

Министерство образования и науки РФ  
Федеральное государственное бюджетное образовательное учреждение  
высшего образования  
**АМУРСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ**  
**(ФГБОУ ВО «АмГУ»)**

**АНАЛИЗ ДАННЫХ**  
**сборник учебно-методических материалов**  
для направления подготовки 38.03.05 – Бизнес-информатика

Благовещенск, 2017

*Печатается по решению  
редакционно-издательского совета  
факультета математики и информатики  
Амурского государственного  
университета*

*Составитель: Акилова И.М.*

Анализ данных: сборник учебно-методических материалов для направления подготовки для направления подготовки 38.03.05 «Бизнес информатика». – Благовещенск: Амурский гос. ун-т, 2017.

© Амурский государственный университет, 2017  
© Кафедра Информационных и управляющих систем, 2017  
© Акилова И.М., составление

## КРАТКОЕ ИЗЛОЖЕНИЕ ЛЕКЦИОННОГО МАТЕРИАЛА

### Методы и средства интеллектуального анализа данных

**Интеллектуальный анализ данных** — одно из новых направлений искусственного интеллекта. Этот термин является кратким и весьма неточным переводом с английского языка терминов *Data Mining* и *Knowledge Discovery in Databases (DM&KDD)*. Более точный перевод — «добыча данных» и «выявление знаний в базах данных».

*Data Mining*— это процесс обнаружения в сырых данных (raw data) ранее неизвестных, нетривиальных, практически полезных, доступных интерпретации знаний (закономерностей), необходимых для принятия решений в различных сферах человеческой деятельности (Г.Пятецкий-Шапиро).

Появление технологий *DM&KDD* обусловлено накоплением огромных объемов информации в компьютерных базах данных, которые стало невыгодно хранить и которыми стало трудно пользоваться традиционными способами. Последнее обстоятельство связано со стремительным развитием вычислительной техники и программных средств для представления и обработки данных. Большие объемы накопленных данных постоянно приходится модифицировать из-за быстрой смены аппаратного и программного обеспечения БД, при этом неизбежны потери и искажение информации. Одним из средств для преодоления подобных трудностей является создание информационных хранилищ данных, доступ к которым не будет сильно зависеть от изменения данных во времени и от используемого программного обеспечения. Другой подход ориентирован на сжатие больших объемов данных путем нахождения некоторых общих закономерностей (знаний) в накопленной информации. Оба направления актуальны с практической точки зрения. Второй подход более интересен для специалистов в области ИИ, так как связан с решением проблемы приобретения новых знаний. Следует заметить, что наиболее плодотворным является сочетание обоих направлений.

Наличие хранилища данных — необходимое условие для успешного проведения всего процесса *KDD*. Хранилищем данных называют предметно-ориентированное, интегрированное, привязанное ко времени, неизменяемое собрание данных, используемых для поддержки процесса принятия управленческих решений. Предметная ориентация означает, что данные объединены в категории и хранятся в соответствии с теми областями, которые они описывают, а не в соответствии с приложениями, которые их используют. Такой принцип хранения гарантирует, что отчеты, сгенерированные различными аналитиками, будут опираться на одну и ту же совокупность данных. Привязанность ко времени означает, что хранилище можно рассматривать как собрание *исторических* данных, т.е. конкретные значения данных однозначно связаны с определенными моментами времени. Атрибут времени всегда явно присутствует в структурах хранилищ данных. Данные, занесенные в хранилище, уже не изменяются в отличие от оперативных систем, где присутствуют только последние, постоянно изменяемые версии данных. Для хранилищ данных характерны операции добавления, а не модификации данных. Современные средства администрирования хранилищ данных обеспечивают эффективное взаимодействие с программным инструментарием *DM* и *KDD*. В качестве примера можно привести разработки компании SAS Institute: *SAS Warehouse Administrator* и *SAS Enterprise Miner*.

Рассмотрим простой пример, иллюстрирующий технологии *DM&KDD*. В базах данных можно хранить большую таблицу значений переменных  $X$  и  $Y$ , но если удалось установить зависимость между этими переменными, то без существенных потерь информации можно значительно сократить объем занимаемой памяти, поместив туда найденную зависимость, например  $Y = \sin(kX)$ . В общем случае зависимости, выявляемые в базах данных, могут быть представлены правилами, гипотезами, моделями нейронных сетей и т.п. Интеллектуальные средства извлечения информации позволяют почерпнуть из БД более глубокие сведения, чем традиционные системы оперативной обработки транзакций (*OLTP* — *On-Line Transaction Processing*) и оперативной аналитической обработки (*OLAP*). Выведенные из данных зако-

номерности и правила можно применять для описания существующих отношений и закономерностей, а также для принятия решений и прогнозирования их последствий.

Извлечение знаний из БД является одной из разновидностей машинного обучения, специфика которой заключается в том, что реальные БД, как правило, проектируются без учета потребностей извлечения знаний и содержат ошибки.

В технологиях *DM&KDD* используются различные математические методы и алгоритмы: классификация, кластеризация, регрессия, прогнозирование временных рядов, ассоциация, последовательность.

*Классификация*— инструмент обобщения. Она позволяет перейти от рассмотрения единичных объектов к обобщенным понятиям, которые характеризуют некоторые совокупности объектов и являются достаточными для распознавания объектов, принадлежащих этим совокупностям (классам). Суть процесса формирования понятий заключается в нахождении закономерностей, свойственных классам. Для описания объектов используются множества различных признаков (атрибутов). Проблема формирования понятий по признаковым описаниям была сформулирована М. М. Бонгартом (Михаил Моисеевич). Ее решение базируется на применении двух основных процедур: обучения и проверки. В процедурах обучения строится классифицирующее правило на основе обработки обучающего множества объектов. Процедура проверки (экзамена) состоит в использовании полученного классифицирующего правила для распознавания объектов из новой (экзаменационной) выборки. Если результаты проверки признаны удовлетворительными, то процесс обучения заканчивается, в противном случае классифицирующее правило уточняется в процессе повторного обучения.

*Кластеризация*— это распределение информации (записей) из БД по группам (кластерам) или сегментам с одновременным определением этих групп. В отличие от классификации здесь для проведения анализа не требуется предварительного задания классов.

*Регрессионный анализ* используется в том случае, если отношения между атрибутами объектов в БД выражены количественными оценками. Построенные уравнения регрессии позволяют вычислять значения зависимых атрибутов по заданным значениям независимых признаков.

*Прогнозирование временных рядов* является инструментом для определения тенденций изменения атрибутов рассматриваемых объектов с течением времени. Анализ поведения временных рядов позволяет прогнозировать значения исследуемых характеристик.

*Ассоциация* позволяет выделить устойчивые группы объектов, между которыми существуют неявно заданные связи. Частота появления отдельного предмета или группы предметов, выраженная в процентах, называется *распространенностью*. Низкий уровень распространенности (менее одной тысячной процента) говорит о том, что такая ассоциация не существенна. Ассоциации записываются в виде правил:  $A \Rightarrow B$ , где  $A$ — посылка,  $B$ — следствие. Для определения важности каждого полученного ассоциативного правила необходимо вычислить величину, которую называют *доверительностью*  $A$  к  $B$  (или взаимосвязь  $A$  и  $B$ ). Доверительность  $\delta(A/B)$  показывает, как часто при появлении  $A$  появляется  $B$ , и рассчитывается как  $\delta(A/B) = \varepsilon(A \cap B) / \varepsilon(A)$ , где  $\varepsilon(A \cap B)$ — распространенность совместного появления  $A$  и  $B$ ;  $\varepsilon(A)$  - распространенность  $A$ . Например, если  $\delta(A/B) = 20\%$ , то это значит, что при покупке товара  $A$  в каждом пятом случае приобретается и товар  $B$ . Необходимо отметить, что если  $\varepsilon(A) \neq \varepsilon(B)$ , то  $\delta(A/B) \neq \delta(B/A)$ . В самом деле, покупка компьютера влечет за собой покупку дисков, но покупка дисков не ведет к покупке компьютера. Важной характеристикой ассоциации является мощность, которая рассчитывается по формуле  $M(A/B) = \delta(A/B) / \varepsilon(B)$ . Чем больше мощность, тем сильнее влияние, которое наличие  $A$  оказывает на появление  $B$ .

Типичным примером применения ассоциации является анализ структуры покупок. Например, при проведении исследования в супермаркете можно установить, что 65 % купивших картофельные чипсы берут также и «кока-колу», а при наличии скидки за такой комплект «колу» приобретают в 85 % случаев. Подобные результаты представляют ценность при формировании маркетинговых стратегий.

*Последовательность* — это метод выявления ассоциаций во времени. В данном случае определяются правила, которые описывают последовательное появление определенных групп событий. Такие правила необходимы для построения сценариев. Кроме того, их можно использовать, например, для формирования типичного набора предшествующих продаж, которые могут повлечь за собой последующие продажи конкретного товара.

К интеллектуальным средствам *DM&KDD* относятся нейронные сети, деревья решений, индуктивные выводы, методы рассуждения по аналогии, нечеткие логические выводы, генетические алгоритмы, алгоритмы определения ассоциаций и последовательностей, анализ с избирательным действием, логическая регрессия, эволюционное программирование, визуализация данных. Иногда перечисленные методы применяются в различных комбинациях.

*Нейронные сети* относятся к классу нелинейных адаптивных систем с архитектурой, условно имитирующей нервную ткань, состоящую из нейронов. Математическая модель нейрона представляет собой некий универсальный нелинейный элемент, допускающий возможность изменения и настройки его характеристик. Подробнее вопросы построения моделей нейронных сетей рассмотрены в главе 5. Нейронные сети широко применяются для решения задач классификации. Построенную сеть сначала нужно «обучить» на примерах, для которых известны значения исходных данных и результаты. Процесс «обучения» сети заключается в подборе весов межнейронных связей и модификации внутренних параметров активационной функции нейронов. «Обученная» сеть способна классифицировать новые объекты (или решать другие примеры), однако правила классификации остаются не известными пользователю.

*Деревья решений* — метод структурирования задачи в виде древовидного графа, вершины которого соответствуют продукционным правилам, позволяющим классифицировать данные или осуществлять анализ последствий решений. Этот метод дает наглядное представление о системе классифицирующих правил, если их не очень много. Простые задачи решаются с помощью этого метода гораздо быстрее, чем с использованием нейронных сетей. Для сложных проблем и для некоторых типов данных деревья решений могут оказаться неприемлемыми. Кроме того, для этого метода характерна проблема значимости. Одним из последствий иерархической кластеризации данных является то, что для многих частных случаев отсутствует достаточное число обучающих примеров, в связи с чем классификацию нельзя считать надежной. Методы деревьев решений реализованы во многих программных средствах, а именно: *C5.0* (RuleQuest, Австралия), *Clementine* (Integral Solutions, Великобритания), *SIPINA* (University of Lyon, Франция), *IDIS* (Information Discovery, США).

