

Министерство образования и науки Российской Федерации

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«САРАТОВСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИМЕНИ Н. Г. ЧЕРНЫШЕВСКОГО»

Кафедра математической
кибернетики и компьютерных наук

**ПОСТРОЕНИЕ ЭКСПЕРТНЫХ СИСТЕМ НА ОСНОВЕ МИВАРНОЙ
КОНЦЕПЦИИ ПРЕДСТАВЛЕНИЯ ЗНАНИЙ**

АВТОРЕФЕРАТ БАКАЛАВРСКОЙ РАБОТЫ

студента 4 курса 451 группы
направления 09.03.04 — Программная инженерия
факультета КНиИТ
Рудакова Александра Александровича

Научный руководитель

к. ф.-м. н., доцент

А. С. Иванов

Заведующий кафедрой

к. ф.-м. н.

С. В. Миронов

Саратов 2016

СОДЕРЖАНИЕ

ВВЕДЕНИЕ	3
1 Традиционные подходы к описанию знаний.....	4
2 Миварные сети	5
3 Реализация программного комплекса для анализа работы миварной сети	8
3.1 Описание генератора миварных матриц	8
3.2 Описание программы, создающей миварную матрицу по спис- кам правил и переменных, полученных после работы генератора .	9
3.3 Описание программы, использующей миварный метод обработ- ки информации	10
3.4 Результаты обработки миварной сети	10
ЗАКЛЮЧЕНИЕ	12
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ	13

ВВЕДЕНИЕ

Системы, способные решать задачи, связанные с принятием решений логическими рассуждениями над определенным полем знаний называются интеллектуальными (экспертными) системами.

Традиционный подход к созданию экспертных систем использует продукции, однако этот способ зачастую не является наилучшим: (например, время работы таких систем неприемлемо большое). Подход к созданию интеллектуальных систем нового поколения — миварный подход — позволяет создавать экспертные системы, способные автоматически обучаться и рассуждать, при этом обрабатывать большое количество данных по сравнению с интеллектуальными системами, разработанными с помощью традиционных методов.

Целью данной работы является рассмотрение миварного метода обработки информации, разработка экспертной системы, использующей этот метод и вспомогательного средства — генератора списков правил и переменных для получения больших объемов экспериментальных данных.

В структуру бакалаврской работы входят: введение, три главы («Подходы к описанию знаний в экспертных системах», «Описание миварного метода» и «Реализация программного комплекса для анализа работы миварной сети»), заключение.

В первой главе описываются традиционные подходы к описанию знаний, при этом основной объем занимает рассмотрение продукционной модели, так как именно она является основой для развития миварных сетей.

Вторая глава содержит описание миварного метода. Приведен переход от логических «И-ИЛИ» сетей к двудольным графам, а также показан способ логико-вычислительной обработки данных.

В третьей главе рассматриваются части программного комплекса, использующего миварный метод обработки данных.

1 Традиционные подходы к описанию знаний

Традиционные классы моделей представления знаний:

- продукционная модель;
- семантические сети;
- фреймы;
- формальные логические модели.

Продукционная модель — модель, основанная на правилах. В общем виде продукционная модель позволяет представить знания в виде предложений «Если..., то...». С помощью продукций можно представлять различные формулы, правила, процедуры. [1]

Наиболее распространенные методы выводов, лежащие в основе большинства достоверных рассуждений — описания в виде деревьев или сетей (графов) вывода. Вершинам дерева (сети) соответствуют некоторые утверждения F_i , а дуги (ребра) определяют логические связи между этими утверждениями. Должны быть определены утверждения, известные априори. С помощью логических связей можно производить выводы о значениях неизвестных ранее утверждений. Таким образом, для вывода значения вершины, не соединенной непосредственно с уже известными вершинами, необходимо вычислить значения промежуточных. Формализованным представлением таких систем является однодольный граф. [2]

Основная проблема, с которой сталкиваются проектировщики систем, основанных на подобных схемах вывода — факториальный рост вычислительной сложности при увеличении количества правил. Это обусловлено тем, что любой вывод носит переборный характер. При таком подходе современные компьютеры способны обрабатывать маленькое количество правил, при этом многие предметные области требуют ввода многих тысяч правил и переменных. [3] [4]

2 Миварные сети

Как было указано выше, традиционный подход к описанию сети правил — представление сети в виде однодольного графа. Однако сети правил и процедур целесообразнее представлять в виде двудольных графов.

