". Следовательно, применение продукционного подхода для логико-вычислительной обработки разнообразных данных является вполне обоснованным и целесообразным.

Для решения многих практических задач применения информационных систем и процессов требуется проводить как логическую, так и вычислительную обработку данных. Исторически так сложилось, что области логического вывода и вычислительной обработки развивались самостоятельно и успешно решали различные классы задач. В некотором смысле, даже существовало противоречие между этими подходами [2, 3, 4, 10-17, 23-32]. Кроме того, разделяли проблемы обработки и хранения различных данных.

Базы данных преимущественно использовались только для хранения и поиска требуемых данных, а системы логического вывода и вычислений применялись для обработки информации, поиска решений и т.п. Получалось, что эти области относительно слабо пересекались, хотя, в плане перспектив развития в каждой из них регулярно провозглашались цели объединения всех функций по накоплению и обработке информации в одной системе [4, 10, 23-32].

Если проводить аналогию с человеком, то наш разум одновременно накапливает и хранит данные, комплексно решает и логические, и вычислительные, и логико-вычислительные задачи. Миварный подход позволяет в едином формализме проводить и эволюционное накопление данных в миварном информационном пространстве, и выполнять совмещенную логико-вычислительную обработку в миварных логических сетях.

Миварный метод логико-вычислительной обработки позволяет решать большой класс сложных научных и практических задач. Прежде всего, проведем анализ существовавших ранее подходов к решению различных классов задач и оценим их ограничения. Затем перейдем к анализу проблем, достижений и перспектив в области баз данных и миварном информационном пространстве унифицированного представления данных и правил.

### Возможности и ограничения продукционного подхода

Для анализа проблем метода логико-вычислительной обработки данных и его возможных применений очень важным является следующее заключение Пospelова Д.А.: "Мы хотим отметить, что ядром всех основных типов рассмотренных интеллектуальных систем являются база знаний и блок, осуществляющий вывод с помощью знаний (решатель, планировщик или логический блок). Этот вывод составляет основную процедуру, реализуемую в интеллектуальных системах" [23, стр. 129].

В настоящее время продолжается дискуссия о роли и возможных применениях различных логических механизмов и, прежде всего, исчисления высказываний и исчисления предикатов и продукций. У продукционного подхода есть важные преимущества. Приведем мнение Пospelова Д.А.: "Для описания знаний в интеллектуальных системах используются специальные языки описания знаний (ЯОЗ). ... Простейшими видами таких ЯОЗ являются языки исчисления высказываний или исчисления предикатов вместе с теми процедурами вывода, которые для них известны. Однако в современных интеллектуальных системах такие языки используются довольно редко. Куда более распространены в них языки, основанные на *продукциях*. ... Продукции в общем виде можно записать в форме "если..., то...", но к продукциям относятся не только выражения, имеющие эту форму, но и многие другие" [23, стр. 129]. У Пospelова Д.А. приведены 9 типов продукций и специально подчеркнута, что возможны продукции и других типов [23, стр. 131-134]. Далее делается вывод: "... продукции могут иметь весьма различное значение. В качестве их левых и правых частей могут выступать и некоторые утверждения, и действия" [23, стр. 134]. Также "...приводят немало примеров, когда знания, внешне не имеющие продукционной формы, удастся перевести в систему продукций" [23, стр. 129-130]. Далее там же приведен пример перевода в продукционную форму записи химических реакций, для чего используются различные виды продукций. "К ним (продукциям), по сути, сводятся все каузальные, т.е. причинно-следственные утверждения..." [23, стр. 130]. Отметим, что Кузнецов О.П. в [24, стр. 282-283] под продукциями понимает множество правил вывода в канонических системах (системах продукций Поста), в которых есть посылки и следствия. С точки зрения анализа метода логико-вычислительной обработки данных принципиально важным является то, что в системе продукций можно представлять самые разнообразные правила, процедуры, формулы или сервисы. Следовательно, применение продукционного подхода для логико-вычислительной обработки разнообразных данных является целесообразным.

Также Пospelовым Д.А. дается определение: "*Продукционной системой* будем называть любую совокупность продукций, в которую могут входить продукции любого из перечисленных выше типов" [23, стр. 134]. Существуют различные конструкции продукций. В наиболее общем виде "вместо продукций типа  $a \Rightarrow b$  рассматривают более сложные конструкции. В общей форме продукции имеют вид:

$$i, P, A \Rightarrow B, Q.$$

Здесь  $A \Rightarrow B$ - обычная продукция "если ..., то ...", которая носит название *ядра продукции*. Элемент  $P$  характеризует внешние условия или *условия применимости* продукции, определяемые факторами, не входящими непосредственно в  $A$ , например, целями, которые стоят перед рассуждающей системой. Условия  $P$  позволяют из всех продукций, у которых в левой части ядра стоит  $A$ , отбирать нужную часть продукций. Элемент  $Q$  характеризует *сферу проблемной области* базы знаний, или *предусловия применимости* продукции. Эти

предусловия ничем не отличаются от  $P$ , но выделяют подсистемы продукций на ранг выше тех, которые выделяют условия. Предусловия задают формальную систему, в рамках которой будут проводиться логические рассуждения... Наконец,  $Q$  характеризует постусловия продукции, указывающие на те изменения, которые необходимо внести в базу знаний и в систему продукций после реализации данной продукции" [23, стр. 134-135].

В общем виде продукции встречаются весьма редко. Отметим, что существует аналогичный подход, основанный на гиперправилах с мультиактивизаторами [4]. Хорошевский В.Ф. в [31, стр. 82-83] при описании "слоеного пирога" SemanticWeb выделяет промежуточный "слой правил", для которого ведутся исследования различных систем вывода на правилах.

Под логической обработкой принято понимать некий вывод, лежащий в основе человеческих рассуждений. Для проведения анализа используем описания и исходную информацию из широко известных источников. В работе Поспелова Д.А. выдвинуто следующее основополагающее положение: "Всякий вывод, как бы он не был организован, носит переборный характер. ... программа вынуждена перебирать варианты, заходить в тупики, проходить циклы прежде, чем она сможет найти правильный путь решения. Повышение эффективности процесса вывода - центральная проблема всех автоматизированных систем дедуктивного вывода" [23, стр. 79].

Особый интерес представляет описание общей схемы выводов, лежащей "в основе большого количества моделей человеческих достоверных рассуждений" [23, стр. 83]. Сначала Поспелов Д.А. приводит такое описание в виде некоторого дерева вывода: "Вершинам этого дерева соответствуют определенные утверждения  $F_i$ , а дуги определяют порядок получения новых утверждений. Те дуги, которые сходятся в зачерненные точки, образуют *конъюнктивные условия вывода*, а те дуги, которые между собой соединены "дужкой", образуют *дизъюнктивные условия вывода*. ... Дерево вывода с такими условиями переходов от вершины к вершине носит название *И-ИЛИ дерева*. В И-ИЛИ дереве ориентация дуг показывает направление вывода. Естественное разбиение вершин дерева по ярусам отражает глубину вывода (число шагов, необходимых для получения утверждений данного яруса). Первый ярус дерева образуют вершины..., играющие роль аксиом или утверждений, истинность которых задается извне" [23, стр. 83-84]. Однако далее Поспелов Д.А. пишет: "Схема вывода не обязательно описывается в виде дерева. Она может иметь вид произвольной сети, ориентированной, неориентированной или частично ориентированной" [23, стр. 84]. На рисунке 1 показан пример неориентированной сети, аналогичный рисунку в [23, стр. 83]. "Такая сеть (наличие или отсутствие ориентации не играет здесь роли) называется *И-ИЛИ сетью*. Процесс вывода на И-ИЛИ сети протекает следующим образом. Пусть мы хотим доказать утверждение  $F_6$  (на рисунке 1 этому соответствует *целевая вершина* (выделена двойным контуром)). В качестве априорно доказанного задано утверждение  $F_1$  (ему соответствует начальная вершина, которая на рисунке 1 заштрихована). Как из  $F_1$  можно получить  $F_6$ ? Если считать, что все связи допускают ориентацию в нужную сторону, то из  $F_1$  можно получить  $F_3$ , затем  $F_5$  и, наконец,  $F_6$ . Но этот путь нам удалось отыскать потому, что сеть, показанную на рисунке 9, мы видим "с высоты птичьего полета". Лабиринт поиска лежит в виде чертежа перед нами. Именно это позволяет нам не делать лишних попыток, не двигаться в ненужную сторону, а идти кратчайшим путем к цели" [23, стр. 84]. Это выражало общее мнение ученых. Отметим, что

на рисунке 1 фактически изображен однодольный граф, т.е. все вершины графа принадлежат одному классу [24, стр. 125]. Этот факт будет очень важен для дальнейшего анализа.

Поспелов Д.А. описал два метода логического вывода. "Эта процедура известна среди специалистов под названием *метода прямой волны*. Волна поиска путей к целевой площадке распространяется от всех площадок, играющих роль начальных. ... Возможен и другой способ поиска доказательства. Он носит название *метода обратной волны*. В этом методе волна начинает свое движение от целевых площадок и движется в направлении начальных площадок лабиринта. ... Различие между прямой и обратной волной состоит в том, что они порождают в процессе своего движения различные промежуточные "фронты" площадок, что приводит к различному числу шагов при поиске. Часто используется смешанный метод вывода, при котором одновременно движутся прямая и обратная волны. При встрече этих волн формируется путь вывода от начальных аксиом к целевым выражениям" [23, стр. 85].

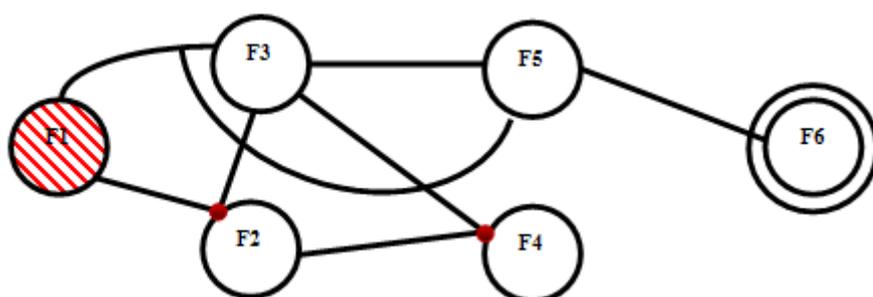


Рисунок 1. Пример однодольного графа неориентированной "И-ИЛИ сети".