*Индуктивные выводы* позволяют получить обобщения фактов, хранящихся в БД. В процессе индуктивного обучения может участвовать специалист, поставляющий гипотезы. Такой способ называют *обучением с учителем*. Поиск правил обобщения может осуществляться без учителя путем автоматической генерации гипотез. В современных программных средствах, как правило, сочетаются оба способа, а для проверки гипотез используются статистические методы. Примером системы с применением индуктивных выводов является *XpertRule Miner*, разработанная фирмой Attar Software Ltd. (Великобритания).

*Рассуждения на основе аналогичных случаев* (*Case-based reasoning* — *CBR*) основаны на поиске в БД ситуаций, описания которых сходны по ряду признаков с заданной ситуацией. Принцип аналогии позволяет предполагать, что результаты похожих ситуаций также будут близки между собой. Недостаток этого подхода заключается в том, что здесь не создается каких-либо моделей или правил, обобщающих предыдущий опыт. Кроме того, надежность выводимых результатов зависит от полноты описания ситуаций, как и в процессах индуктивного вывода. Примерами систем, использующих *CBR*, являются: *KATE Tools* (Acknosoft, Франция), *Pattern Recognition Workbench* (Unica, США).

*Нечеткая логика* применяется для обработки данных с размытыми значениями истинности, которые могут быть представлены разнообразными лингвистическими переменными. Нечеткое представление знаний широко применяется в системах с логическими выводами (дедуктивными, индуктивными, абдуктивными) для решения задач классификации и прогно-

зирования, например в системе *XpertRule Miner* (Attar Software Ltd., Великобритания), а также в *AIS* и *NeuFuz* и др.

*Генетические алгоритмы* входят в инструментарий *DM&KDD* как мощное средство решения комбинаторных и оптимизационных задач. Они часто применяются в сочетании с нейронными сетями (см. главу 6). В задачах извлечения знаний применение генетических алгоритмов сопряжено со сложностью оценки статистической значимости полученных решений и с трудностями построения критериев отбора удачных решений. Представителем пакетов из этой категории является *GeneHunter* фирмы Ward Systems Group. Генетические алгоритмы используются также в пакете *XpertRule Miner* и др.

*Логическая (логистическая) регрессия* используется для предсказания вероятности появления того или иного значения дискретной целевой переменной. Дискретная зависимая (целевая) переменная не может быть смоделирована методами обычной многофакторной линейной регрессии. Тем не менее вероятность результата может быть представлена как функция входных переменных, что позволяет получить количественные оценки влияния этих параметров на зависимую переменную. Полученные вероятности могут использоваться и для оценки шансов. Логическая регрессия — это, с одной стороны, инструмент классификации, который используется для предсказания значений категориальных переменных, с другой стороны — регрессионный инструмент, позволяющий оценить степень влияния входных факторов на результат.

*Эволюционное программирование* — самая новая и наиболее перспективная ветвь *DM&KDD*. Суть метода заключается в том, что гипотезы о форме зависимости целевой переменной от других переменных формулируются компьютерной системой в виде программ на определенном внутреннем языке программирования. Если это универсальный язык, то теоретически он способен выразить зависимости произвольной формы. Процесс построения таких программ организован как эволюция в мире программ. Когда система находит программу, достаточно точно выражающую искомую зависимость, она начинает вносить в нее небольшие модификации и отбирает среди построенных дочерних программ те, которые являются наиболее точными. Затем найденные зависимости переводятся с внутреннего языка системы на понятный пользователю язык (математические формулы, таблицы и т.п.). При этом активно используются средства визуализации. Методы эволюционного программирования реализованы в системе *PolyAnalyst* (Unica, США).

В современных средствах *DM&KDD* часто используются комбинированные методы. Например, продукт компании *SAS Enterprise Miner 3.0* содержит модуль автоматического построения результирующей гибридной модели, определенной на множестве моделей, которые предварительно были созданы различными методами: деревьях решений, нейронных сетей, обобщенной многофакторной регрессии. Программная система *Darwin*, разработанная компанией *Thinking Machines*, позволяет не только строить модели на основе нейронных сетей или деревьев решений, но также использовать визуализацию и системы рассуждений по аналогии. Кроме того, этот продукт включает своеобразный генетический алгоритм для оптимизации моделей. Активно работает в области интеллектуального анализа данных компания IBM. Многие из полученных в ее лабораториях результатов нашли применение в выпускаемых инструментальных пакетах, которые можно отнести к четырем из пяти стандартных типов приложений «глубокой переработки» информации: классификации, кластеризации, выявлению последовательностей и ассоциаций.

В настоящее время на рынке представлены разнообразные программные средства, реализующие технологии *DM&KDD*. Следует отметить, что большинство из них имеет очень высокую стоимость. Рассмотрим некоторые известные пакеты. Среди инструментальных средств создания интеллектуальных приложений для бизнеса определенный интерес вызывает семейство программных продуктов *Business Intelligence (BI)* компании Cognos, которое включает четыре взаимосвязанные и дополняющие друг друга системы: *Impromptu*, *PowerPlay*, *Scenario* и *4Thought*. Отличительной особенностью рассматриваемых средств является сочетание эффективности реализуемых в них методов с дружественным ин-

терфейсом, что делает их легкодоступными для освоения непрограммирующими пользователями [48].

Система *Impromptu* обеспечивает доступ к базам данных, позволяя непрофессиональному пользователю формировать разнообразные запросы и отчеты. Система поддерживает работу с распространенными типами СУБД: Oracle, MS SQL Server, Sybase SQL Server, Sybase NetGateway, OmniSQL Gateway, MDI DB2 Gateway, Informix, CA-Ingres, Gupta SQLBase, а также обеспечивает доступ через ODBC. Отчеты *Impromptu* могут использоваться в качестве источников данных для остальных систем семейства BI, выступая при этом в роли информационных витрин.

Система *PowerPlay* является средством для проведения *OLAP*-анализа. Технология *OLAP* позволяет существенно повысить эффективность обработки информации в реляционных БД за счет многомерного представления данных в виде гиперкубов; привязки информации ко времени, дающей возможность анализа динамики данных; реализации сложной вычислительной обработки больших массивов данных [54].

*PowerPlay* обеспечивает многомерный просмотр данных с нисходящим и уровневый анализом, в процессе которого существует возможность выявления исключений и особых случаев, ранжирования и разнообразной обработки данных. Гиперкубы имеют неограниченную размерность и могут создаваться как на серверах, так и на клиентских компьютерах. Новые версии *PowerPlay* обеспечивают возможность работы с гиперкубами через Web-браузеры.

Система *Scenario* предназначена для выявления взаимосвязей в данных статистическими методами, в частности по критерию Хи-квадрат (метод CHAID) для нахождения однородных сегментов данных с «аналогичным» поведением относительно целевого показателя. Кроме того, в системе используется метод деревьев решений для построения классификаций. Результаты обработки данных наглядно отображаются средствами визуализации.

В системе реализованы следующие виды анализа данных:

- *ранжирование* — упорядочение факторов по степени их влияния на целевой показатель. С каждым фактором связывается весовой коэффициент, дающий количественную оценку степени влияния;
- *сегментация* — разделение области значений фактора на сегменты для проведения дальнейшего нисходящего анализа;
- *профилирование лучших образцов* - выявление основных характеристик наиболее успешных результатов (регионов, филиалов, клиентов и т.д.);
- *выявление ассоциаций* — поиск ассоциированных групп значений факторов;
- *выявление исключений* — поиск элементов, выпадающих из общей картины. Появление подобных элементов может быть вызвано как ошибками в данных, которые следует исправить, так и инеобычными ситуациями в работе компании, требующими определенных действий со стороны руководства.

В системе *Scenario* реализованы три стратегии анализа: 1) режим исследования, предназначенный для предварительного анализа задачи; 2) режим тестирования, ориентированный на высокую точность и надежность результатов; 3) режим верификации, позволяющий проводить оценку достоверности и значимости полученных знаний.

Система *4Thought* осуществляет извлечение знаний из БД с применением нейронных сетей, предоставляя следующие возможности:

- моделирование сложных нелинейных зависимостей между факторами и целевыми показателями;
- выявление тенденций в данных (при наличии временных рядов);
- работа с неполными и зашумленными данными при относительно небольшом объеме исходной информации.

Анализируемые данные представляются в виде электронной таблицы, столбцам которой соответствуют атрибуты из таблиц базы данных, а строкам — записи. При подготовке данных для анализа пользователь может редактировать таблицы, а также включать в них вы-

числяемые столбцы. Как и в пакете Scenario, для постановки задачи моделирования должны быть указаны целевой (моделируемый) показатель и переменные-факторы. Данные в таблице рассматриваются системой как примеры для обучения нейронной сети. Совокупность всех данных разбивается на два подмножества — модельное и тестовое. В каждом цикле обучения сеть сначала обучается на модельном подмножестве, а затем проверяется корректность ее работы на тестовых данных. Обучение сети проводится до тех пор, пока точность результатов, полученных на модельном и на тестовом подмножествах, растет. При работе с небольшими объемами данных, когда выбор тестового подмножества существенно влияет на результаты обучения, пользователь может проводить обучение последовательными циклами, выбирая для каждого цикла новое разбиение на тестовые и модельные данные.

Кроме рассмотренных программных продуктов фирмы Cognos можно привести длинный список программных средств, ориентированных на поддержку *DM&KDD*. Особенно активно в последние годы развиваются средства с использованием нейронных сетей, примерами которых являются: *BrainMaker (CSS)*, *NeuroShell (Ward Systems Group)*, *OWL (HyperLogic)*.

Применение технологий *DM&KDD* имеет большие перспективы, так как существенно влияет на увеличение доходов предприятия путем выбора правильных стратегий деятельности. Лидерами в применении технологий интеллектуального анализа данных являются телекоммуникационные компании и компании, выпускающие кредитные карточки. Средства *DM&KDD* активно применяют страховые компании и фондовые биржи. Серьезные успехи связаны с применением этих подходов в медицине, где можно прогнозировать эффективность применения медикаментов, хирургических процедур и медицинских тестов. Прогнозирование в финансовой сфере всегда было одной из самых актуальных задач. В настоящее время компании, действующие на финансовом рынке, на основе *DM*-технологий определяют рыночные и отраслевые характеристики для предсказания индивидуальных и фондовых предпочтений в ближайшем будущем.