Миварные сети могут быть представлены в виде двудольного графа, при этом одному классу принадлежат объекты переменные, другому — правила-процедуры (развития продукций). При этом дугами будут обозначены связи между правилами и переменными. Например, правило $R1$ использует переменные $P1$ и $P2$ для нахождения переменной $P3$. Стоит заметить, что переменные являются объектами, и, в зависимости от предметной области могут принимать отличные от числовых значения. [4]

На практике сначала составляют списки объектов ($P1...Pn$) и правил ($R1...Rm$). Выделение их из предметной области — самый трудный шаг, который выполняется человеком. Далее появляется возможность передать управление обработкой данных непосредственно ЭВМ.

Выделяют три основных этапа миварной обработки:

1. формирование матрицы описания предметной области, основу которой представляют списки объектов и правил;
2. обработка матрицы и выведение алгоритма решения задачи;
3. выполнение вычислений по полученному алгоритму.

Для миварной сети с n переменными и m правилами строится матрица $V(m \times n)$. Строки соответствуют правилам; при этом все входные переменные для определенного правила обозначаются X , все выходные — Y . Кроме того, добавляются дополнительные строка и столбец. В дополнительной строке все искомые переменные отмечаются буквой W , а все переменные и уже известным значением (заданным изначально или полученным в результате работы) отмечаются буквой Z . [4]

В итоге получается матрица $V((m+1) \times (n+1))$, структура которой может меняться во время обработки данных. Далее данная матрица используется для поиска маршрута логического вывода.

Маршрут вывода строится следующим образом:

1. В служебной строке $m + 1$ известные объекты помечают z , а искомые — w ;
2. Производится поиск таких правил, у которых известны все входные пере-

менные. Если такие правила есть, то в служебном столбце $n + 1$ делается пометка, например в ячейке с номером правила ставится буква L (также можно делать пометку n — номер текущей итерации). Если ни одного правила не найдено, то логического вывода нет; необходимо добавить данные.

3. Далее, если существует несколько правил, у которых известны все входные переменные, производится выбор правила, которое должно быть активизировано в первую очередь.
4. Производится запуск (имитация запуска) правила: переменные, которые можно вывести из этого правила, помечаются в служебной строке $m + 1$ буквами Z .
5. После имитации запуска анализируют матрицу на предмет оставшихся искомым переменных (т.е. проверяют, остались ли в служебной строке ячейки с буквой W .) Если таких переменных нет, то задача считается завершённой, а порядок запуска правил образует маршрут логического вывода.
6. В случае, если остались ненайденные переменные, возвращаются к шагу 2 и проверяют, появились ли правила, у которых известны все переменные. Если таковых нет, то маршрута логического вывода дальше нет и нужен запрос на добавление необходимых данных.

После пошагового выполнения этого метода в матрице V не осталось неизвестных переменных.

Таблица 1 – Матрица V после выполнения

	1	2	3	4	5	...	$n - 2$	$n - 1$	n	$n + 1$
1	X	X	X					Y	Y	L
2			X	Y	Y			X	X	L
...						...				
m		X		X	X		Y			L
$m + 1$	Z	Z	Z	Z	Z		Z	Z	Z	

Закончено построение двудольного графа (см. рисунок 1.)