Отметим, что эти работы не нашли широкого практического применения в прошлом веке. Это обусловлено тем, что не было предложено метода быстрой логической обработки данных. Как было показано выше, ученые исходили из предпосылки, что любой вывод носит переборный характер, что означает факториальный рост вычислительной сложности при увеличении количества правил. Даже современные суперкомпьютеры при таком подходе позволяют обрабатывать в реальном масштабе времени лишь 20 правил. Фактически, это позволяет решать только "игрушечные" модельные задачи, т.к. современные требования начинаются с сотен и достигают десятков тысяч переменных и правил даже для относительно простых предметных областей. Миварный подход и создание логико-вычислительных сетей позволил эффективно реализовать предложенные для продукций Поспеловым Д.А. *методы прямой и обратной волны* [23, стр. 85].

### Системы искусственного интеллекта и мивары

Рассмотрим основные подходы к созданию искусственного интеллекта (ИИ) и экспертных систем. Приведем мнение Дж. Люгера, который считал "... объединение разрозненных областей искусственного интеллекта с помощью детального описания его теоретических основ ... Интеллект - это сложная область знаний, которую невозможно описать с помощью какой-то одной теории. Ученые строят целую иерархию теорий, характеризующие его на разных уровнях абстракции" [32, стр. 20]. Формальный подход Рассела и Уайтхеда к математическим умозаключениям сделал возможной его автоматизацию в реальных вычислительных машинах. Логический синтаксис и формальные правила вывода, разработанные Расселом и Уайтхедом, лежат в основе систем

автоматического доказательства теорем и составляют теоретические основы искусственного интеллекта [32, стр. 34-35]. Однако, с другой стороны, это наложило и важные ограничения на возможности описания реальных предметных областей: формализм, как выясняется, тоже имеет свои недостатки вместе с преимуществами. Нельзя не отметить и роль А. Тарского, чья теория ссылок сыграла принципиальную роль в процессе формирования ИИ. Согласно этой теории, правильно построенные формулы Фреге или Рассела-Уайтхеда определенным образом ссылаются на объекты реального мира, а эта концепция лежит в основе большинства теорий формальной семантики. Люгер справедливо отмечает, что ИИ не стал жизнеспособной научной дисциплиной до появления цифровых вычислительных машин.

Однако архитектура цифровых компьютеров наталкивает на специфичное представление теории ИИ. Получается, что интеллект - это способ обработки информации. Далее Люгер формулирует замечательную мысль, к которой мы с удовольствием присоединяемся: "Мы часто забываем, что инструменты, которые мы создаем для своих целей, влияют своим устройством и ограничениями на формирование наших представлений о мире". Такое, казалось бы, стесняющее наш кругозор взаимодействие является важным аспектом развития человеческого знания: инструмент (а научные теории, в конечном счете, тоже инструменты) создается для решения конкретной проблемы. По мере применения и совершенствования инструмент подсказывает другие способы его использования, которые приводят к новым вопросам и, в конце концов, разработке новых инструментов [32, стр. 35]. Если говорить о миварах и перечисленных выше формализмах Фреге, Рассела и Уайтхеда, Тарского и многих других, то надо помнить, что наука развивается по спирали и старые формализмы требуют своего продолжения на новом витке. Упомянутые выше исчисление предикатов и другие ранние фундаментальные формализмы при необходимости могут быть реализованы и в правилах (отношениях) миварных сетей. Только надо помнить, что есть и другие формализмы представления информации для ИИ, а у исчисления предикатов первого порядка есть достаточно жесткие ограничения и не очень большие возможности, если вспоминать не об "игрушечных" задачах, а говорить о создании глобальных познающе-диагностических систем, решающих сложные логико-вычислительные задачи в реальном времени. Как видно из [23-32], в современной теории ИИ кроме предикатного подхода уже разработано большое количество других подходов, кардинально отличающихся от исчисления предикатов и т.п. Подчеркнем, что продукционный подход и его развитие в миварных сетях являются еще одной альтернативой исчислению предикатов при создании ИИ.

В области ИИ существует две фундаментальные проблемы - это представление знаний и поиск [32, стр. 42]. Первая проблема относится к получению новых знаний с помощью формального языка. Поиск - это метод решения проблемы, в котором систематически просматривается пространство состояний задачи, т.е. альтернативных стадий ее решения. Автоматическое доказательство теорем - одна из старейших областей возможного применения ИИ, где было много достижений, исследований и программ, включая Универсальный решатель задач Ньюэлла и Саймона. Дж. Люгер подчеркивает, что именно "...эта ветвь принесла наиболее богатые плоды..." [32, стр. 44]. Благодаря исследованиям в этой области были формализованы алгоритмы поиска и разработаны языки формальных представлений, такие как исчисление предикатов и логический язык программирования Prolog.

Приведем обоснование Дж. Люгера: "... привлекательность автоматического доказательства теорем основана на строгости и общности логики. В формальной системе логика располагает к автоматизации. Разнообразные проблемы можно попытаться решить, представив описание задачи и существенно относящуюся к ней информацию в виде логических аксиом и рассматривая различные случаи задачи как теоремы, которые нужно доказать. Этот принцип лежит в основе автоматического доказательства теорем и систем математических обоснований" [32, стр. 44]. Далее следует вывод в этой наиболее богатой ветви: "К сожалению, в ранних пробах написать программу для автоматического доказательства, не удалось разработать систему, которая бы единообразно решала сложные задачи" [32, стр. 44]. Таким образом, Дж. Люгер подтверждает наш тезис о том, что в прошлом веке даже в самых передовых областях ИИ ученые не смогли решить сложные задачи, а значит, нужны принципиально новые подходы и исследования, к числу которых относится и миварный подход. Приведем обоснование ограниченности возможностей автоматического доказательства теорем. "Это было обусловлено способностью любой относительно сложной логической системы сгенерировать бесконечное количество доказуемых теорем: без мощных методик (эвристик), которые бы направляли поиск, программы доказывали большие количества не относящихся к делу теорем, пока не наткнулись на нужную. Из-за этой неэффективности многие утверждают, что чисто формальные синтаксические методы управления поиском в принципе не способны справиться с такими большими пространствами, и единственная альтернатива этому - положиться на неформальные, специально подобранные к случаю (лат "ad hoc") стратегии, как это, похоже, делают люди. Это один из подходов, лежащих в основе экспертных систем... и он оказался достаточно плодотворным" [32, стр. 44].

Таким образом, возникают новые проблемы: работа с бесконечными множествами теорем и разработка эвристик, про которые ранее было показано, что они не гарантируют решение задачи. Конечно, за прошедшее время ученым удалось разработать мощные эвристики, основанные на оценке синтаксической формы логического выражения, которые в результате понижают сложность пространства поиска. Кроме того, пришло понимание, что системе не обязательно решать особо сложные проблемы без человеческого вмешательства. "Многие современные программы доказательств работают как умные помощники, предоставляя людям разбивать задачи на подзадачи и продумывать эвристики для перебора в пространстве возможных обоснований" [32, стр. 44]. Этот вывод может служить обоснованием нашего утверждения, что ИИ - это усилитель человеческих способностей и автоматизация мыслительных процессов. Кроме того, это косвенно подтверждает необходимость введения шкалы измерений интеллектуальности автоматических систем и наличие относительно слабых форм интеллекта у уже существующих компьютерных программ и устройств. Значит, в таком смысле ИИ уже существует и продолжает развиваться, помогая человеку решать сложные задачи.

Вместе с тем, рассмотренные проблемы доказательств поднимают вопросы необходимости формализации представлений и описаний различных предметных областей. Однако такая возможность есть далеко не всегда. Значит, надо исследовать возможности обработки менее формализованной информации или альтернативных моделей представления данных и знаний, что также развивается в миварном подходе.

Миварный подход позволяет работать с различными формами представления данных и правил (знаний), включая и работу с бесконечными описаниями сущностей - вещей, отношений и свойств в миварном многомерном динамическом информационном пространстве унифицированного представления данных и правил (знаний).

Если учитывать, что в области ИИ существует две фундаментальные проблемы - представление знаний и поиск, то миварные сети позволяют по новому представлять информацию, что, в свою очередь, позволило кардинально ускорить поиск решений и перейти от NP-сложности логического вывода к линейной вычислительной сложности миварного метода обработки данных и правил.

### **Представление сетей продукций в виде двудольных графов**

Перейдем к формализованному представлению продукций и сетей, которые могут быть сформированы из них. Выше было отмечено, что Пospelов Д.А. [23, стр. 83-84] представляет сеть правил в таком виде, что она отображается в виде однодольного графа. В работах [1-22] предложен подход по взаимосвязи сетей правил и графов. Более того, показано, что некоторые задачи логического вывода можно решать на основе подходов теории графов. Сети правил и процедур целесообразнее представлять в виде двудольных графов, получая нечто аналогичное сетям Петри, но с соответствующим развитием до миварных логических сетей [1-22]. Итак, напомним определения двудольных графов, которые необходимы для продолжения анализа. "Граф  $G=(V, E)$  называется двудольным, если существует разбиение  $V=\{V_1, V_2\}$  такое, что никакие две вершины из  $V_1$  или из  $V_2$  не являются смежными" [25, стр. 223]. "Двудольным графом  $G=(X, Y, E)$  называется неориентированный граф, вершины которого можно разбить на два класса  $X$  и  $Y$  так, что концы каждого ребра принадлежат разным классам" [24, стр. 125]. Приведем следующее важное замечание: "введенные понятия допускают естественное обобщение. Неориентированный граф называется  $k$ -дольным, если его вершины можно разбить на  $k$  классов так, что концы каждого ребра принадлежат разным классам" [24, стр. 125]. Таким образом, можно использовать и двудольные, и трехдольные и многодольные ( $k$ -дольные) графы для разных предметных областей. Развивается новый подход к единому представлению знаний и данных, который получил название "миварный". Миварный подход является обобщением и развитием продукционного подхода, сетей Петри и других формализмов, применяемых для логической обработки данных.

### **Миварные сети**

Миварные сети могут быть представлены в виде двудольного графа, состоящего из объектов-переменных и правил-процедур. Для этого, прежде всего, составляются два списка, которые и образуют две непересекающиеся доли графа: список объектов и список правил. Объекты изображены овалами (кружочками) на рисунке 2. Каждое правило в миварной сети является развитием продукций, гиперправил с мультиактивизаторами или вычислительных процедур. Доказано, что с точки зрения дальнейшей их обработки все эти формализмы идентичны и представляют собой, по сути, вершины двудольного графа, которые изображены прямоугольниками на рисунке 2. На рисунке 3 показан пример записи "объектов" и "правил" миварной сети в формате XML, а алгоритм логико-вычислительной обработки для программного комплекса УДАВ показан на рисунке 4 [10-22].