Ключом успеха в применении методов *DM&KDD* являются качество данных, мощность используемого программного обеспечения и мастерство аналитика, который участвует в процессе построения модели. На эффективность обработки данных положительно влияют следующие параметры программного обеспечения: развитые средства формирования запросов и визуализации результатов, наличие графического инструментария, возможность оперативной аналитической обработки, разнообразие и эффективность алгоритмов построения моделей,

К типовым задачам *DM&KDD* в области экономики и бизнеса относятся:

- прогнозирование — в частности, при планировании и составлении бюджета фирмы необходимо прогнозировать объемы продаж и другие параметры с учетом многочисленных взаимосвязанных факторов: сезонных, региональных, общеэкономических и т.п.;
- маркетинговый анализ, в результате которого определяется зависимость спроса от таких факторов, как стоимость товара, затраты на продвижение продукции и рекламу и т.д.;
- анализ работы персонала — производительность труда служащих зависит от уровня подготовки, от оплаты труда, опыта работы, взаимоотношений с руководством и т.д. Установив степень влияния этих факторов, можно выработать методику повышения производительности труда, а также предложить оптимальную стратегию подбора кадров в будущем;
- анализ эффективности продажи товаров по почте — выявляется круг потенциальных покупателей, оценивается вероятность возможных покупок и исследуются различные формы рекламной переписки;
- профилирование клиентов — формирование «портрета типичного клиента компании», т.е. определение группы клиентов, сотрудничество с которыми наиболее выгодно. Кроме того, важно выяснить, почему работа с некоторыми из заказчиков стала неэффективной, и выработать стратегию поиска подходящих клиентов в будущем;

- оценка потенциальных клиентов — выявление характерных особенностей заявок, которые закончились реальными продажами. Полученные знания используются в процессах планирования переговоров и сделок;

- анализ работы региональных отделений компании;
- сравнительный анализ конкурирующих фирм.

Перечисленные задачи актуальны практически для всех отраслей бизнеса: банковского дела, страхования, финансовых рынков, производства, торговли и т.д.

### **ПРОГРАММНЫЕ ИНСТРУМЕНТАЛЬНЫЕ СРЕДСТВА ИАС**

Состав программных инструментальных средств ИАС

В предыдущих разделах были рассмотрены основные функции информационно-аналитических систем, в том числе: информационного хранилища ИХ ( DW ), оперативного анализа ( OLAP ), интеллектуального анализа ( DMg ), представления пользователю результатов анализа и подготовки принятия решений ( DMt ). Для реализации функций применяется набор программных инструментальных средств. Полный набор этих средств включает ряд крупных модулей.

Крупные функциональные модули могут быть органической частью ИАС или самостоятельным программным продуктом. Во втором варианте они входят в ИАС в качестве комплектующего элемента. В свою очередь эти модули состоят из ряда блоков. Перечислим основные блоки:

- средства импорта, перекачки данных из операционных баз и других источников информации, взаимодействующие с различными операционными системами и СУБД;

- средства преобразования данных, осуществляющие проверку на правильность, преобразование структур, агрегирование;

- набор или комплекс программ, которые выполняют операционные функции оперативного ( OLAP ) анализа; основу их составляет язык запросов Structured Query Language ( SQL ) усеченного или расширенного типа, в развитых ИАС в комплект входят специализированные языки различного уровня;

- средства графического и визуального конструирования отчетов, рассчитанные на конечного пользователя, как правило, дублируются языковыми средствами;

- средства удаленного доступа, обеспечения работы в распределенном и режиме "клиент-сервер", коллективного доступа и работы в глобальных сетях;

- средства администрирования ИАС;

- средства интеллектуального анализа данных, обычно называемые " Miner ";

- приложения, разработанные встроенными в комплекс программ средствами.

- средства моделирования объектов и процессов.

Структура программных средств ИАС изображена на рис. 8.1.

Рассмотрим подробнее назначение и функции программных модулей, входящих в состав ИАС. Многие из них имеют самостоятельное значение и могут быть использованы для различных целей.

## **МЕТОДИЧЕСКИЕ УКАЗАНИЯ К ПРАКТИЧЕСКИМ ЗАНЯТИЯМ**

### **Практическая работа № 1**

#### ***Представление знаний и получение выводов***

#### ***С помощью логики предикатов***

*Предикатом* [2], или *логической функцией*, называется функция от любого числа аргументов, принимающая истинностные значения: истинно (1) и ложно (0). Аргументы принимают значения из произвольного, конечного или бесконечного множества D, называемого предметной областью. Предикат от n аргументов называют n-местным предикатом.

Предикат F(x), определенный на множестве D, задает определенное свойство элементам множества D и интерпретируется как высказывание "x обладает свойством F", причем F принимает значение "истинно", если это высказывание истинно, и значение "ложно",

если оно ложно. Предикат  $F(x_1, x_2, \dots, x_n)$  задает отношение между элементами  $x_1, x_2, \dots, x_n$  и интерпретирует как высказывание “ $x_1, x_2, \dots, x_n$  находятся между собой в отношении  $F$ ”.

Например:  $D$  – множество натуральных чисел.  $F(x)$  может означать, что  $x$  – четное или  $x$  – нечетное.  $G(x, y)$  –  $x > y$  или  $x$  делится на  $y$ .

Алфавит языка предикатов первого порядка состоит из:

- 1) разделители: запятая, открывающая и закрывающая скобки;
- 2) константы, обозначаемые строчными буквами или соединением таких букв, – например:  $a, st$ , друг;
- 3) переменные, обозначаемые прописными буквами, – например:  $X, ST, АДРЕС$ ;
- 4) предикаты, обозначаемые прописными буквами, – например:  $P, Q$ , БОЛЬШЕ;

5) функции, устанавливающие зависимость и отображающие значения одной предметной области в значения другой (или той же),  $n$ -местные функции могут служить аргументами предиката. Функции будем обозначать строчными буквами:  $f, g, t$ ;

б) логические функции:

$\neg$  (отрицание или дополнение). Высказывание “ $\neg A$ ” читается “не  $A$ ”. Оно истинно (И), если высказывание  $A$  – ложно (Л);

$\wedge$  (конъюнкция). Высказывание “ $A \wedge B$ ” читается “ $A$  и  $B$ ”. Оно истинно в том случае, когда истинно как  $A$ , так и  $B$ ;

$\vee$  (дизъюнкция). Высказывание “ $A \vee B$ ” читается “ $A$  или  $B$ ”. Оно истинно, если истинно хотя бы одно из высказываний;

$\rightarrow$  (импликация). Высказывание “ $A \rightarrow B$ ” читается “если  $A$ , то  $B$ ”. Оно ложно в том и только в том случае, если  $A$  истинно, а  $B$  ложно;

$\leftrightarrow$  (эквивалентность). Высказывание “ $A \leftrightarrow B$ ” читается “ $A$  тогда и только тогда, когда  $B$ ”. Оно истинно в тогда и только тогда, когда  $A$  и  $B$  имеют одно и тоже истинностное значение;

$\exists$  (квантор существования). Высказывание “ $\exists A$ ” читается “существует  $A$ ”;

$\forall$  (квантор общности). Высказывание “ $\forall A$ ” читается “для любого  $A$ ”.

*Пропозициональной формой* [2], или *формулой алгебры логики*, называют всякое высказывание, составленное из некоторых исходных высказываний посредством логических операций, т. е. если  $F$  и  $G$  – пропозициональные формы, то  $\neg F$ ,  $(F \vee G)$ ,  $(F \wedge G)$ ,  $(F \rightarrow G)$ ,  $(F \leftrightarrow G)$  – пропозициональные формы.

*Пропозициональной функцией* [5] называется выражение, содержащее переменную и превращающееся в истинное или ложное высказывание при подстановке вместо переменной имени предмета из определенной предметной области.

Пропозициональные функции делятся на одноместные, содержащие одну переменную, называемые свойствами (например, « $x$  – композитор», « $x-7=3$ »), и содержащие две и более переменных, называемые отношениями (например, « $x-c=16$ », «объем куба  $x$  равен объему куба  $y$ »).

Определение формулы – основного объекта логики предикатов – включает понятие “терм”.

*Терм* – выражение, включающее константы, переменные или  $n$ -местные функции  $f(t_1, t_2, \dots, t_n)$ , где  $t_1, t_2, \dots, t_n$  – термы.

Например:  $f(X, Y)$ ,  $f(b, \text{вес}(Z))$  – термы,  $P(X, \text{голубой})$ ,  $\text{вес}(P(b))$  – не термы, т.к.  $P$  – предикат.

*Атом*, или *элементарная (атомная) формула* – это выражение, включающее константы, переменные, функции и предикаты. Таким образом, если  $P$  –  $n$ -местный предикат, а  $t_1, t_2, \dots, t_n$  – термы, то  $P(t_1, t_2, \dots, t_n)$  – атом.

Например:  $P(X, \text{голубой})$ ,  $\text{ВХОД}(\text{стол}, X, \text{под}(\text{окно}))$  – являются атомами,  $f(X, Y)$  – не атом.

Формула или правильно построенная формула (ППФ) определяется следующим образом: всякий атом есть ППФ.

Если  $F$  и  $G$  – ППФ, а  $X$  – переменная, тогда  $\neg H$ ,  $(G \vee H)$ ,  $(G \wedge H)$ ,  $(\exists X)G$ ,  $(\forall X)H$  – ППФ.

Выражение “первого порядка” во фразе “исчисление предикатов первого порядка” связано с определением ППФ, в которых запрещается квантифицировать символы предикатов и функций.

Например:

$(\forall P)P(a)$   
 $(\forall f)(\forall X)P(f(X), b)$  – не являются ППФ логики предикатов первого порядка.

На практике ППФ используется для представления знаний. Не всегда легко представить знания, выраженные на естественном языке, с помощью ППФ. Например, выражение “если два объекта равны, то они имеют одинаковые свойства” можно представить так;

$(\forall P)(\forall X)(\forall Y)(PABHO(X, Y) \rightarrow (P(X) \leftrightarrow P(Y)))$ .

Но это выражение не является формулой первого порядка, так как квантифицируется предикат.

Покажем, каким образом можно выявлять логическую структуру мысли. Приведем несколько сложных суждений, структуру которых надо выразить в виде формул, используя введенные логические термины.

1. Если у меня будет свободное время ( $a$ ) и я сдам экзамены по математике ( $b$ ) и физике ( $c$ ), то поеду отдыхать в Крым ( $d$ ) или на Кавказ ( $e$ ).

Формула:  $(a \wedge b \wedge c) \rightarrow (d \vee e)$ .

2. «Если человек с детства и юности своей не давал нервам властвовать над собой, то они не привыкнут раздражаться и будут ему послушны» (К.Д.Ушинский).

Формула:  $(\neg a \wedge \neg b) \rightarrow (\neg c \wedge d)$ .

Здесь буква  $a$  обозначает суждение: «человек с детства давал нервам властвовать над собой». А так как у нас имеется отрицание («не давал»), то запишем  $\neg a$ .

3. Если добродетель неправильно приложат, то она может стать пороком.

Формула:  $a \rightarrow b$ .

4. Если у меня будет свободное время, то я прочитаю книгу или посмотрю телевизор.

Формула:  $a \rightarrow (b \vee c)$ .

5. Некоторые элементарные частицы имеют положительный заряд.

Формула:  $(\exists X)(S(X) \wedge P(X))$ .