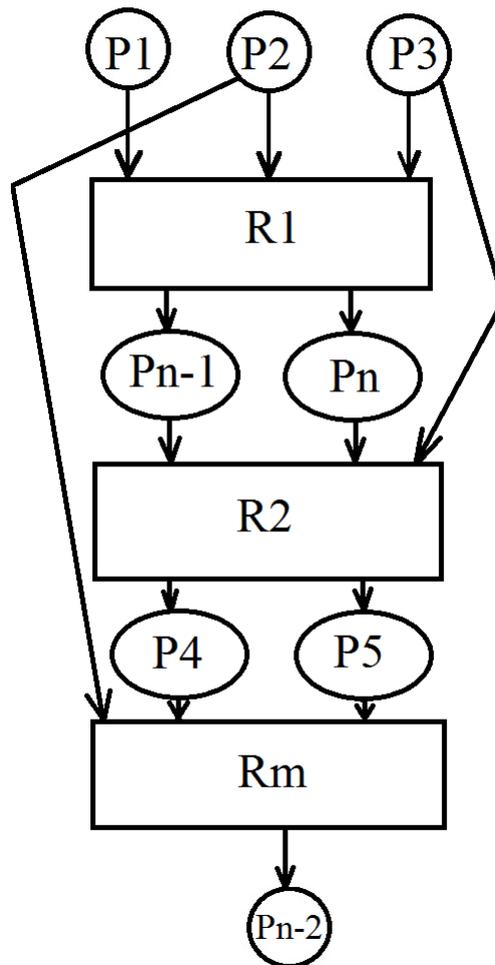


Рисунок 1 – Пример получившегося двудольного графа миварной сети

3 Реализация программного комплекса для анализа работы миварной сети

В ходе выполнения бакалаврской работы был построен программный комплекс, позволяющий выполнять следующие задачи:

- выполнять вычисления (логические выводы) с помощью миварного способа обработки данных;
- делать измерения времени работы программы для выполнения таких вычислений;
- генерировать необходимое количество данных, подлежащих обработке.

Таким образом, данный программный комплекс включает в себя следующие компоненты:

- экспертную систему, построенную с помощью миварного метода;
- генератор списков правил и переменных;
- обработчик полученных от генератора данных: выполняет переход от списков правил и переменных к миварной матрице, используемой непосредственной экспертной системой.

Программный комплекс разработан на языке программирования Python.

3.1 Описание генератора миварных матриц

В ходе выполнения бакалаврской работы был реализован генератор списков объектов и правил, предназначенный для создания базы знаний по заданному количеству объектов миварной сети. Программа выполнена на языке Python.

Для вывода генератор использует файл `mivar.txt`, который будет создан в том же каталоге, откуда запускается генератор. В случае, если этот файл уже существует, программа сначала произведет его очистку.

Так как основной задачей генератора являлась генерация искусственных списков правил и объектов для последующих замеров времени, работа функции построена по следующей схеме:

- на вход подаются значения для двух первых объектов сети, а также количество объектов, необходимое в построенном списке — p_1 , p_2 и n соответственно;
- так как в данном случае важную роль играло количество правил, а не сложность, генератор использовал простые арифметические правила для

введенного количества объектов; каждое правило должно было получать на вход значения двух объектов и в результате его выполнения получался другой объект;

- генерация простых правил происходила следующим образом: для каждого объекта (кроме первых двух) система псевдослучайным образом выбирала арифметическую операцию и два элемента, уже описанных ранее: один из них выбирался псевдослучайным образом, второй элемент всегда был предыдущим по списку (это необходимо для того, чтобы исключить случаи, когда для поиска последнего элемента нет необходимости считать все предыдущие). Построив правило для получения нового объекта, программа строила еще два новых правила, в которых результатами выступали соответственно два объекта, участвовавших в получении нового. Например: для получения нового объекта P_6 использовались объекты P_1 и P_2 . Далее строилось правило $P_3 = P_1 - P_2$. Из этого правила выражались $P_1 = P_3 + P_2$ и $P_2 = P_1 - P_3$. Таким образом достигалось наполнение списка правил; [4]
- формат выходного файла: в первой строке записано n — количество объектов-переменных, во второй m — количество правил; далее пустая строка; в последующих n строках записаны значения объектов (если значение априори неизвестно, то стоит 0). После пустой строки находятся m правил.