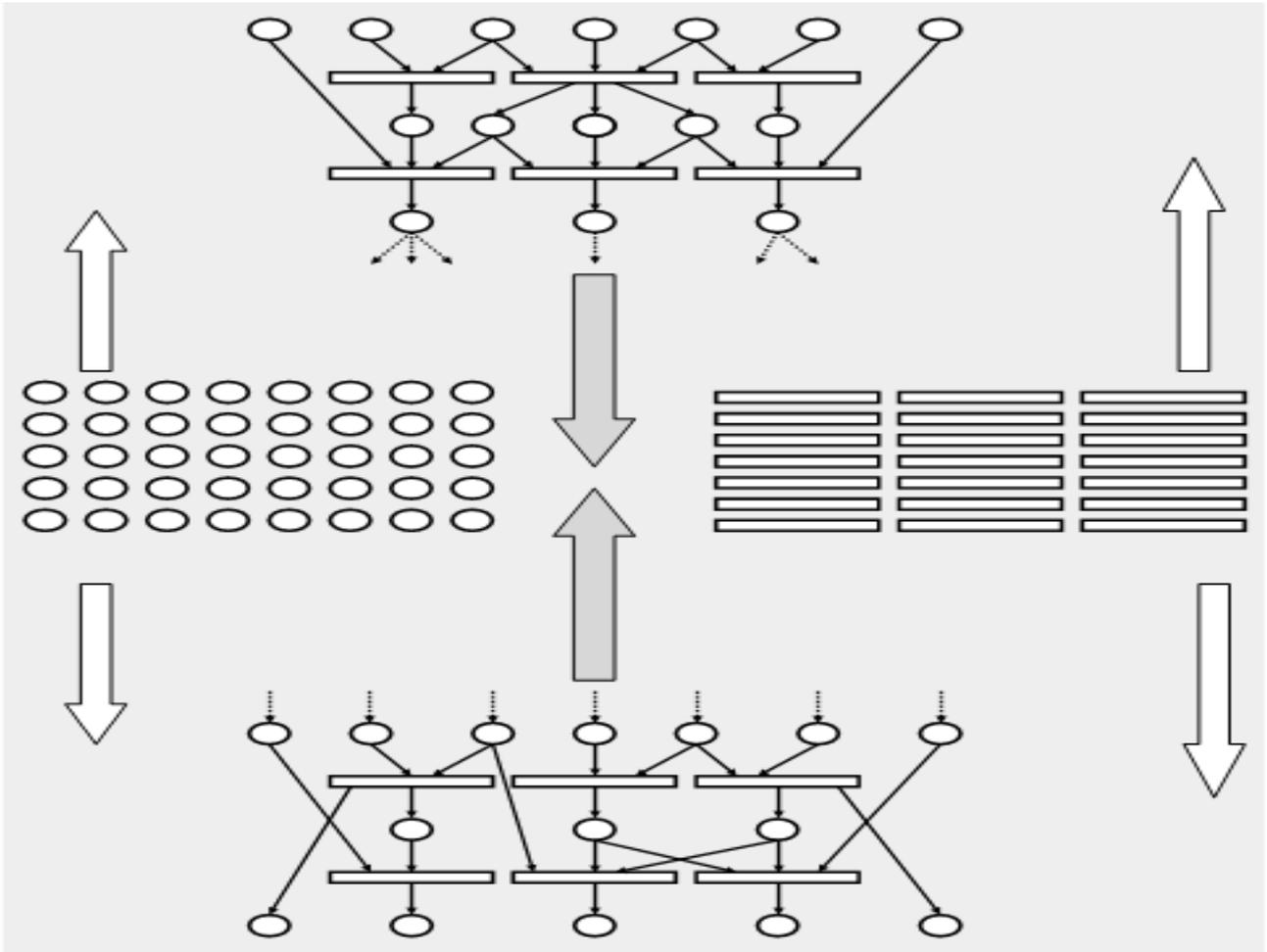


Рисунок 2. Представление миварной сети в виде двудольного графа.

```

<?xml version="1.0" encoding="UTF-8" ?>
-<root>
-<parametr>
<parametr id="P1" value="0.0" description="Угол А, против. стороне а (град)" />
<parametr id="P2" value="0.0" description="Угол В, против. стороне b (град)" />
<parametr id="P3" value="0.0" description="Угол С, против. стороне с (град)" />
...
</parametr>
-<rules>
<rule id="R1" resultId="P1" initId="P2,P3" value="180-P2-P3" description="Сумма" />
<rule id="R2" resultId="P2" initId="P1,P3" value="180-P1-P3" description="Сумма" />
<rule id="R3" resultId="P3" initId="P1,P2" value="180-P1-P2" description="Сумма" />
...
</rules>
-<metadata>
<idParametr inc="33" />
<idRule inc="161" />
</metadata>
</root>

```

Рисунок 3. Пример записи объектов и правил в формате XML.

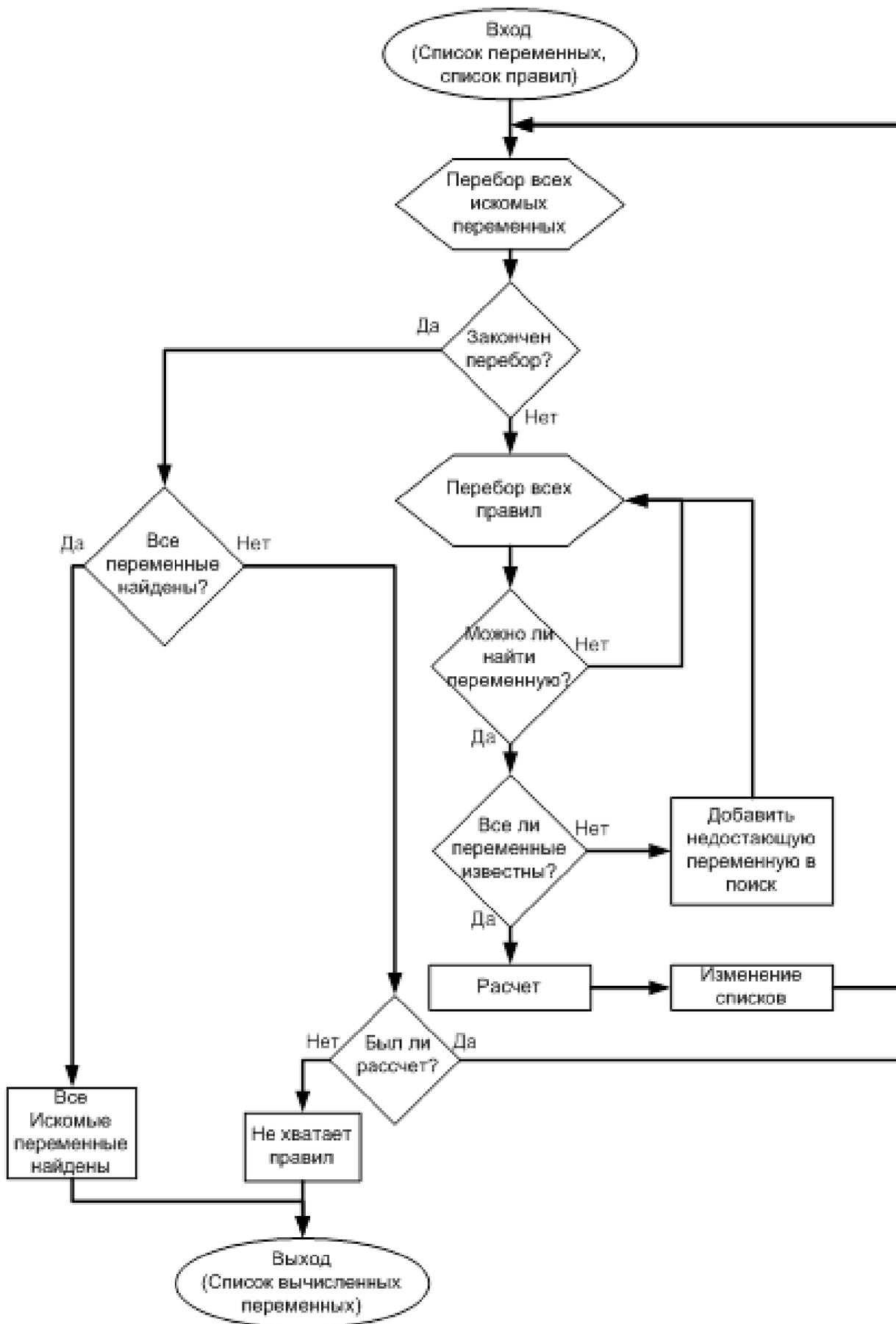


Рисунок 4. Алгоритм миварной логико-вычислительной обработки для ПК УДАВ.

## Миварный метод логико-вычислительной обработки данных

Для реализации метода логико-вычислительной обработки данных должно быть выполнено следующее. Прежде всего, должно быть разработано формализованное описание предметной области. Для этого на основе миварного подхода выделяют основные объекты-переменные и правила-процедуры, а затем создают соответствующие списки "объектов" и "правил". Формализованное представление метода логико-вычислительной обработки данных аналогично двудольному графу миварной логической сети, показанному на рисунке 2. Теоретические основы миварного метода логико-вычислительной обработки информации (миварного метода) были впервые опубликованы в [2-4] еще в 2002 году. В статье [2] представлены теоретические основы создания линейного матричного метода определения маршрута логического вывода на адаптивной сети правил.

Выделим три основных этапа миварной обработки информации:

- 1) формирование миварной матрицы описания предметной области;
- 2) работа с матрицей и конструирование алгоритма решения заданной задачи;
- 3) по полученному алгоритму выполнение всех вычислений и нахождение ответа.

Первый этап по существу этапом синтеза концептуальной модели предметной области и ее формализации в виде продукционных правил с переходом на миварные правила:

*"входные объекты - правила/процедуры - выходные объекты".*

В настоящее время именно этот этап является наиболее сложным и требует участия человека-специалиста (эксперта) для создания миварной модели предметной области.

На втором этапе собственно и выполняется автоматическое конструирование алгоритма решения или логический вывод. Исходными данными для этого являются - миварная матрица описания предметной области, а также заданные входные ("ДАНО") и искомые ("НАЙТИ") объекты-переменные. На третьем этапе должно выполняться решение по полученному алгоритму. Однако, в настоящее время в программном комплексе УДАВ совмещены работы по второму и третьему этапам. Отметим, что уже создано более 7 различных реализаций миварного метода. Часть этих реализаций в виде программных комплексов выделяет все три основных этапа и обрабатывает их отдельно. В данной работе основное внимание уделим программному комплексу УДАВ, в котором все три этапа могут быть совмещены.

Кратко опишем теоретические основы работы миварного метода логико-вычислительной обработки данных [2-4, 10, 15, 20]. Напомним, что практические реализации в виде конкретных алгоритмов и программных комплексов могут отличаться от приведенного описания, которое является наиболее общим и носит теоретический характер.