*Правилом вывода* называют процедуру, которая из одной или нескольких ППФ производит другие ППФ. Задавая фиксированное множество правил вывода, можно рассматривать следующее семейство проблем: исходя из выбранного множества ППФ применением некоторого числа раз правил вывода, можно получить заранее заданную ППФ.

Исходные ППФ называют *аксиомами*, а ППФ, полученные из правил вывода, называют *теоремами*. Цель применения правил вывода, ведущих от аксиомы к теореме, называют *доказательством теоремы*.

*Интерпретация формул.* Формула имеет определенный смысл, т. е. обозначает некоторое высказывание, если существует какая-либо интерпретация. Интерпретировать формулу – это значит связать с ней определенное непустое множество  $D$ , т. е. конкретизировать предметную область, называемую также *областью интерпретации*, и указать:

для каждой константы в формуле – конкретный элемент из  $D$ ;

для каждой  $n$ -местной функциональной буквы в формуле – конкретную  $n$ -местную функцию на  $D$ ;

для каждой  $n$ -местной предикатной буквы в формуле – конкретное отношение между  $n$  элементами из  $D$ .

*Свойства правильно построенных формул.* При заданной интерпретации значения истинности правильно построенную формулу (ППФ) можно вычислить по правилам, объединенным в таблице истинности.

Если F и G – любые две ППФ, то значения истинности составного выражения, построенного из этих ППФ, даются таблицей истинности (табл. 1).

Таблица 1. Таблица истинности

F	G	$\neg F$	$F \vee G$	$F \wedge G$	$F \rightarrow G$	$F \leftrightarrow G$
И	И	Л	И	И	И	И
Л	И	И	И	Л	И	Л
И	Л	Л	И	Л	Л	Л
Л	Л	И	Л	Л	И	И

Если истинности двух ППФ независимо от интерпретации совпадают, то говорят, что эти ППФ являются эквивалентными. Пользуясь таблицей истинности, легко установить следующие эквивалентности (табл. 2.).

### КОНТРОЛЬНЫЕ ВОПРОСЫ

1. Что называется предикатом?
2. Из чего состоит алфавит языка предикатов первого порядка?
3. Что называют формулой алгебры логики?
4. Что такое элементарная формула?
5. Какую процедуру называют правилом вывода?
6. Что значит интерпретировать правильно построенную формулу?

Таблица 2. Таблица эквивалентности

Эквивалентные формулы		
$\neg(\neg F)$	F	
$F \rightarrow G$	$\neg F \vee G$	
$F \leftrightarrow G$	$(F \rightarrow G) \wedge (G \rightarrow F)$	
$\neg(F \wedge G)$	$\neg F \vee \neg G$	Законы де Моргана
$\neg(F \vee G)$	$\neg F \wedge \neg G$	
$F \wedge (G \vee H)$	$(F \wedge G) \vee (F \wedge H)$	Законы дистрибутивности
$F \vee (G \wedge H)$	$(F \vee G) \wedge (F \vee H)$	
$F \vee G$	$G \vee F$	Законы коммутативности
$F \wedge G$	$G \wedge F$	
$(F \wedge G) \wedge H$	$F \wedge (G \wedge H)$	Законы ассоциативности
$(F \vee G) \vee H$	$F \vee (G \vee H)$	
$F \rightarrow G$	$\neg G \rightarrow \neg F$	
$F \wedge \neg F$	0	
$F \vee \neg F$	1	
$F \wedge 0$	0	
$F \wedge 1$	F	
$F \vee 0$	F	
$F \vee 1$	1	
$\forall X F(X)$ $\exists X F(X)$	$\forall Y F(Y)$ $\exists Y F(Y)$	

$\neg(\exists X F(X))$	$\forall X(\neg F(X))$
$\neg(\forall X F(X))$	$\exists X(\neg F(X))$
$\forall X(F(X) \wedge G(X))$	$\forall X F(X) \wedge \forall X G(X)$
$\exists X(F(X) \vee G(X))$	$\exists X F(X) \vee \exists X G(X)$

### ЗАДАНИЕ

Выразить в символической форме следующие сложные суждения:

1. Если встать рано на рассвете и пойти в сад или парк, то можно услышать чудесные песни птиц.
2. Если эта фигура квадрат, то диагонали в ней равны, взаимно перпендикулярны и в точке пересечения делятся пополам.
3. Некоторые спортсмены не являются мастерами спорта.
4. «Некоторые лекарства опаснее самих болезней» (Сенека).
5. Некоторые люди не изучают логику.
6. Если мне дадут отпуск летом, то я поеду отдыхать к морю или по туристической путевке в Канаду.
7. Добро не умрет, а зло пропадет.
8. «Видеть несправедливость и молчать – это значит самому участвовать в ней» (Ж.-Ж. Руссо).
9. Если вы любите детей, полны жажды познания, имеете доброе сердце, мечтаете посвятить себя интересному творческому труду, то смело выбирайте профессию учителя.
10. «Если больному после разговора с врачом не становится легче, то это не врач» (В. М. Бехтерев).
11. «Если верный конь, поранив ногу, вдруг споткнулся, а потом опять, не вини коня, вини дорогу и коня не торопись менять» (Р. Гамзатов).

### Практическая работа № 2

#### *Преобразование правильно построенных формул в предложение*

Аппарат логики предикатов используется для представления задачи в виде теоремы: формула  $H$  логически следует из формулы  $G$ , т.е.  $G \rightarrow H$ . Доказательство этой теоремы состоит в том, чтобы показать, что каждая интерпретация, удовлетворяющая  $G$ , удовлетворяет и  $H$ , или, что  $\neg G \vee H$  истинно для любой интерпретации; так как  $\neg(\neg G \vee H) = G \wedge \neg H$ , то на практике обычно доказывают невыполнимость множества предложений  $G \wedge \neg H$ .

Чтобы упростить доказательства теоремы, все формулы представляются в виде дизъюнкций литералов. *Литералом* (или *литерой*) называют атом или его отрицание. Формулу, представляющую собой дизъюнкцию литералов, называют *предложением* (или *дизъюнктом*).

Например:  $R(Z, a, g(X)) \vee (\neg T(U)) \vee (\neg U(b, k(c)))$ .

Иными словами, в каждой формуле необходимо исключить все логические операции (включая кванторы), кроме дизъюнкции, и уменьшить область действия знака отрицания до одной предикатной буквы.

*Преобразование ППФ в предложение.* Перед тем как объяснить процесс резолюции, покажем, что любую ППФ исчисления предикатов первого порядка можно преобразовать во множество предложений, используя законы эквивалентности. Процесс такого преобразования состоит из последовательных этапов, которые покажем на следующем примере ППФ:

$(\forall X)((P(X) \wedge Q(X, a)) \rightarrow R(X, b)) \wedge ((\forall Y)((\forall Z)(R(Y, Z) \rightarrow T(X, Y)))) \vee ((\forall X)S(X))$ .

*Исключение символов эквивалентности и импликации.* Для этого воспользуемся следующими законами эквивалентности:  $(F \rightarrow G)$  заменим на  $(\neg F \vee G)$ . В нашем примере такая подстановка даст:

$$(\forall X)((\neg(P(X)\wedge Q(X,a))\vee R(X,b))\wedge((\forall Y)(\neg(\forall Z)(R(Y,Z)\vee T(X,Y))))\vee((\forall X)S(X))).$$

*Уменьшение области действия знаков отрицания.* Нужно выполнить такое преобразование, чтобы каждый знак отрицания применялся не более чем к одной атомной формуле. Для этого используем следующие законы эквивалентности:

$$\begin{aligned} &(\neg(\neg F)) \text{ и } F \\ &(\neg(F\wedge G)) \text{ и } \neg F\vee(\neg G) \\ &(\neg(F\vee G)) \text{ и } \neg F\wedge(\neg G) \\ &(\neg(\exists X)F(X)) \text{ и } (\forall X)(\neg F(X)) \\ &(\neg(\forall X)F(X)) \text{ и } (\exists X)(\neg F(X)) \end{aligned}$$

Полученная нами ППФ преобразуется к виду:

$$(\forall X)((\neg P(X)\vee\neg Q(X,a)\vee R(X,b))\wedge((\forall Y)((\exists Z)(\neg R(Y,Z)\vee T(X,Y))))\vee((\forall X)S(X)))$$

*Разделение переменных.* В пределах области действия квантора, переменная, связываемая с этим квантором, представляет собой немую переменную. Такую переменную в области действия квантора можно заменить любой другой (не встречающейся) переменной, при этом значение истинности ППФ не изменится. Стандартизация переменных в пределах ППФ означает переименование немых переменных с той целью, чтобы каждый квантор имел свою, свойственную только ему, немую переменную. Так, вместо записи

$$(\forall X)(P(X)\vee(\exists X)Q(X))$$

можно написать

$$(\forall X)(P(X)\vee(\exists Y)Q(Y)).$$

*Полученная на предыдущем этапе ППФ преобразуется к виду:*

$$(\forall X)((\neg P(X)\vee\neg Q(X,a)\vee R(X,b))\wedge((\forall Y)((\exists Z)(\neg R(Y,Z)\vee T(X,Y))))\vee((\forall U)S(U)))$$

*Исключение кванторов существования.* Рассмотрим ППФ

$$(\forall X)(\exists Y)P(X,Y),$$

которую можно прочитать как: "Для всех  $X$  существует некоторое  $Y$  (возможно, зависящее от  $X$ ) такое, что  $P(X,Y)$ ". Поскольку квантор существования находится в области некоторого квантора общности, то можно допустить, что "существующий"  $Y$  зависит от  $X$ . Пусть эта зависимость определяется функцией  $Y=g(X)$ , отображающей каждое  $X$  в  $Y$ . Такая функция называется *сколемовской*, или *функцией Сколема*. Заменяв  $Y$  на  $g(X)$ , мы исключим квантор существования, и наша ППФ примет вид:

$$(\forall X)(P(X,q(X))).$$

Пусть ППФ  $(\exists Y)P(Y)$  является подформулой другой ППФ, в которой  $X_1, \dots, X_n$  квантифицированы универсально. Тогда удаляют  $Y$  и заменяют встречающуюся  $Y$  сколемовской функцией  $f(X_1, \dots, X_n)$ . Заметим, что эта функция будет содержать столько аргументов, сколько имеется универсальных квантификаторов слева от формулы  $(\exists Y)P(Y)$ . Эта функция, как мы уже говорили, просто выражает существование соответствия априори между совокупностью значений  $(\exists Y)P(Y)$  и значениями  $Y$ . Поскольку эта функция заранее неизвестна, то в каждой замене  $Y$  мы должны записывать новую функциональную букву, которая отличается от уже имеющихся. Когда слева от символа " $\exists$ " отсутствуют символы " $\forall$ ", то вводимая функция Сколема не будет содержать аргументов: следовательно, это будет новая константа, называемая константой Сколема. Так, ППФ  $(\exists Y)P(Y)$  заменится на  $P(a)$ , где про константу " $a$ " известно, что она существует.