3.2 Описание программы, создающей миварную матрицу по спискам правил и переменных, полученных после работы генератора

Была разработана программа для анализа файла, полученного в результате работы генератора списков правил и объектов. Для работы программы необходим файл `mivar.txt`, находящийся в каталоге, откуда запускается программа.

Из файла программа выбирает первые две строки, в которых записаны n и m — количество объектов и правил соответственно. Далее производится пропуск одной строки, после чего выбираются значения объектов-переменных, которые заносятся в специальный список. В последнюю очередь программа считывает строки с описанием правил, производит их обработку, и из строк вида $P_5 = P_4 + P_2$ получает списки вида ['+', '2', '3', '1']. Порядок элементов в таком списке: арифметическая операция, индекс получаемого объекта

после применения правила, а также индексы объектов, требуемых правилу.

Таким образом, программа возвращает количество переменных и правил (n и m соответственно), список значений переменных, списки с описанием правил и миварную матрицу $V((m + 1) \times (n + 1))$.

3.3 Описание программы, использующей миварный метод обработки информации

Была реализована интеллектуальная система, использующая миварный метод обработки информации. Данная система получает на вход миварную матрицу, список объектов-переменных и список правил, а выводит результат выполнения и время работы миварного метода. Программа настроена таким образом, что на выход всегда подается значение последнего объекта-переменной (например, если количество объектов $n = 5$, то выводить она будет P_5). Это необходимо для адекватности и чистоты дальнейших экспериментов. [4].

3.4 Результаты обработки миварной сети

Для запуска тестовой программы использовался интерпретатор CPython 2.7.9 в операционной системе Ubuntu Linux 16.04. ЭВМ, на которой производились вычисления, имеет процессор с тактовой частотой 3,4 ГГц и 8 ГБ ОЗУ. Многопоточные вычисления не использовались.

Была выполнена обработка баз знаний, искусственно созданных с помощью генератора списков объектов и правил. Произведено несколько тестовых запусков, с разным количеством входных объектов-переменных и правил (от 100 до 100000 переменных). В каждом случае в качестве искомого элемента выбирался последний; два первых априори известны (P_1 и P_2). Результаты работы миварного метода — количество объектов, правил и время, затраченное на поиск представлены в таблице 2.

Таблица 2 – Результаты замеров времени работы программы с соответствующим количеством переменных

Количество переменных	Количество правил	Время работы (с)
100	294	0,0111
1000	2994	0,9761
5000	14994	12,938
10000	29994	366,189
50000	149994	709,115
100000	299994	1755,76

ЗАКЛЮЧЕНИЕ

В бакалаврской работе показан переход от традиционных подходов созданию экспертных систем — сетей продукций к миварным логиковычислительным сетям; приведена реализация интеллектуальной системы, основанной на миварном методе обработки информации, а также измерения ее работы. Таким образом, миварный подход позволяет на практике создать автоматическую рассуждающую экспертную систему, при этом простота метода обеспечивает как быструю реализацию, так и качественную работу такой системы.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

- 1 Искусственный интеллект и ЭС.— URL: http://expro.ksu.ru/materials/ii_i_es/book.html (Дата обращения 02.05.2016). Загл. с экр. Яз. рус.
- 2 Практическая реализация линейной вычислительной сложности логического вывода на правилах «если, то» в миварных сетях и обработка более трех миллионов правил.— URL: [http://auts.esrae.ru/pdf/2013/1\(3\)/66.pdf](http://auts.esrae.ru/pdf/2013/1(3)/66.pdf) (Дата обращения 23.05.2016). Загл. с экр. Яз. рус.
- 3 Продукционная модель знаний.— URL: <http://aiportal.ru/articles/knowledge-models/production-model.html> (Дата обращения 17.04.2016). Загл. с экр. Яз. рус.
- 4 *Варламов, О. О.* Мивар: переход от продукций к двудольным миварным сетям и практическая реализация автоматического конструктора алгоритмов, управляемого потоком входных данных и обрабатывающего более трех миллионов правил / О. О. Варламов. — 2011.