Прежде всего, для миварной сети логических правил, представленной в списочной форме, строится матрица. Затем, на основе анализа этой матрицы определяется факт наличия успешного маршрута вывода, потом определяются возможные маршруты логического вывода, а на последнем этапе из этих маршрутов выбирают "кратчайший", наиболее оптимальный по заданным критериям оптимальности.

Пусть известны  $m$  правил и  $n$  переменных (входящих в правила либо в качестве исходных, активизирующих их, либо в качестве получаемых, т.е. выходных). Тогда в матрице  $V (m \times n)$ , каждая строка которой соответствует одному из правил и содержит информацию об используемых в правиле переменных, могут быть представлены все взаимосвязи между правилами и переменными. При этом в каждой строке все входные

переменные этого правила на соответствующих позициях матрицы помечаются символом  $x$ , все выходные –  $y$ ; все переменные, которые уже получили в процессе вывода или задания исходных данных некоторое конкретное значение, –  $z$ ; а все искомые (выходные) переменные, т.е. те, которые необходимо "вывести" из исходных (входных) данных, –  $w$ . Кроме того, добавим в матрицу  $V$  одну строку и один столбец для хранения в них служебной информации. Напомним, что количество служебной информации может видоизменяться для различных конкретных реализаций программных комплексов. Получаем матрицу  $V$  размерности  $(m+1) \times (n+1)$ , в которой отражена вся структура исходной сети правил. Сразу отметим, что структура этой логической сети может изменяться в любое время, т.е. это сеть правил с изменяемой (эволюционной) структурой. Пример такой матрицы показан на рисунке 5. На рисунке 6 эта же матрица показана в виде исходного задания двудольного графа миварной логической сети.

V	1	2	3	4	5	...	n-2	n-1	n	n+1
1	x	x	x					y	y	
2			x	y	y			x	x	
...						...				
m		x		x	x		y			
m+1										

Рисунок 5. Пример исходного задания матрицы  $V$  размерности  $(m+1) \times (n+1)$  со структурой сети правил.

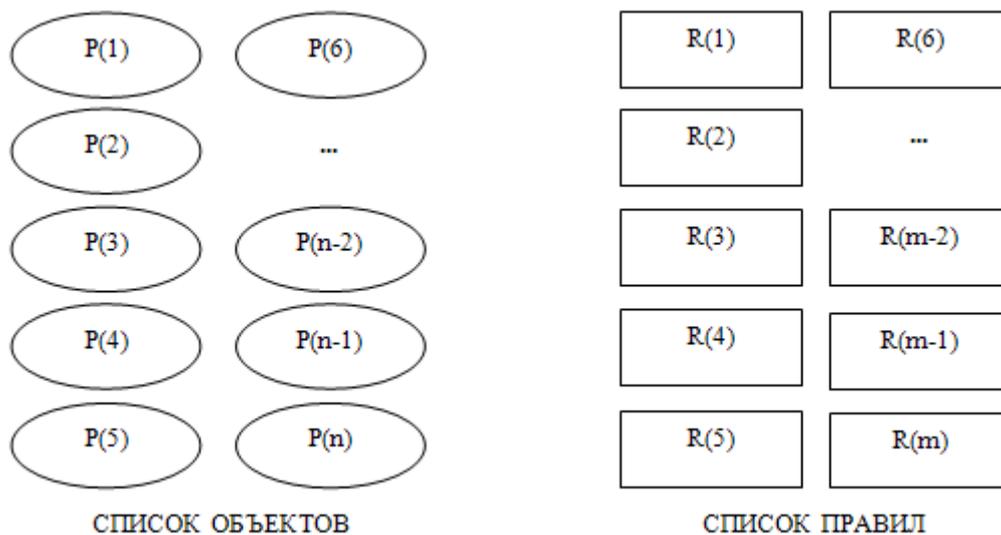


Рисунок 6. Пример исходного задания двудольного графа миварной логической сети.

Опишем пример работы метода. Для поиска маршрута логического вывода на полученной матрице производят действия по следующим (пронумерованным) этапам.

1. В строке  $(m+1)$  известные переменные помечают  $z$ , а искомые –  $w$ . Например, на рисунке 7 в строке  $(m+1)$  символом  $z$  отмечены позиции: 1, 2, 3, а символом  $w$  – позиция  $(n-2)$ . На рисунке 8 это действие показано на примере построения двудольного графа миварной сети.

2. Осуществляют последовательно, например, сверху вниз, поиск таких правил, которые могут быть активизированы, т.е. у которых известны все входные переменные. Если таких правил нет, то маршрута логического вывода нет и необходимо выдать запрос на уточнение (добавление) входных данных. Если такие правила, которые могут быть активизированы, есть, то у каждого из них в соответствующем месте служебной строки делается пометка, что правило может быть запущено. Например, можно ставить в этих клетках таблицы цифру 1, что и показано на рисунке 9 в клетке  $(1, n+1)$ . На рисунке 10 это действие показано на примере построения двудольного графа миварной логической сети.

3. Если таких правил несколько, то осуществляется выбор по заранее определенным критериям такого или таких правил, которые должны быть активизированы в первую очередь. При наличии достаточных ресурсов, одновременно могут запускаться сразу несколько правил.

4. Имитация запуска правила (процедуры) осуществляется путем присваивания выводимым в этом правиле переменным значений "известно", т.е., в нашем примере, - z. Запущенное правило, для удобства дальнейшей работы помечается дополнительно, например, цифрой 2 (это не обязательно). Например, как показано на рисунке 11: в клетках  $(m+1, n-1)$ ,  $(m+1, n)$  и  $(1, n+1)$  произведены соответствующие изменения. На рисунке 12 это действие показано на примере построения двудольного графа миварной логической сети.

5. После имитации запуска правил проводят анализ достижения цели, т.е. анализируют получение требуемых значений путем сравнения служебных символов в служебной строке. Если в служебной строке  $(m+1)$  осталось хоть одно значение "искомая" (т.е. w), то осуществляют дальнейший поиск маршрута логического вывода. В противном случае, задача считается успешно решенной, а все задействованные правила в соответствующем порядке их запуска и образуют искомый маршрут логического вывода. На рисунках 11 и 12 запущено только одно правило, но искомые значения не определены, значит надо переходить к следующему этапу.

6. Прежде всего, определяют наличие таких правил, которые могут быть запущены после определения новых значений на предыдущем этапе. Если таких правил нет, то маршрута вывода нет и поступают аналогично пункту 2 этого метода. Если такие правила есть, то продолжают поиск маршрута вывода. В нашем примере такие правила есть, что и показано на рисунке 13. В клетке  $(2, n+1)$  появилась 1 как признак возможности запуска этого правила. На рисунке 14 это действие показано на примере построения двудольного графа миварной логической сети.

7. На следующем этапе опять аналогично этапу 4 запускают правила (имитируют запуск), далее аналогично этапам 5 и 6 выполняют необходимые действия столько раз, сколько требуется для получения результата. При необходимости, повторяют все этапы со 2 по 7 до достижения результата. При этом результат может быть как положительный - маршрут вывода существует, так и отрицательный - вывода нет из-за неопределенности входных данных. Для наглядности продолжим пошаговое выполнение нашего примера. Итак, в нашем примере необходимо провести имитацию запуска правила 2, что и показано на рисунке 15. На рисунке 16 это действие показано на примере построения двудольного графа миварной логической сети.

8) В клетках  $(m+1, 4)$  и  $(m+1, 5)$  получаем признак выводимости переменных 4 и 5, а в клетке  $(2, n+1)$  формируем признак того, что правило уже было запущено, т.е. ставим цифру

2. После этого, проводим анализ служебной строки и видим, что не все искомые переменные известны. Значит необходимо продолжить обработку матрицы  $V$  размерности  $(m+1) \times (n+1)$ . Анализ этой матрицы показывает возможность запуска правила  $m$ , что и отражено на рисунке 17, а на рисунке 18 это же действие показано на примере построения двудольного графа миварной логической сети.

9) Продолжим дальнейшую обработку примера. При запуске правила  $m$  появляются новые значения, в том числе и для искомых переменных, что и показано на рисунке 19. На рисунке 20 это же действие показано на примере построения двудольного графа миварной логической сети.

10) Итак, в нашем примере в служебной строке больше не осталось искомых правил, а в клетках таблицы появились новые значения: в клетке  $(m, n+1) - 2$ , а в клетке  $(m+1, n-2)$  вместо значения  $w$  появилось значение  $z$ . Таким образом, получен положительный результат, следовательно, маршрут логического вывода при данных исходных значениях существует.

V	1	2	3	4	5	...	n-2	n-1	n	n+1
1	x	x	x					y	y	
2			x	y	y			x	x	
...						...				
m		x		x	x		y			
m+1	z	z	z				w			

Рисунок 7. Шаг № 1 примера обработки матрицы V.

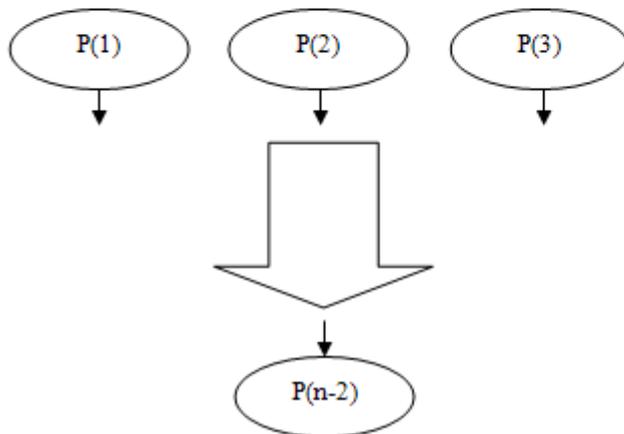
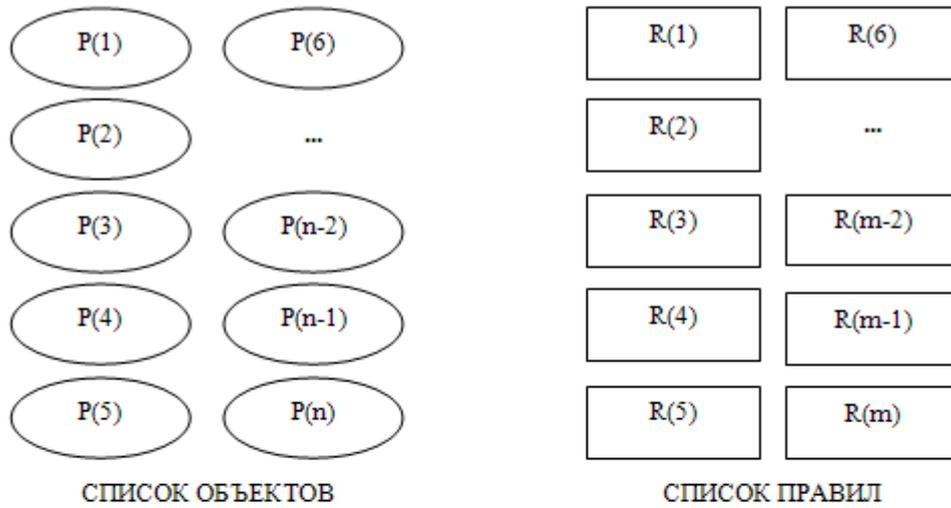


Рисунок 8. Иллюстрация при помощи двудольного графа шага № 1.