После выполнения замены  $Z$  функцией Сколема  $g(X,Y)$  наша ППФ будет иметь вид

$$(\forall X)((\neg P(X)\vee\neg Q(X,a)\vee R(X,b))\wedge((\forall Y)(\neg R(Y,g(X,Y))\vee T(X,Y))))\vee((\forall U)S(U)).$$

*Перемещение всех квантификаторов общности в начало ППФ* (без изменения их относительного порядка). Каждый квантор общности имеет свою переменную, и поэтому такие кванторы можно переместить в начало формулы, считая, что область действия каждого из них включает всю последующую часть ППФ. Говорят, что результирующая ППФ находится в префиксной форме. Она состоит из цепочки кванторов, называемой *префиксом*, и следующей за ней бескванторной формулы, называемой *матрицей*.

Префиксная форма для нашей ППФ имеет вид:

$$(\forall X)(\forall Y)(\forall U)((\neg P(X) \vee \neg Q(X,a) \vee R(X,b)) \wedge (\neg R(Y,Z) \vee T(X,Y))) \vee S(U)).$$

*Приведение матрицы к конъюнктивной нормальной форме.* Матрица находится в конъюнктивной нормальной форме, если она записана как конъюнкция конечного множества дизъюнкций литералов. Например:

$$(P(X) \vee Q(X,Y)) \wedge (P(Y) \vee \neg R(Y))$$

$$Q(X) \wedge R(X,Y)$$

$$\neg R(Y)$$

Любую матрицу можно привести к конъюнктивной нормальной форме, используя законы ассоциативности и дистрибутивности операций  $\vee$  и  $\wedge$ . После приведения матрицы нашего примера ППФ в конъюнктивную нормальную форму эта ППФ примет следующий вид:

$$(\forall X)(\forall Y)(\forall U)(\neg P(X) \vee \neg Q(X,a) \vee (R(X,b) \vee S(U))) \wedge$$

$$(\neg R(Y,g(X,Y)) \vee T(X,Y) \vee S(U)).$$

*Исключение кванторов общности.* Поскольку все переменные в ППФ должны быть связаны, то можно предположить, что для каждой из них имеется квантор общности. При этом порядок расположения этих кванторов роли не играет, так что можно их исключить из ППФ. После исключения кванторов из ППФ у нас остается лишь матрица в конъюнктивной нормальной форме.

*Исключение символов конъюнкции.* Теперь можно исключить символы  $\wedge$ , путем замены выражения типа  $(P \wedge Q)$  на множество ППФ  $\{P, Q\}$ . В результате выполнения таких замен получим конечное множество ППФ, каждая из которых – дизъюнкция литералов. А любая ППФ, содержащая дизъюнкцию литералов, как мы условились, является предложением.

Наш пример ППФ преобразуется в следующее множество предложений:

$$\{\neg P(X) \vee \neg Q(X,a) \vee (R(X,b) \vee S(U))\}$$

$$\{\neg R(Y,g(X,Y)) \vee T(X,Y) \vee S(U)\}.$$

*Переименование (разделение) переменных.* Символы переменных должны быть изменены так, чтобы каждый появился не более чем в одном предложении. Напомним, что  $(\forall X(P(X) \wedge Q(X)))$  эквивалентно  $(\forall X)P(X) \wedge (\forall Y)Q(Y)$ . Теперь наши предложения приобретают вид:

$$\neg P(X) \vee \neg Q(X,a) \vee (R(X,b) \vee S(U))$$

$$\neg R(Y,g(Z,Y)) \vee T(A,B) \vee S(C)$$

Вместо переменных в литералах могут находиться термы, не содержащие переменных. Тогда говорят, что имеет место основной частный случай. Например,  $P(a,f(g(b)))$  является основным частным случаем  $P(X,Y)$ .

По структуре высказывания делятся на простые (они имеют логическую форму «S есть P» либо «S не есть P») и сложные (грамматически выражаются сложными предложениями). Простые высказывания бывают

атрибутивные (в них выражается принадлежность или непринадлежность какого-то свойства объекту или классу объектов);

об отношениях между несколькими объектами;

о существовании или несуществовании какого-либо объекта или явления).

В атрибутивные высказывания часто включаются кванторные связки.

По качеству простые высказывания делятся на утвердительные и отрицательные. С количественной точки зрения высказывания делятся на единичные, частные и общие.

Перечислим основные типы высказываний и их обозначения [5]:

**A** – общеутвердительные («Всякий S есть P»),

**E** – общеотрицательные («Всякий S не есть P»),

**I** – частноутвердительные («Некоторый S есть P»),

**O** – частноотрицательные («Некоторый S не есть P»).

В частных суждениях слово «некоторые» не исключает варианта «все».

### **Пример.**

Преобразовать высказывание "Все дети имеют матерей" в предложение, т.е. в дизъюнкцию литералов. Это высказывание можно представить следующей ППФ:

$$\forall X \exists Y (\text{ребенок}(X) \rightarrow \text{мать}(Y) \wedge \text{относится}(X, Y)).$$

Для удаления квантора существования заменим переменную  $Y$  функцией Сколема  $g(A)$ , т.е.  $Y=g(X)$ , откуда получим:

$$\forall X (\text{ребенок}(X) \rightarrow \text{мать}(g(X)) \wedge \text{относится}(X, g(X))).$$

Теперь можно исключить квантор общности, который распространяет свое действие на все выражения. Кроме того, заменим символ "импликация" дизъюнкцией, используя закон  $P \rightarrow Q \equiv \neg P \vee Q$ , откуда получаем:

$$\neg \text{ребенок}(X) \vee (\text{мать}(g(X)) \wedge \text{относится}(X, g(X))).$$

Используя закон

$$P \vee (Q \wedge R) \equiv (P \vee Q) \wedge (P \vee R),$$

можем написать

$$\neg \text{ребенок}(X) \vee \text{мать}(g(X)),$$

$$\neg \text{ребенок} \vee \text{относится}(X, g(X)).$$

Два полученные предложения можно интерпретировать так: "Если  $X$  –ребенок, то он относится к матери, выражаемой функцией  $g(X)$ ". Из них следует:

$\text{ребенок}(X) \rightarrow \text{мать}(g(X))$  и

$\text{ребенок}(X) \vee \text{относится}(X, g(X))$ .

### **КОНТРОЛЬНЫЕ ВОПРОСЫ**

1. Дайте определение литерала.
2. Что называют дизъюнктом?
3. Какая функция называется функцией Сколема?
4. Как любую ППФ можно преобразовать во множество предложений?
5. Когда ППФ находится в префиксной форме?
6. Какая ППФ называется матрицей?
7. Когда матрица находится в конъюнктивной нормальной форме?

### **ЗАДАНИЕ**

Преобразовать высказывания в дизъюнкцию литералов.

1. Некоторые растения являются грибами.
2. Все предложения со второстепенными членами являются распространенными.
3. Все одноклеточные не являются червяками.
4. Некоторые космонавты – летчики.
5. Некоторые студенты – не лодыри.
6. Все дельфины – теплокровные.
7. Ни один кит не является рыбой.
8. Все свидетели дают истинные показания.
9. Ни одна балалайка не является клавишным инструментом.
10. Некоторые люди не любят природу.

### **Практическая работа № 3**

#### ***Сущность принципа резолюций.***

#### ***Примеры использования принципа резолюций***

*Резольвенты* [2]. Говорят, что литерал является конкретным, если он не содержит никакой переменной. Например,  $\neg P(a)$  или  $Q(a, f(b))$  являются конкретными литералами, в то время как  $\neg P(X)$  или  $Q(X, f(y))$  не являются таковыми. Конкретным предложением является дизъюнкция конкретных литералов.

Пусть имеются два конкретных предложения

$$P_1 \vee P_2 \vee \dots \vee P_n \text{ и } \neg P_1 \vee Q_2 \vee \dots \vee Q_n$$

Здесь литерал  $\neg P_1$  является отрицанием литерала  $P_1$ . Из этих двух предложений можно вывести новое предложение, называемое *резольвентой*, или *резольвцией*. Резольвентой является дизъюнкция этих предложений с последующим исключением пары  $P_1$  и  $\neg P_1$ .

Из предложений  $P_1$  и  $\neg P_1$  получается *пустое предложение*, которое является признаком противоречия и обозначается символом  $\square$ .

Два конкретных предложения могут не иметь резольвенты или могут иметь множество резольвент. Например, из  $P \vee Q \vee R$  и  $\neg P \vee \neg Q \vee S$  получаются резольвенты  $Q \vee \neg Q \vee R \vee S$  или  $P \vee \neg P \vee R \vee S$  (которые в действительности являются эквивалентными).

*Сущность принципа резолюций.* На практике невыполнимость множества предложений устанавливается посредством принципа резолюций. Речь идет о процедуре логического вывода новых предложений из множества исходных.

Принцип резолюций состоит:

1) в получении новых предложений на основании множества исходных и вновь получаемых предложений;

2) в отыскании частных случаев формул  $F(t_1, \dots, t_n)$  из  $F(V_1, \dots, V_n)$  при подстановке вместо  $V_i$  произвольных термов  $t_i$ , т.е.  $F(V_1, \dots, V_n) \rightarrow F(t_1, \dots, t_n)$ .

Резольвенту двух предложений можно получить следующим образом:

1) переименовать переменные двух предложений так, чтобы последние стали одинаковыми;

2) найти подстановку, преобразующую литерал одного предложения в дополнительный по отношению к некоторому литералу другого предложения и произвести эту замену в обоих предложениях;

3) вычеркнуть дополнительные друг к другу литералы;

4) удалить одинаковые литералы в предложении кроме одного;

5) дизъюнкция литералов, оставшихся в обоих предложениях, является резольвентой.

Если некоторая последовательность резолюций, применяемых к исходному множеству предложений  $E$  и множеству резольвент, полученных в процессе резолюции, приводит к пустому предложению, то множество  $E$  является *невыполнимым*.

Для доказательства невыполнимости множества предложений пользуются *опровержением*. Доказательство выполнимости множества  $G \rightarrow H$ , что эквивалентно  $\neg G \vee H$ , – это опровержение его невыполнимости, или, что то же самое, доказательство невыполнимости  $\neg(\neg G \vee H)$ , что эквивалентно  $G \wedge \neg H$ .

Общий алгоритм опровержения с помощью резолюций может иметь следующий

вид:

**резолюция** ( $G, H$ )

1)  $S$ : – множество предложений, полученных путем преобразования формул множества  $G$ ;

2) добавить к множеству  $S$  предложения, полученные из  $\neg H$ ;

3) пока пустое предложение не появится в  $S$ , **выполнить:**

**начало:** выбрать два различных предложения в  $S$ ;

**если** они имеют резольвенты, то найти одну резольвенту и добавить ее к множеству

$S$ .