<b>V</b>	1	2	3	4	5	...	n-2	n-1	n	n+1
1	x	x	x					y	y	<b>1</b>
2			x	y	y			x	x	
...						...				
m		x		x	x		y			
m+1	<b>z</b>	<b>z</b>	<b>z</b>				<b>w</b>			

Рисунок 9. Шаг № 2 примера обработки матрицы V.

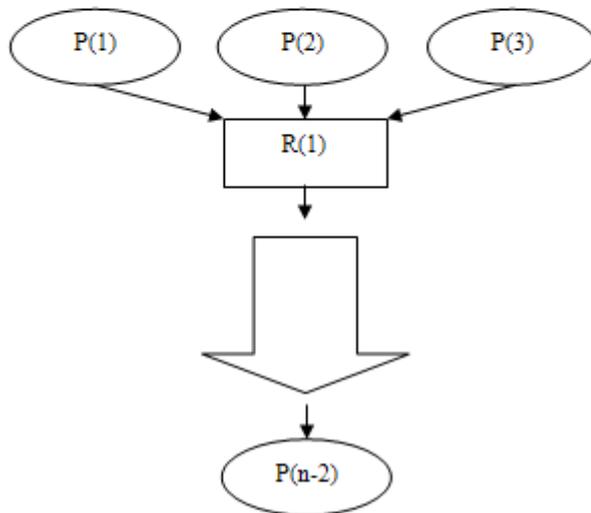
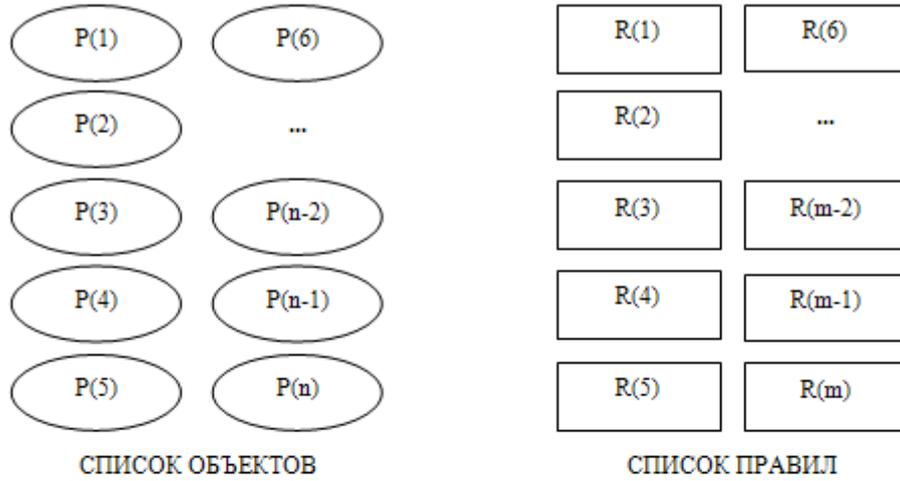


Рисунок 10. Иллюстрация при помощи двудольного графа шага № 2.

V	1	2	3	4	5	...	n-2	n-1	n	n+1
1	x	x	x					y	y	2
2			x	y	y			x	x	
...						...				
m		x		x	x		y			
m+1	z	z	z				w	z	z	

Рисунок 11. Шаг № 3 примера обработки матрицы V.

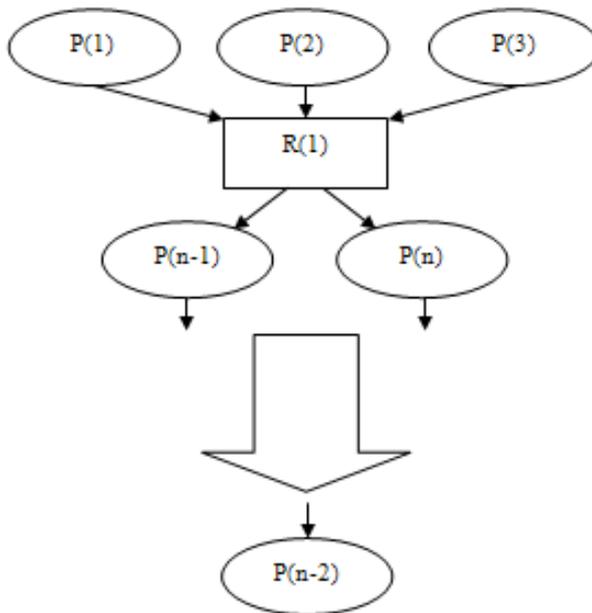
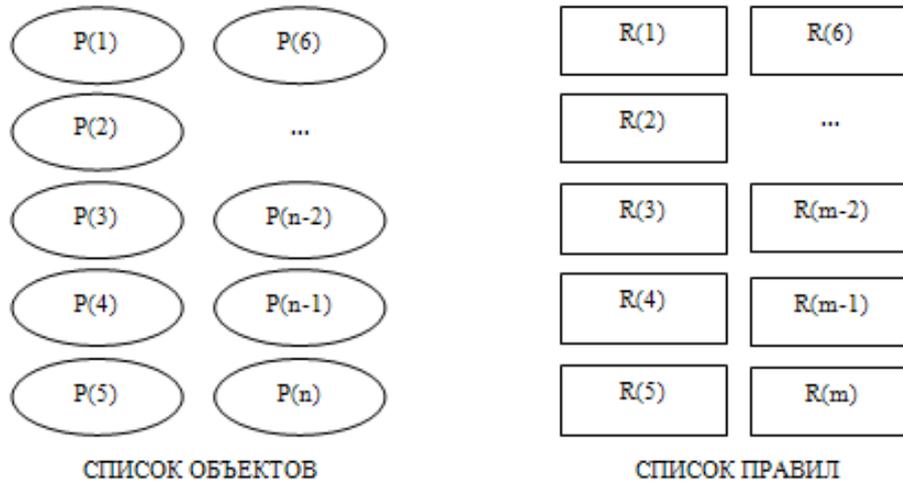


Рисунок 12. Иллюстрация при помощи двудольного графа шага № 3.

<b>V</b>	1	2	3	4	5	...	n-2	n-1	n	n+1
1	x	x	x					y	y	<b>2</b>
2			x	y	y			x	x	<b>1</b>
...						...				
m		x		x	x		y			
m+1	<b>z</b>	<b>z</b>	<b>z</b>				<b>w</b>	<b>z</b>	<b>z</b>	

Рисунок 13. Шаг № 4 примера обработки матрицы **V**.

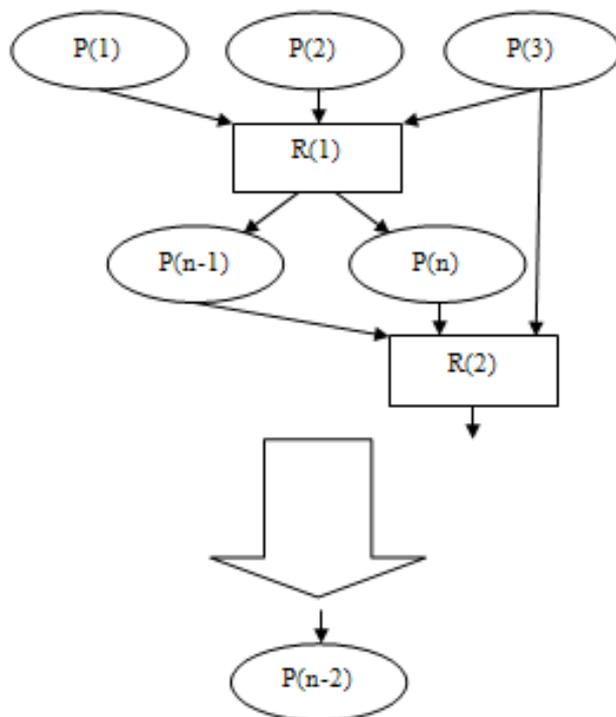
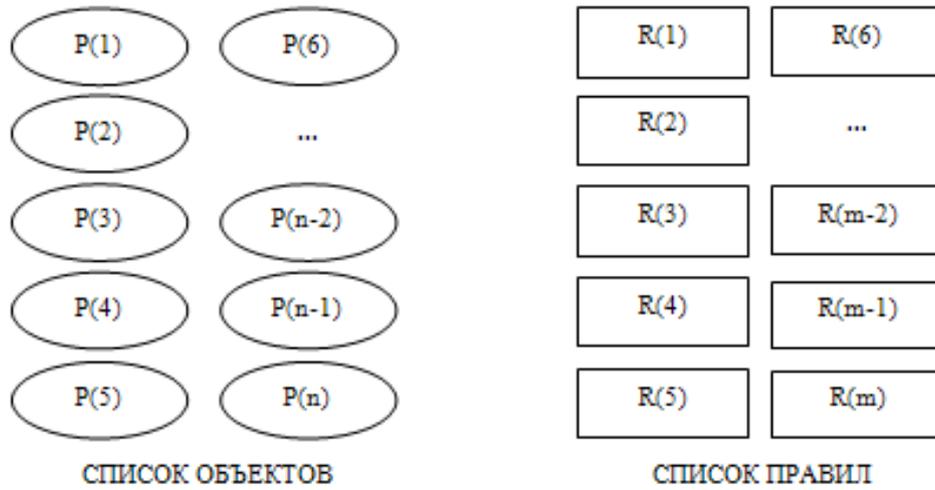


Рисунок 14. Иллюстрация при помощи двудольного графа шага № 4.

<b>V</b>	1	2	3	4	5	...	n-2	n-1	n	n+1
1	x	x	x					y	y	<b>2</b>
2			x	y	y			x	x	<b>2</b>
...						...				
m		x		x	x		y			
m+1	<b>z</b>	<b>z</b>	<b>z</b>	<b>z</b>	<b>z</b>		<b>w</b>	<b>z</b>	<b>z</b>	

Рисунок 15. Шаг № 5 примера обработки матрицы V.

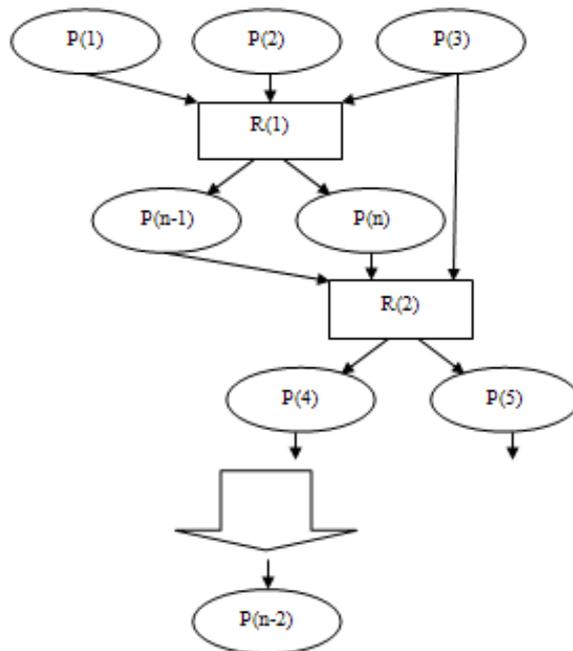
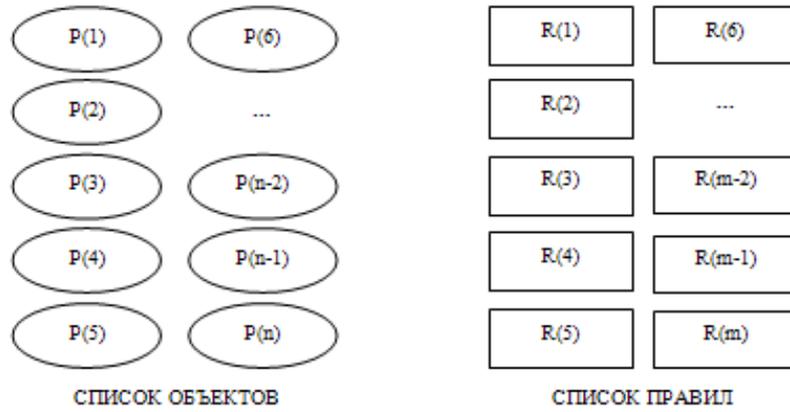


Рисунок 16. Иллюстрация при помощи двудольного графа шага № 5.

<b>V</b>	1	2	3	4	5	...	n-2	n-1	n	n+1
1	x	x	x					y	y	<b>2</b>
2			x	y	y			x	x	<b>2</b>
...						...				
m		x		x	x		y			<b>1</b>
m+1	<b>z</b>	<b>z</b>	<b>z</b>	<b>z</b>	<b>z</b>		<b>w</b>	<b>z</b>	<b>z</b>	

Рисунок 17. Шаг № 6 примера обработки матрицы **V**.

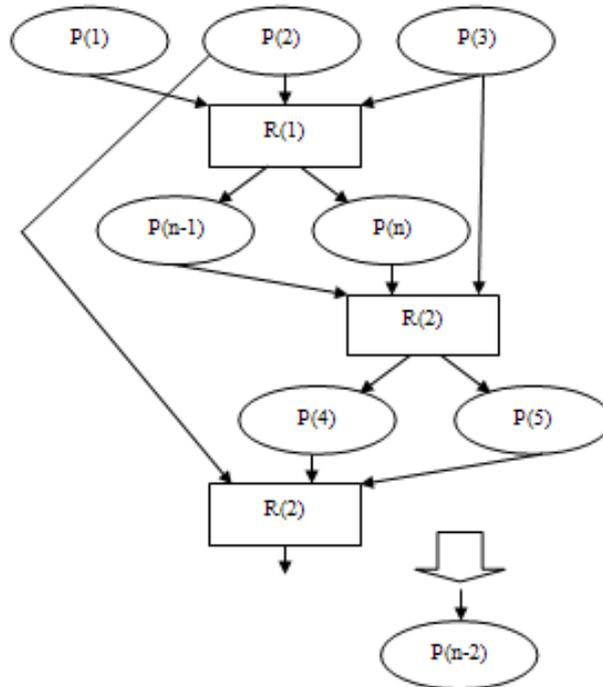
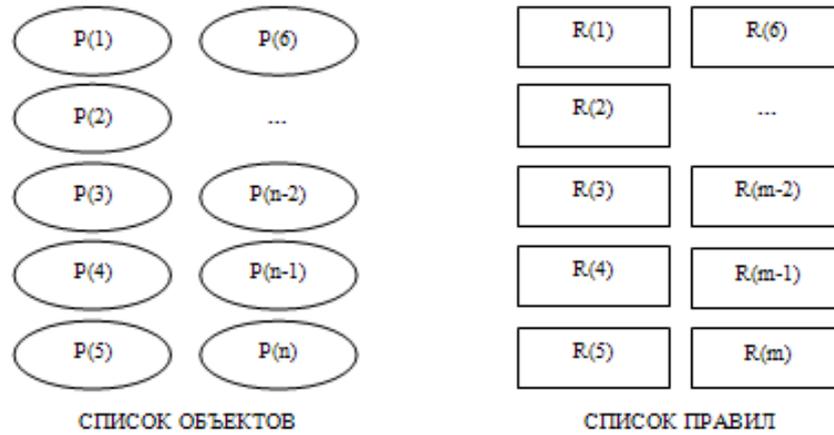


Рисунок 18. Иллюстрация при помощи двудольного графа шага №6.

<b>V</b>	1	2	3	4	5	...	n-2	n-1	n	n+1
1	x	x	x					y	y	2
2			x	y	y			x	x	2
...						...				
m		x		x	x		y			2
m+1	z	z	z	z	z		z(w)	z	z	

Рисунок 19. Шаг № 7 примера обработки матрицы V.

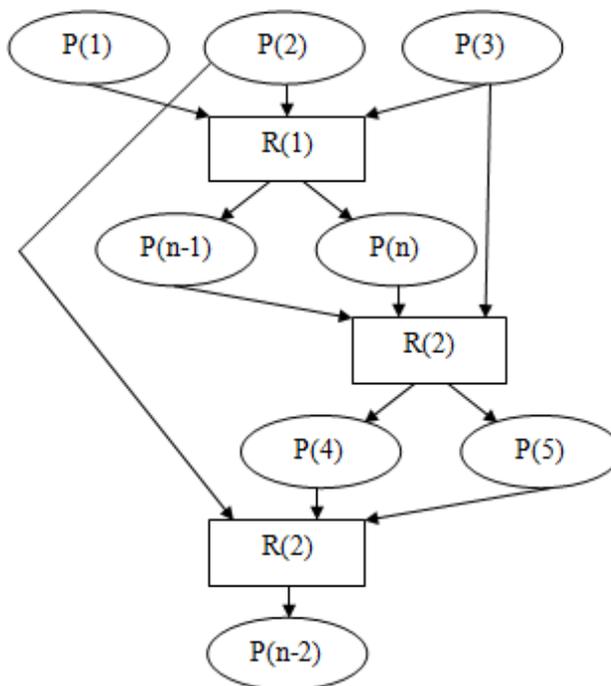
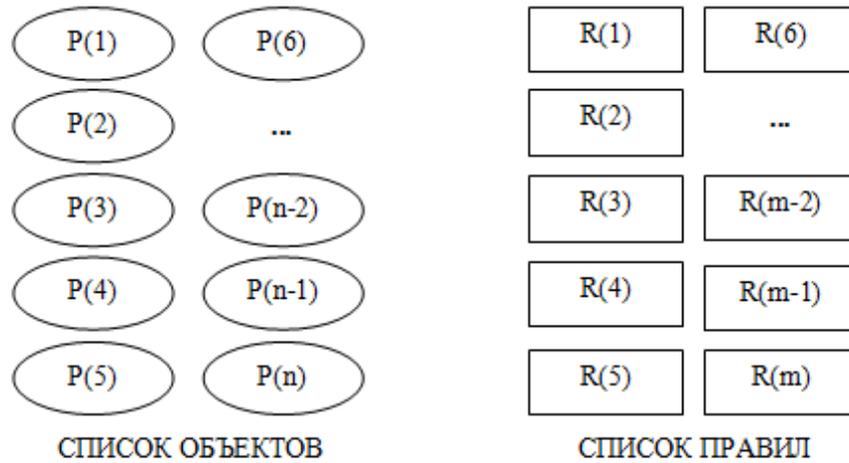


Рисунок 20. Иллюстрация при помощи двудольного графа шага №7.

### **Преимущества и новизна миварного метода обработки**

Таким образом, показано, что для экспертных систем на основе продукционного подхода и вместо традиционных однодольных графов необходимо и достаточно использовать двудольные графы миварных сетей. С научно-практической точки зрения основную сложность при использовании экспертных систем составляет именно концептуальное, продукционное, описание предметной области и формирование необходимых двух списков: объектов и правил для миварных логических сетей. Непосредственная обработка проводится на основе универсального механизма, описанного в [1-22]. Выделенные правила-процедуры при необходимости могут быть представлены в виде продукции, что соответствует традиционному подходу, но в то же время позволяет реализовать многомерное эволюционное представление данных. Универсальные возможности миварного подхода обусловлены тем, что он обобщает все известные модели данных, включая семантические сети, модель "сущность-связь" и онтологии [4, 10, 15, 18-20]. Преимуществами миварного подхода являются:

- 1) линейная вычислительная сложность и реальное время работы;
- 2) решение логических и вычислительных (и других) задач;
- 3) управление потоком входных данных и оперативная диагностика;
- 4) адаптивное описание и непрерывное решение задач;
- 5) активная работа с запросами или уточнениями входных данных на эволюционной сети правил и объектов (самообучение).

### **Теоретический расчет вычислительной сложности миварного метода обработки**

Общее количество действий при миварном методе определяется суммой действий на каждом этапе:

- 1) присваивание известных  $z$  и искомым  $w$  значений клеткам служебной строки  $(m+1)$  (общее количество таких действий не более  $n$ );
- 2) присвоение признака обработки правил в служебном столбце  $(n+1)$  (количество действий не более  $2m$ , но может быть не более числа  $m$ );
- 3) присваивание признака известности ( $z$ ) клеткам служебной строки  $(m+1)$  (общее количество таких действий не более  $n$ );
- 4) определение новых значений клеткам строки  $(m+1)$  (количество таких действий не более  $n$ ).

Действия на этапах 1, 3 и 4 выполняются над одним массивом данных -строкой  $(m+1)$ . При этом общее, суммарное, количество действий на всех этих этапах (1, 3 и 4) не должно превышать общего количества клеток в этой строке, т.к. обработанные значения "вычеркиваются" и более не обрабатываются. Получаем, что общее количество действий (**KD**) при миварном методе обработки, т.е. его вычислительная сложность, не превышает количества клеток в служебных частях матрицы:

$$O(m+n), \text{ т.е. } KD \leq (m+n)$$

В том случае, когда нельзя реализовать предложенные сокращения вычислений, этот метод решает задачу поиска маршрута логического вывода с вычислительной сложностью:

$$KD = O(mn),$$

т.е. имеет место линейная зависимость либо от количества правил, либо от количества переменных. В худшем случае по каждой клетке "проходят" один раз.