**конец.**

*Примеры использования метода резолюций.*

**Пример 1.** Предположим, что покупательная способность людей падает, если цены растут на товары и услуги. Предположим также, что люди несчастны, когда их покупательная способность падает. Предположим, что цены растут. Из этого можно заключить, что люди несчастны.

Применим следующие обозначения:

1)  $P$  – цены растут;

2)  $S$  – покупательная способность уменьшается;

3)  $U$  – люди несчастны.

В этом примере четыре утверждения:

- 1) если цены растут, то покупательная способность падает;
- 2) если покупательная способность падает, то люди несчастны;
- 3) цены растут;
- 4) люди несчастны.

Эти утверждения можно записать в виде следующих формул:

- 1)  $P \rightarrow S$ ,
- 2)  $S \rightarrow U$ ,
- 3)  $P$ ,
- 4)  $U$ .

Преобразуем эти формулы в предложения:

- 1)  $\neg P \vee S$ ,
- 2)  $\neg S \vee U$ ,
- 3)  $P$ ,
- 4)  $U$ .

Докажем путем опровержения, что  $U$  – логическое следствие из 1), 2), 3). Отрицаем 4) и получаем следующее доказательство:

- 1)  $\neg P \vee S$ ,
- 2)  $\neg S \vee U$ ,
- 3)  $P$ ,
- 4)  $\neg U$  отрицание заключения,
- 5)  $S$  резольвента 3) и 1),
- 6)  $U$  резольвента 5) и 2),
- 7)  $\square$  резольвента 6) и 4).

Пример 2. Все доктора имеют пациентов и некоторые пациенты любят своих докторов. Ни один пациент не любит знахаря. Следовательно, никакой доктор не является знахарем.

Обозначим:

- 1)  $P(X)$ :  $X$  – пациент;
- 2)  $D(X)$ :  $X$  – доктор;
- 3)  $Q(X)$ :  $X$  – знахарь;
- 4)  $L(X, Y)$ :  $X$  любит  $Y$ .

Тогда факты и заключения можно записать следующим образом:

- $F1: (\exists X)(P(X) \wedge (\forall Y)(D(Y) \rightarrow L(X, Y)))$ ,  
 $F2: (\forall X)(P(X) \rightarrow (\forall Y)(Q(Y) \rightarrow \neg L(X, Y)))$ ,  
 $G: (\forall X)(D(X) \rightarrow \neg Q(X))$ .

Преобразуем эти формулы в предложения:

- 1)  $P(a)$  из  $F1$ ,
- 2)  $\neg D(Y) \vee L(a, Y)$  из  $F1$ ,
- 3)  $\neg P(X) \vee \neg Q(Y) \vee \neg L(X, Y)$  из  $F2$ ,
- 4)  $D(b)$  из  $G$ ,
- 5)  $Q(b)$  из  $G$ .

Используя метод резолюций, получим следующее доказательство:

- 6)  $L(a, b)$  резольвента 4) и 2),
- 7)  $\neg Q(Y) \vee \neg L(a, Y)$  резольвента 3) и 1),
- 8)  $\neg L(a, b)$  резольвента 5) и 7),
- 9)  $\square$  резольвента 6) и 8).

Поясним естественность этого доказательства.

Из  $F1$  можно предположить, что существует пациент  $a$ , который любит каждого доктора 1) и 2).

Предположим, что заключение неверно, т.е.  $b$  – одновременно и доктор, и знахарь.  
 Так как пациент любит каждого доктора, то  $a$  любит  $b$  6).  
 Так как  $a$  – пациент, то  $a$  не любит никакого знахаря 7).  
 Однако  $b$  – знахарь. Следовательно,  $a$  не любит  $b$  8).

Это невозможно из-за 6). Таким образом, мы закончили доказательство.

Пример 3. Допустим, что если Верховный Совет отказывается принимать новые законы, то забастовка не будет закончена, если только она не длится более года и зарплата не повышается. Закончится ли забастовка, если Верховный Совет отказывается действовать и забастовка только началась?

Применим следующие обозначения:

- 1)  $P$  – Верховный Совет отказывается действовать;
- 2)  $Q$  – забастовка заканчивается;
- 3)  $R$  – зарплата повышается;
- 4)  $S$  – забастовка длится более года.

Тогда приведенные выше утверждения можно представить следующими формулами:

- 1)  $F1: (P \rightarrow (\neg Q \vee (R \wedge S)))$  – если Верховный Совет отказывается принимать новые законы, то забастовка не будет закончена, если она не длится более года и зарплата не повышается;
- 2)  $F2: P$  – Верховный Совет отказывается действовать;
- 3)  $F3: \neg S$  – забастовка только началась.

Требуется доказать, что  $\neg Q$  – логическое следствие  $F1 \wedge F2 \wedge F3$ . Отрицаем  $Q$  и преобразуем  $F1, F2$  и  $F3$  в предложения:

- 1)  $\neg P \vee \neg Q \vee R$  из  $F1$ ,
- 2)  $\neg P \vee \neg Q \vee S$  из  $F1$ ,
- 3)  $P$  из  $F2$ ,
- 4)  $\neg S$  из  $F3$ ,
- 5)  $Q$  отрицание заключения.

Используя резолюции, получим следующее доказательство:

- 6)  $\neg Q \vee S$  резольвента из 3) и 2),
- 7)  $S$  резольвента из 6) и 5),
- 8)  $\square$  резольвента из 7) и 4).

Пример 4. Таможенники обыскивают каждого, кто въезжает в страну, кроме высокопоставленных лиц. Некоторые люди, способствующие провозу наркотиков, въезжали в страну и были обысканы людьми, также способствующими провозу наркотиков. Никто из высокопоставленных лиц не способствовал провозу наркотиков. Доказать, что некоторые из таможенников способствовали провозу наркотиков.

Примем следующие обозначения:

- 1)  $E(X)$ :  $X$  въезжал в страну;
- 2)  $V(X)$ :  $X$  – высокопоставленное лицо;
- 3)  $S(X, Y)$ :  $Y$  обыскивает  $X$ ;
- 4)  $C(X)$ :  $X$  – таможенник;
- 5)  $P(X)$ :  $X$  способствует провозу наркотиков.

Заключение представляется формулой:

$$(\exists X)(P(X) \wedge C(X)).$$

Отрицаем заключение и преобразуем формулы в предложения:

- 1)  $\neg E(X) \vee V(X) \vee S(X, f(X))$ ,
- 2)  $\neg E(X) \vee V(X) \vee C(f(X))$ ,
- 3)  $P(a)$ ,
- 4)  $E(a)$ ,
- 5)  $\neg S(a, Y) \vee P(Y)$ ,

6)  $\neg P(X) \vee \neg V(X)$ ,

7)  $\neg P(X) \vee \neg C(X)$ .

Доказательство методом резолюций выглядит следующим образом:

8)  $\neg V(a)$  из 3) и 6),

9)  $V(a) \vee C(f(a))$  из 2) и 4),

10)  $C(f(a))$  из 8) и 9),

11)  $V(a) \vee S(a, f(a))$  из 1) и 4),

12)  $S(a, f(a))$  из 8) и 11),

13)  $P(f(a))$  из 12) и 5),

14)  $\neg C(f(a))$  из 13) и 7)

15)  $\square$  из 10) и 14).

Заключение доказано.

**Пример 5.** Каждый, кто хранит деньги в кассе, получает проценты. Докажем, что если не существует процентов, то никто не хранит денег.

Примем следующие обозначения:

1)  $S(X, Y)$ : X хранит деньги Y;

2)  $M(X)$ : X есть деньги;

3)  $I(X)$ : X есть проценты;

4)  $E(X, Y)$ : X получает Y проценты.

Посылка записывается в виде:

$$(\forall X)(\exists Y)(S(X, Y) \wedge M(Y)) \rightarrow (\exists Y)(I(Y) \wedge E(X, Y)),$$

а заключение в виде:

$$\neg(\exists X)I(X) \rightarrow (\forall X)(\forall Y)(S(X, Y) \rightarrow \neg M(Y)).$$

Преобразуем посылку в предложения:

1)  $\neg S(X, Y) \vee \neg M(Y) \vee I(f(X))$ ,

2)  $\neg S(X, f(X)) \vee \neg M(Y) \vee E(X, f(X))$ .

Преобразованное в предложение отрицание заключения имеет вид:

3)  $\neg I(z)$ ,

4)  $S(a, b)$ ,

5)  $M(b)$ .

Доказательство заключения:

6)  $\neg S(X, Y) \vee \neg M(Y)$  из 3) и 1),

7)  $\neg M(b)$  из 6) и 4),

8)  $\square$  из 7) и 5).

### КОНТРОЛЬНЫЕ ВОПРОСЫ

1. В чем смысл процедуры резолюции?
2. Что такое резольвента?
3. Какое предложение называется пустым?

### ЗАДАНИЕ

Используя метод резолюций, доказать истинность заключения.

1. Лошадь есть животное, поэтому голова лошади есть голова животного.
2. Полиция обыскивает всех въехавших в страну, за исключением дипломатов. Шпион въехал в страну, однако распознать личность шпиона может только шпион. Шпион не является дипломатом. Среди полицейских имеется шпион.
3. Благородный труд заслуживает уважения, так как благородный труд способствует прогрессу общества. Труд учителя есть благородный труд, так как труд учителя заключается в обучении и воспитании подрастающего поколения. Труд учителя заслуживает уважения.
4. Все тюльпаны – цветы. Все цветы – растения. Все растения используют для питания углекислый газ атмосферы и выделяют в нее кислород. Все растения, использующие для пи-

тания углекислый газ атмосферы и выделяющие в нее кислород, содержат хлорофилл. Все тюльпаны содержат хлорофилл.

5. Все, что требует мужества и героизма, есть подвиг. Первый полет человека в космос требовал мужества и героизма. Первый полет человека в космос есть подвиг.

6. Если животное млекопитающее, то оно относится к типу хордовых. Это животное не является млекопитающим. Это животное не относится к типу хордовых.

7. Всякое преступление карается законом, поскольку оно общественно опасно. Грабеж есть преступление, так как грабеж – это открытое хищение личного имущества граждан. Грабеж карается законом.

8. Если должностное лицо получает взятку, то оно совершает преступление. Должностное лицо не получает взятку. Данное должностное лицо не совершает преступления.

9. Если на металле появились следы ржавчины, то началась коррозия. Коррозия не началась. На металле не появились следы ржавчины.

10. Если эта машина – двигатель внутреннего сгорания, то она является тепловым двигателем. Если эта машина является тепловым двигателем, то в ней топливо сжигается внутри цилиндра. Если эта машина – двигатель внутреннего сгорания, то в ней топливо сжигается внутри цилиндра.