В работах [12-15, 20-22] и на сайтах [18-19] изложены результаты практической реализации в программном комплексе "УДАВ" миварных сетей. Таким образом, в работах [1-22] доказано, что существует достаточно универсальный миварный подход для решения многих практических задач применения информационных систем и процессов с одновременным проведением как логической, так и вычислительной обработки данных. Следовательно, существовавшее ранее противоречие между логическим выводом и вычислительной обработкой успешно преодолено с помощью миварных сетей.

### **Результаты практических исследований и экспериментов**

В настоящее время создано уже несколько программных комплексов - прототипов экспертных систем, которые реализовали на практике миварный метод обработки данных и правил. Прежде всего, это программный комплекс УДАВ [9-22], на основе которого начата реализация проекта "Миварная активная энциклопедия". В Миварной энциклопедии описано три предметные области: геометрия - "треугольники", "стереометрия" и "окружности". Независимо от этих работ, Сергушиным Г.С. реализован программный комплекс "физика падающего тела" [10, 18-19] и прототип миварной экспертной системы управления двигателем автомобиля. Коллективом авторов создан прототип экспертной системы - тренажера логических переключений электрических подстанций и распределенных электросетей с трехмерным интерфейсом. Этот тренажер позволяет адаптивно перестраивать конфигурацию логической схемы и самостоятельно добавлять пользователям новые элементы.

Как было указано выше, наибольшую трудность составляет описание предметной области в виде миварной сети. Поэтому для проведения вычислительных экспериментов был создан специальный программный комплекс, основанный на программе УДАВ. Прежде всего, в новом тестовом комплексе реализован генератор миварных матриц описания предметных областей. Такое решение было необходимо для создания миварных матриц больших размеров. Например, в процессе экспериментов были созданы миварные матрицы размерности 1 170 007 объектов на 3 510 015 правил, т.е. матрица миллион на 3,5 миллиона клеток. Для генерации таких матриц используется некоторый специально разработанный алгоритм, который по некоторым правилам формирует элементарные строки миварной матрицы.

На вход этой программе задается количество объектов - переменных. Далее генератор матриц сам генерирует простые арифметические правила для введенного количества объектов. В качестве примера простых правил могут использоваться следующие:

$$a+b=c \text{ и/или } c-b=a \text{ и/или } c-a=b$$

Генератор матриц выполняет формирование матрицы и сохраняет ее в специальный файл. Затем тестовая программа считывает тестовую матрицу и начинает обработку. Для однозначности и адекватности проводимых сравнений все генерируемые матрицы в определенном смысле являются подобными. Это обеспечивается за счет алгоритма формирования простых правил и заданием следующих управляющих данных. В качестве известных объектов всегда берутся объекты P1=10 и P2=10. А в качестве искомого объекта всегда задается последний по номеру объект. Таким образом, если количество входных параметров = 100, то программа будет искать P100, а если количество входных параметров = 10000, то программа будет искать P10000.

При необходимости, пользователь может сгенерировать свои матрицы или изменить уже созданные. Такие эксперименты были проведены. Данная тестовая программа написана только для оценки времени работы ядра УДАВ при элементарных правилах и с пропорциональным наращиванием миварных матриц.

В конце своей работы тестовая программа строит граф решения, где: розовым цветом показаны - параметры и внутри пишется его идентификатор, а оранжевым цветом показано - правило и внутри него тоже пишется идентификатор. Соответственно наверху графа рисуются исходные данные, а в внизу параметр который искали. Генератор устроен таким образом, чтобы решение всегда существовало, а его нахождение требовало пропорционального увеличения числа шагов искомого алгоритма решения задачи.

Отметим, что время работы программы определяется по внутренним часам компьютера и может сильно зависеть от общей нагрузки и параллельно выполняемых задач. В настоящий момент реализованы и однопоточный вариант программы, и многопоточный. Но большая часть экспериментов была проведена на однопоточном варианте, которому и уделено основное внимание. Подчеркнем, что тестовая программа не учитывает время создания тестовой матрицы на первом этапе и время формирования графа решения в конце.

Время решения задач выводится в миллисекундах (ms). Учитывая сформулированные особенности работы тестовой программы, всем испытателям было рекомендовано запускать тестовую программу с определенным шагом увеличения объектов и в относительно равнозначных условиях внешней загрузки компьютера. Например, было рекомендовано отключить все внешние функции и не запускать параллельно никаких программ, кроме тестовой. В процессе проведения экспериментов выявили, что основным ограничением по количеству обрабатываемых объектов - переменных является ресурс оперативной памяти компьютера и корректность работы с ней операционной системы.

Для проведения тестов были задействованы различные компьютеры, начиная от простейших нетбуков с оперативной памятью 512 Мб и до небольших серверов с процессорами Intel 3,8 ГГц и оперативной памятью 4 Гб. Количество ядер не влияло на скорость работы, т.к. основные тесты проводились с использованием однопоточной программы. Использовались два семейства операционных систем: Windows и MacOS.

Отметим, что на семействе операционных систем Windows удалось получить следующие результаты - до 150 000 объектов и 450 000 правил на компьютерах с оперативной памятью 4 Гб. В зависимости от процессора, время решения составляло от нескольких миллисекунд до 2 793 672 миллисекунд, что примерно составляет 47 минут.

Наибольший интерес представляют эксперименты, проведенные на ноутбуке MacBook, операционная система MacOSX версии 10.6.7, процессор Intel Core 2DUO с частотой 2 ГГц и с оперативной памятью DDR3 объемом 4 Гб. На этом ноутбуке в различные периоды времени были проведены тестовые испытания двух версий тестовой программы: В1 и В2. Результаты испытаний приведены на рисунке 21.

Итак, на обычном серийном ноутбуке MacBook не оптимизированная тестовая программа, написанная на языке ЯВА (не самом быстром языке), движок УДАВ обрабатывает миварные матрицы размерности миллион объектов на 3,5 миллиона параметров за время, примерно равное 3 часам. Испытания на максимальное значение получили следующий результат: 1 170 007 объектов - переменных, 3 510 015 правил и время решения 12 239 183 миллисекунды, что примерно равно 200 минутам и немногим более

3 часов. **3,5 миллиона правил** - это очень большое число. Для сравнения приведем следующие оценки: система управления крупными атомными станциями оперирует всего 20 000 правил; описание всей программы средней школы и технического ВУЗа потребует около 300 правил на каждую научную область и в сумме не превысит 100 000 правил. Напомним, что классические продукционные и предикатные логические системы не могли работать и с сотней правил.

Таким образом, возможности программного комплекса УДАВ и миварного метода обработки позволяют создать глобальную экспертную систему, в которой будут собраны все формализованные знания человечества. Это проект "Миварная активная энциклопедия".

В совокупности, эксперименты были проведены более чем на 15 компьютерах. Для каждого значения количества объектов - переменных проводилось более 10 однотипных испытания в примерно одинаковых условиях. Расчеты показали, что ошибка измерения времени работы программы не превышала 3 процентов. При этом все проведенные эксперименты подтвердили на практике линейную вычислительную сложность миварного метода обработки данных и правил. Следовательно, проведенные вычислительные эксперименты подтверждают теоретические выводы о линейной вычислительной сложности миварного метода обработки и о возможности создания глобальной мультипредметной экспертной системы "Миварная активная энциклопедия".

Если у кого-то еще остаются сомнения, то предлагается создать тестовый пример, разработать тестовые миварные матрицы и прислать их для проведения эксперимента. Будет выполнена загрузка матриц в миварный движок УДАВ, получен ответ и распечатан граф алгоритма решения тестовой задачи. При необходимости, можно провести совместные эксперименты, т.к. уже проведенные исследования достаточно полно позволили проверить и протестировать миварный движок УДАВ. Ждем предложений!

## Зависимость времени решения задач от количества объектов и правил миварной сети

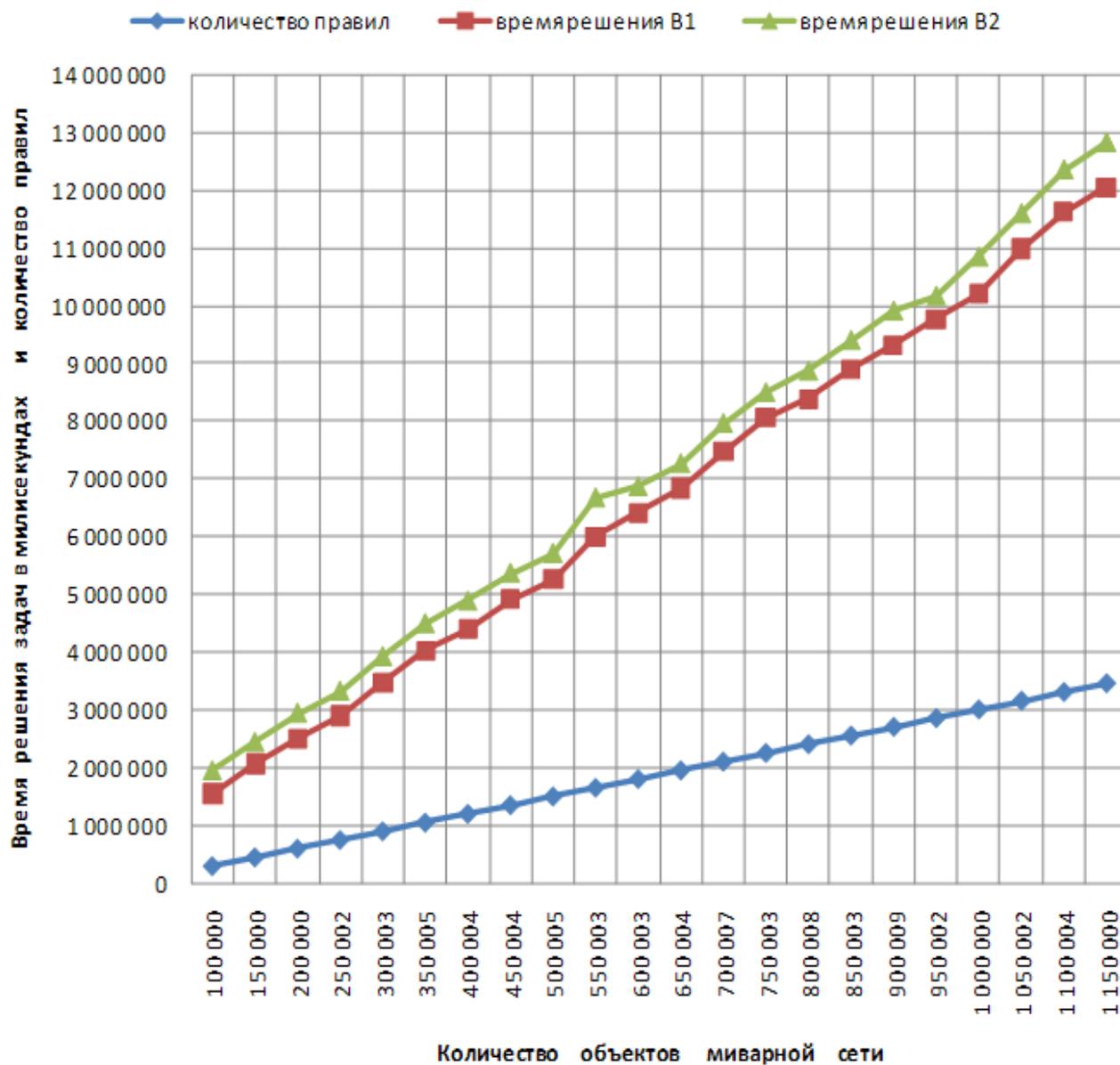


Рисунок 21. Результаты практических исследований и экспериментов.