11. Некоторый остров заселен исключительно рабами и сеньорами. На этом острове царствует некая принцесса. Сеньоры всегда говорят правду, а рабы все время лгут. Некоторые сеньоры бедны, другие богаты. Рабы также делятся на богатых и бедных. Принцесса желает выйти замуж за богатого раба. Каким же образом раб может всего лишь одним заявлением убедить принцессу в том, что он достоин быть ее супругом (какая фраза могла бы убедить принцессу в том, что говорящий эту фразу является богатым рабом)?

## **МЕТОДИЧЕСКИЕ УКАЗАНИЯ К ЛАБОРАТОРНЫМ ЗАНЯТИЯМ**

### **ЛАБОРАТОРНАЯ РАБОТА № 1. РАЗРАБОТКА ЭКСПЕРТНЫХ СИСТЕМ, БАЗИРУЮЩИХСЯ НА ПРАВИЛАХ.**

#### **Цель**

1. Ознакомление с принципами построения экспертных систем.
2. Изучение структуры экспертных систем, базирующихся на правилах.
3. Построение простейшей экспертной системы, базирующейся на правилах.

#### **Общие сведения**

Экспертная система - это компьютерная программа, созданная для выполнения тех видов деятельности, которые под силу только человеку- эксперту, – например, проектирования, планирования, постановки диагноза, перевода, реферирования, ревизии, выдачи рекомендаций. Сферы применения экспертных систем – бизнес, проектирование, исследования, управление.

Программы ЭС обычно работают таким способом, который воспринимается как “интеллектуальный”, т. е. они имитируют образ действий человека-эксперта.

Эти программы специфичны, поскольку, как правило, используют механизм автоматического рассуждения (вывода) и так называемые слабые методы – такие как поиск, или эвристика. Они существенно отличаются от точных и хорошо аргументированных алгоритмов и не похожи на математические процедуры большинства традиционных разработок.

Основными компонентами экспертных систем являются:

- база знаний (БЗ), содержащая формализованное описание методов и знаний, привлекаемых при решении задач из области применения экспертных систем;
- механизм вывода (МВ), содержащий формализованное описание правил извлечения знаний из БЗ;
- система пользовательского интерфейса (СПИ), осуществляющая передачу

знаний от МВ к пользователю.

В процессе работы экспертной системы (консультации) входные данные сопоставляются с данными из БЗ. Результатом сопоставления является утвердительный или отрицательный ответ. В экспертных системах, базирующихся на правилах, утвердительный ответ является результатом наличия в БЗ соответствующего продукционного правила. Выбор и активизацию продукционного правила реализует интерпретатор МВ. В каждом цикле работы интерпретатора (называемом распознавание – действие) производятся следующие действия:

- образец правила сопоставляется с элементами данных из БЗ;
- если можно активизировать более одного правила, то для выбора правила используется механизм разрешения конфликта (здесь не рассматривается);
- применяется выбранное правило.

Пример реализации экспертной системы выбора породы собаки, базирующейся на правилах, приведен в прил. 1 (LAB01.PRO).

В программу включен дополнительный раздел **database**, содержащий определение предикатов динамической базы данных (БД).

Запись данных в БЗ производится стандартным предикатом **asserta (Факт)**,

в результате активизации которого указанный в скобках факт будет добавлен в начало БД.

Удаление фактов из БД производится стандартным предикатом **retract (Факт)**,

в результате активизации которого из БД удаляется первый факт, отождествленный с фактом, указанным в скобках.

При сопоставлении правил с содержимым БД используется стандартная функция отрицания (*not*), считающаяся выполненной успешно, если заданный в ней атом представляет собой цель, которая не достигается.

Задание к лабораторной работе

1. Провести тестирование программы LAB01.PRO (см. прил. 1).
2. Изменить программу LAB01.PRO так, чтобы перед окончанием работы выводилось содержимое БД, например, путем использования предиката вида `wr_bd:-dpositive(P,Q)`,

```
write("dpositive(",P,", ",Q,")"), nl,  
fail.
```

3. Изменить программу LAB01.PRO так, чтобы она обеспечивала распознавание животных в соответствии с правилами, приведенными в прил. 2.

Порядок выполнения задания

1. Загрузить Турбо- Пролог.
2. Загрузить программу LAB01.PRO и убедиться в правильности ее работы.
3. Внести требуемые изменения.

## *ЛАБОРАТОРНАЯ РАБОТА № 2* **РАЗРАБОТКА ЭКСПЕРТНЫХ СИСТЕМ, БАЗИРУЮЩИХСЯ НА ЛОГИКЕ**

Цель

1. Изучение структуры экспертных систем, базирующихся на логике.
2. Построение простейшей экспертной системы, базирующейся на логике.

Общие сведения

Структура экспертной системы, базирующейся на логике, аналогична структуре экспертной системы, базирующейся на правилах - БЗ состоит из утверждений в виде предложений логики предикатов; МВ реализует процесс распознавание – действие; СПИ выполняет те же функции, что и в системах, базирующихся на

правилах.

Пример экспертной системы по породам собак, базирующейся на логике, приведен в прил. 1 в виде программы на Турбо-Прологе.

Программа выдает начальное меню, предлагая пользователю выбор между **consultation** (консультацией) и **exit from the system** (выходом из системы). Если пользователь выбирает консультацию, то между пользователем и системой происходит диалог. Затем пользователю сообщается результат. Результатом является либо выбранная порода, либо сообщение **Sorry. I can't help you** (Извините, я не могу вам помочь).

БЗ содержит утверждения логики предикатов, которые представлены либо в форме **rule** (правило), либо в форме **cond** (условие). В форме **rule** хранятся данные о породе; в форме **cond**-атрибуты (условия), характеризующие породу. Данные (ответы), получаемые от пользователя, динамически записываются в БД в форме предикатов **yes** (да) и **no**(нет).

МВ организован следующим образом: в результате активизации правила **go** осуществляется просмотр утверждений из БД **rule** и **cond** для выяснения существования или отсутствия подходящих значений данных. С этой целью вызывается правило **check** (проверка). Это правило содержит трассу номеров правил, номера условий и классифицированные объекты в БЗ. Оно пытается сопоставить объекты, классифицированные при помощи номеров условий. Если сопоставление происходит, то в программу добавляются сопоставленные значения и продолжается сопоставление с новыми данными, полученными от пользователя. Если сопоставления не происходит, МВ останавливает текущий процесс и выбирает другую трассу. Поиск и сопоставление продолжаются до тех пор, пока не исчерпаны все возможности. По завершении вывода **go** через интерфейс передает результаты пользователю.

СПИ состоит из трех частей: в первой содержатся правила для организации меню и уничтожения соответствующего окна после выбора пользователем предлагаемой ему программной функции: либо проведение консультации, либо выход из системы; вторая обеспечивает вывод списка собак и инициализацию процесса поиска и сопоставления по образу; третья запрашивает и получает ответы (**yes** или **no**) от пользователя.

Задание к лабораторной работе

1. Провести тестирование программы LABO2.PRO (см. прил. 1).
2. Изменить программу LABO2.PRO так, чтобы она обеспечивала распознавание животных в соответствии с правилами, приведенными в прил. 2.

Порядок выполнения задания

1. Загрузить Турбо- Пролог.
2. Загрузить программу LABO2.PRO и убедиться в правильности её работы.
3. Внести требуемые изменения.

### ИНДИВИДУАЛЬНЫЕ ЗАДАНИЯ НА РАЗРАБОТКУ ФРАГМЕНТОВ ЭС

1. Придумайте систему управления лифтом, предусматривающую кнопки для трех этажей, а также для закрытия и открытия дверей, сенсорный датчик для обнаружения препятствия, возникающего при закрытии дверей, реле–таймер для фиксации времени, в течение которого двери открыты, и отдельные кнопки вызова на каждом этаже. Напишите систему продукций для управления лифтом. Пример типичной продукции:

Если срок действия таймера истек и  
дверь открыта и  
ничто не препятствует закрытию двери  
То закрыть дверь

2. Требуется разработать фрагмент экспертной системы, предназначенной для прогнозирования местной погоды.

3. Требуется разработать фрагмент экспертной системы, предназначенной для обнаружения того, что делать, если автомобиль не заводится.

4. Требуется разработать фрагмент экспертной системы, предназначенной для определения стратегии гоночной яхты в регате. Система должна работать в режиме реального времени на протяжении всего периода гонок.

5. Требуется разработать фрагмент экспертной системы, предназначенной для помощи отвечающему по телефону доверия, когда отвечающий должен определить яд, который мог быть принят звонящим.

6. Требуется разработать фрагмент экспертной системы, предназначенной для определения оптимального маршрута продавца в любой данный день, причем продавец должен посетить всех клиентов и израсходовать возможное количество бензина.

7. Требуется разработать фрагмент экспертной системы, предназначенной для подбора членов экипажа с учетом их совместимости и возраста. При этом учитываются индивидуальные особенности претендентов, их совместимость, пожелания тех или иных кандидатов работать вместе, а также их возраст (командир должен быть старше всех остальных членов экипажа).

8. Поиск информации в базе данных из другого модуля программы. В базу знаний занесена информация о сотрудниках:

- а) фамилия;
- б) год рождения;
- в) место рождения;
- г) национальность;
- д) профессия;
- е) место работы;
- ж) занимаемая должность.

Система должна по вводимой с клавиатуры частичной информации о сотрудниках находить в базе знаний полную информацию и выводить ее на экран либо сообщать, что необходимая информация в базе данных отсутствует.

9. Требуется разработать фрагмент экспертной системы, предназначенной для поиска свободного места в салоне самолета. Система запрашивает требования к свободному месту

- а) класс салона;
- б) у окна или прохода;
- в) для некурящих или нет

и выводит на экран номер свободного места. Программа и данные содержатся в различных модулях.

10. Требуется разработать фрагмент экспертной системы, предназначенной для диагностики неисправностей персонального компьютера. Экспертная система должна исследовать ситуацию и попытаться определить на общем уровне, допускает ли ошибки пользователь или, действительно, имеется неисправность в системном блоке, на диске, в мониторе и т.д. Возможный путь проектирования – беседа с мастером–профессионалом. При оценке ситуации подразумеваются грубые функциональные тесты, без глубокого анализа электронных элементов.

11. Требуется разработать фрагмент экспертной системы, предназначенной для подбора субоптимальной конфигурации персонального компьютера с учетом субъективных и объективных потребностей заказчика.

12. Требуется разработать фрагмент экспертной системы, предназначенной для подбора субоптимальной конфигурации локальной компьютерной сети с учетом множества эксплуатационных, финансовых и прочих важных критериев.

13. Требуется разработать фрагмент экспертной системы, предназначенной для диагностики широко распространенных заболеваний человека по совокупности симптомов. Диагностируется не менее 20 болезней с учетом 15 типовых симптомов. Каждый симптом

может указывать на несколько болезней (возможно, с разной степенью уверенности).