## Выводы

Показан теоретический переход от однодольных продукционных систем к двудольным миварным логико-вычислительным сетям. Приведены примеры реализации миварных сетей в формализмах матриц и графов. Теоретически обоснована линейная вычислительная сложность автоматического конструирования алгоритмов из переменных-объектов и правил-процедур миварных сетей. В качестве миварных правил могут быть использованы различные сервисы, модули и вычислительные процедуры. Автоматический конструктор алгоритмов может использоваться для поиска логического вывода в области создания экспертных систем.

На основе миварных сетей создан программный комплекс УДАВ, который обрабатывает более 1,17 млн переменных и более 3,5 млн правил на обычных компьютерах и ноутбуках. Приведены результаты практических расчетов и решений различных прикладных задач, которые на практике подтверждают линейную вычислительную сложность конструирования алгоритмов в формализме миварных сетей. Программный комплекс УДАВ используется как для решения логических, так и вычислительных задач. Приведены сведения о практической реализации нескольких миварных экспертных систем. Миварные сети позволяют перейти к новому поколению экспертных систем и интеллектуальных пакетов прикладных программ. Миварный подход позволил на практике создать автоматические обучаемые эволюционные активные логически рассуждающие информационные системы. В перспективе на основе миварных сетей будет создана глобальная мультипредметная активная экспертная система под названием "Миварная активная энциклопедия".

Подробно описан линейной вычислительной сложности миварный метод логико-вычислительной обработки данных и правил. Обосновано, что для экспертных систем нового поколения вместо продукционного подхода и традиционных однодольных графов целесообразно использовать двудольные графы миварных сетей.

Практическая реализация миварного метода в программном комплексе УДАВ и проведение вычислительных экспериментов на различных компьютерах убедительно доказывает линейную вычислительную сложность миварного метода.

Впервые предоставлены возможности создания мультипредметных экспертных систем, способных на практике работать с миллионами объектов и правил миварных сетей.

Миварные технологии накопления и обработки информации значительно расширяют технические возможности компьютеров и позволяют создавать принципиально новые экспертные системы, интеллектуальные пакеты прикладных программ и интеллектуальные системы. На основе миваров могут быть построены активные обучаемые логически рассуждающие автоматические системы, которые будут служить прообразом систем искусственного интеллекта.

## Литература

1. Варламов О.О. Алгоритм разреза сети по вершинам и ребрам ее графа сложности  $O(n^2)$  // Труды НИИР: Сб. ст. - М., 1997. - 136 с. С. 92-97.
2. Варламов О.О. Разработка адаптивного механизма логического вывода на эволюционной интерактивной сети гиперправил с мультиактивизаторами, управляемой потоком данных // Искусственный интеллект. 2002. № 3. С. 363-370.

3. Варламов О.О. Разработка линейного матричного метода определения маршрута логического вывода на адаптивной сети правил// Известия вузов. Электроника. 2002. № 6. С.43-51.
4. Варламов О.О. Эволюционные базы данных и знаний для адаптивного синтеза интеллектуальных систем. Миварное информационное пространство. - М.: Радио и связь, 2002. - 288 С.
5. Варламов О.О. Основы многомерного информационного развивающегося (миварного) пространства представления данных и правил // Информационные технологии, 2003. № 5. С. 42 - 47.
6. Варламов О.О. Системный анализ и синтез моделей данных и методы обработки информации в самоорганизующихся комплексах оперативной диагностики: Диссертация на соискание ученой степени доктора технических наук. - М.: МАРТИТ, 2003. - 307 с.
7. Варламов О.О. Системный анализ и синтез моделей данных и методы обработки информации в самоорганизующихся комплексах оперативной диагностики: Автореферат диссертации на соискание ученой степени доктора технических наук. - М.: МАРТИТ, 2003. - 44 с.
8. Варламов О.О. Эволюционные базы данных и знаний. Миварное информационное пространство // Известия Южного федерального университета. Технические науки. 2007. Т. 77. № 2. С. 77-81.
9. Владимиров А.Н., Варламов О.О., Носов А.В., Потапова Т.С. Применение многопроцессорного вычислительного кластера НИИР для распараллеливания алгоритмов в научно-технических и вычислительных задачах // Труды Научно-исследовательского института радио. 2009. № 3. С. 120-123.
10. Санду Р.А., Варламов О.О. Миварный подход к созданию интеллектуальных систем и искусственного интеллекта. Результаты 25 лет развития и ближайшие перспективы. – М.: Стандартиформ, 2010. – 339 с.
11. Варламов О.О., Санду Р.А., Владимиров А.Н., Бадалов А.Ю., Чванин О.Н. Развитие миварного метода логико-вычислительной обработки информации для АСУ, тренажеров, экспертных систем реального времени и архитектур, ориентированных на сервисы // Труды НИИР. 2010. № 3. С. 18 - 26.
12. Владимиров А.Н., Варламов О.О., Носов А.В., Потапова Т.С. Программный комплекс «УДАВ»: практическая реализация активного обучаемого логического вывода с линейной вычислительной сложностью на основе миварной сети правил // Труды Научно-исследовательского института радио. 2010. Т. 1. С. 108-116.
13. Варламов О.О. Миварный подход к разработке интеллектуальных систем и проект создания мультипредметной активной миварной интернет-энциклопедии // Известия Кабардино-Балкарского научного центра РАН. 2011. № 1. С. 55-64.
14. Подкосова Я.Г., Васюгова С.А., Варламов О.О. Использование технологий виртуальной реальности для трехмерной визуализации результатов моделирования и для миварных обучающих систем // Известия Кабардино-Балкарского научного центра РАН. 2011. № 1. С. 226-232.
15. Варламов О.О. Обзор 25 лет развития миварного подхода к разработке интеллектуальных систем и создания искусственного интеллекта // Труды Научно-исследовательского института радио. 2011. № 1. С. 34-44.

16. Подкосова Я.Г., Варламов О.О., Остроух А.В., Краснянский М.Н. Анализ перспектив использования технологий виртуальной реальности в дистанционном обучении // Вопросы современной науки и практики. Университет им. В.И. Вернадского. 2011. № 2 (33). С. 104-111.
17. Подкосова Я.Г., Васюгова С.А., Варламов О.О. Новые возможности и ограничения технологий виртуальной реальности для проведения научных исследований, трехмерной визуализации результатов моделирования и создания миварных обучающих систем и тренажеров // Труды Научно-исследовательского института радио. 2011. № 2. С. 5-16.
18. Материалы веб-сайта д.т.н. Варламов О.О. // <http://www.ovar.narod.ru>. 2011.
19. Материалы веб-сайта компании МИВАР // <http://www.mivar.ru>. 2011.
20. Varlamov O.O. MIVAR technologies of the development of intelligent systems and the creation of the active multi-subject online MIVAR encyclopaedia // Pattern Recognition and Information Processing (PRIP'2011): proceedings of the 11th International Conference (18 - 20 may, Minsk, Republic of Belarus). - Minsk: BSUIR, 2011. 472 p. p. 326-329.
21. Podkosova Y.G., Vasiuhova S.A., Varlamov O.O. MIVAR learning systems, virtual reality and 3D – visualization of scientific modelling results // Pattern Recognition and Information Processing (PRIP'2011): proceedings of the 11th International Conference (18 - 20 may, Minsk, Republic of Belarus). - Minsk: BSUIR, 2011. 472 p. p. 447-450.
22. Nosov A.V., Adamova L.E., Varlamov O.O., Overchuk M.L. About the prospects of active Mivar's Internet-encyclopedia // Proceedings of the Congress on intelligent systems and information technologies "AIS-IT'10". Scientific publication in 4 volumes. – Moscow: Physmathlit, 2010, Vol. 4. – 128 p. ISBN 978-5-9221-1248-2. P. 118-119.
23. Поспелов Д.А. Моделирование рассуждений. Опыт анализа мыслительных актов. - М.: Радио и связь, 1989. - 184 с.
24. Кузнецов О.П. Дискретная математика для инженера. 6-е изд., стер. - СПб: Издательство "Лань", 2009. - 400 с.
25. Кук Д., Бейз Г. Компьютерная математика: Пер. с англ. - М.: Наука, Гл. ред. физ.-мат. лит., 1990. - 384 с.
26. Саймон А.Р. Стратегические технологии баз данных: менеджмент на 2000 год: Пер. с англ./ Под ред. и с предисл. М.Р. Когаловского. - М.: Финансы и статистика, 1999. - 479 с.
27. Марков А.С., Лисовский К.Ю. Базы данных. Введение в теорию и методологию: Учебник. - М.: Финансы и статистика, 2006. - 512 с.
28. Кузнецов С.Д. Базы данных: языки и модели. - М.: ООО "Бином-Пресс", 2008. - 720 с.
29. Когаловский М.Р. Энциклопедия технологий баз данных. - М.: Финансы и статистика, 2005. - 800 с.
30. Ахо А.В., Хопкрофт Д.Э., Ульман Д.Д. Структуры данных и алгоритмы. - М.: Издательский дом "Вильямс", 2007. - 400с.
31. Хорошевский В.Ф. Пространства знаний в сети Интернет и Semantic Web (Часть 1) // Искусственный интеллект и принятие решений, № 1, 2008. с. 80-97.
32. Люгер Дж.Ф. Искусственный интеллект: стратегии и методы решения сложных проблем, 4-е издание.: Пер. с англ. - М.: Издательский дом "Вильямс", 2005. - 864 с.