**14.** Требуется разработать фрагмент экспертной системы, предназначенной для консультации в отношении покупки автомобиля с учетом субъективных факторов, объективных потребностей и платежеспособности клиента, а также сезона и др.

**15.** Требуется разработать фрагмент экспертной системы, предназначенной для консультации в отношении покупки недвижимости с учетом связанных с этим важных факторов (надежность продавца, платежеспособность покупателя, страхование сделки, изменение цен и банковских процентных ставок и др.).

**16.** Требуется разработать фрагмент экспертной системы, предназначенной для выработки рекомендаций по обустройству дорожной сети (установка дорожных знаков, ограждений, нанесение разметки и т.п.). Вид ДТП - наезды на пешеходов, неожиданно выходящих на проезжую часть из-за стоящих транспортных средств.

Возможные мероприятия:

- установка знака 3.27 - "Остановка запрещена";
- установка знака 3.28 - "Стоянка запрещена";
- замена знака 3.28 знаком 3.27;
- обеспечение видимости знака в дневное и ночное время;
- установка дорожного ограждения перильного типа;
- организация поблизости пешеходного перехода;
- организация поблизости оборудованной стоянки транспортных средств;
- принятие мер по соблюдению водителями требований знака.

При выработке рекомендаций следует принять во внимание, не является ли участок дороги, где совершаются наезды, местом притяжения транспортных средств (например, рядом находится крупное учреждение) и/или пешеходов (остановка общественного транспорта, универмаг, кинотеатр и т.п.).

Дорожные знаки должны размещаться с учетом их наилучшей видимости участниками дорожного движения как в светлое, так и в темное время суток, при этом они не должны закрываться от участников дорожного движения какими-либо препятствиями – зелеными насаждениями, мачтами наружного освещения и т.п.. На участках дорог без стационарного освещения следует применять знаки со световозвращающей поверхностью, а на участках со стационарным освещением – знаки с внутренним и внешним освещением. В последнем случае возможна также установка знаков со светоотражающей поверхностью, если обеспечена их видимость с расстояния не менее 100 м.

В соответствии с правилами дорожного движения остановка и стоянка запрещены (даже при отсутствии соответствующего знака) на пешеходных переходах и ближе 5 м перед ними, а также ближе 15 м от остановочных площадок или указателей остановки общественного транспорта, если это создает помехи их движению.

*ПРИЛОЖЕНИЕ 1*

**Текст программы работы №1 (LABO1.PRO).**

```
/*    Пример экспертной системы    */
/*    базирующейся на правилах      */
/*    Эксперт по породам собак      */
domains
```

```
database
    dpositive(symbol, symbol)
    dnegative(symbol, symbol)
```

```
predicates
    do_expert_job
    do_consulting
```

```

ask(symbol, symbol)
dog_is(symbol)
it_is(symbol)
positive(symbol, symbol)
negative(symbol, symbol)
remember(symbol, symbol, symbol)
clear_facts

```

```

goal
do_expert_job.

```

Clauses

```

/* Система пользовательского интерфейса */
do_expert_job:-
    makewindow(1,7,7, "Экспертная система",1,16,22,58),
    nl,write("*****"),
    nl,nl,
    write(" Добро пожаловать в ЖИВОТНУЮ экспертную систему! ;)"),
    nl,nl,write(" Эта система легко определит название животного по "),
    nl,write(" его признакам.                "),
    nl,write(" Отвечайте на вопросы : 'Y'(Да) или 'N'(Нет).  "),
    nl,write("*****"),
    nl,nl,
    do_consulting,
    nl,nl,
    clear_facts,
    write("Нажмите пробел."),nl,
    readchar(_),
    removewindow,
    exit.
do_consulting:-dog_is(X),!,nl,
    write("Похоже, что это - ", X, ".").
do_consulting:-nl, write("Извините, но я ничем не могу вам помочь."), nl,
    write("И вообще, где вы видели такое животное ?..").
ask(X,Y):- write(" Вопрос: ",X," ",Y," ? "),
    readln(Reply),
    remember(X,Y,Reply).

/* Механизм вывода */
positive(X,Y):-dpositive(X,Y),!.
positive(X,Y):-not(negative(X,Y)),!, ask(X,Y).
negative(X,Y):-dnegative(X,Y),!.
remember(X,Y,y):-asserta(dpositive(X,Y)).
remember(X,Y,n):-asserta(dnegative(X,Y)), fail.
clear_facts:-retract(dpositive(_,_)), fail.
clear_facts:-retract(dnegative(_,_)), fail.

/* Продукционные правила */
dog_is("Английский бульдог):-
    it_is("короткая шерсть"),
    positive(has,"рост меньше 55 см"),

```

```

        positive(has,"низкопосаженный хвост"),
        positive(has,"хороший характер"),!.
dog_is("Гончая):-
    it_is("короткая шерсть"),
    positive(has,"рост меньше 55 см"),
    positive(has,"длинные уши"),
    positive(has,"хороший характер"),!.
dog_is ("Дог):-
    it_is("короткая шерсть"),
    positive(has,"низкопосаженный хвост"),
    positive(has,"хороший характер"),
    positive(has,"вес больше 5 кг"),!.
dog_is("Американская гончая):-
    it_is("короткая шерсть"),
    positive(has,"рост меньше 75 см"),
    positive(has,"длинные уши"),
    positive(has,"хороший характер"),!.
dog_is("Коккер-спаниель):-
    it_is("длинная шерсть"),
    positive(has,"рост меньше 55 см"),
    positive(has,"низкопосаженный хвост"),
    positive (has,"длинные уши"),
    positive (has,"хороший характер"),!.
dog_is("Ирландский сеттер):-
    it_is("длинная шерсть"),
    positive(has,"рост меньше 75 см"),
    positive(has,"низкопосаженный хвост"),
    positive(has,"длинные уши"),!.
dog_is ("Колли):-
    it_is("длинная шерсть"),
    positive(has,"рост меньше 75 см"),
    positive(has,"низкопосаженный хвост"),
    positive(has,"хороший характер"),!.
dog_is("Сенбернар):-
    it_is("длинная шерсть"),
    positive(has,"низкопосаженный хвост"),
    positive(has,"хороший характер"),
    positive(has,"вес больше 5 кг"),!.
it_is("короткая шерсть):-
    positive(has,"короткая шерсть"),!.
it_is("длинная шерсть):-
    positive (has,"длинная шерсть"),!.
/*    конец программы        */

```

### Текст программы работы №2 (LABO2.PRO).

```

/*    Пример экспертной системы,        */
/*    базирующейся на логике.            */
/*    Эксперт по породам собак          */
domains
    conditions = bno*
    mo,bno,fno =integer

```

```
category = symbol
```

```
database
```

```
/* предикаты базы данных */  
rule(rno,category,category,conditions)  
cond(bno,symbol)  
yes(bno)  
no(bno)  
topic(symbol)
```

```
predicates
```

```
/* предикаты системы пользовательского интерфейса */  
do_expert_job  
show_menu  
do_consulting  
process(integer)  
info(category)  
goes(category)  
listopt  
erase  
clear  
eval_reply(char)  
/* предикаты механизма вывода */  
go(category)  
check(rno,conditions)  
inpo(rno,bno,string)  
do_answer(rno,string,bno,integer)
```

```
goal
```

```
do_expert_job.
```

```
clauses
```

```
/* база знаний */  
topic("dog").  
topic("Короткошерстная собака").  
topic("Длинношерстная собака").  
  
rule(1,"dog","Короткошерстная собака",[1]).  
rule(2,"dog","Длинношерстная собака",[2]).  
rule(3,"Короткошерстная собака","Английский бульдог", [3,5,7]).  
rule(4,"Короткошерстная собака","Гончая", [3,6,7]).  
rule(5,"Короткошерстная собака","Дог", [5,6,7,8]).  
rule(6,"Короткошерстная собака","Американская гончая", [4,6,7]).  
rule(7,"Длинношерстная собака","Коккер-спаниель", [3,5,6,7]).  
rule(8,"Длинношерстная собака","Ирландский сеттер", [4,6]).  
rule(9,"Длинношерстная собака","Колли", [4,5,7]).  
rule(10,"Длинношерстная собака","Сенбернар", [5,7,8]).  
  
cond(1,"Короткая шерсть").  
cond(2,"Длинная шерсть").  
cond(3,"Рост меньше 55 см").  
cond(4,"Рост меньше 75 см").
```

```

cond(5,"Низкопосаженный хвост").
cond(6,"Длинные уши").
cond(7,"Хороший характер").
cond(8,"Вес более 5 кг").

```

```

/* Система пользовательского интерфейса */

```

```

do_expert_job:-
    makewindow(1,7,7,"DOG EXPERT SYSTEM",0,0,25,80),
    show_menu,
    nl,write("Press spase bar."),
    readchar(_),
    exit.

show_menu:-
    write("                "),nl,
    write("*****"),nl,
    write("*      DOG EXPERT      *"),nl,
    write("*                *"),nl,
    write("*  1. Consultation      *"),nl,
    write("*                *"),nl,
    write("*  2. Exit the system   *"),nl,
    write("*                *"),nl,
    write("*****"),nl,
    write("                "),nl,
    write("Please enter your choice: 1 or 2: "),nl,
    readint(Choice),
    process(Choice).

process(1):-do_consulting.
process(2):-removewindow, exit.
do_consulting:-goes(Mygoal),go(Mygoal),!.
do_consulting:-nl,write("Sorry, I can't help you."),
    clear.
do_consulting.
goes(Mygoal):-clear,clearwindow,nl,nl,
    write("                "),nl,
    write(" WELCOME TO THE DOG EXPERT SYSTEM "),nl,
    write("                "),nl,
    write(" This is a dog identification system."),nl,
    write(" To begin the process of choosing a "),nl,
    write(" dog, please type in 'dog'. If you "),nl,
    write(" wish to see the dog types, please "),nl,
    write(" type in a question mark (?).      "),nl,
    write("                "),nl,
    readln(Mygoal),
    info(Mygoal),!.
info("?"):-clearwindow,
    write("Reply from the KBS."),nl,
    listopt,nl,
    write("Please any key."),
    readchar(_),
    clearwindow,
    exit.
info(X) :- X >< "?".

```

```

listopt :-
    write("The dog types are: "),nl,nl,
    topic(Dog),
    write("      ",Dog),nl,fail.
listopt.
inpo(Rno,Bno,Text) :-
    write("Question :-",Text," ? "),
    makewindow(2,7,7,"Response",10,54,7,20),
    write("Type 1 for 'yes': "),nl,
    write("Type 2 for 'no' : "),nl,
    readint(Response),
    clearwindow,
    shiftwindow(1),
    do_answer(Rno,Text,Bno,Response).
eval_reply('y') :-
    write("I hope you have found this helpful !").
eval_reply('n') :-
    write("I am sorry I can't help you !").
go(Mygoal) :-
    not(rule(_,Mygoal,_,_)),!,nl,
    write("The dog you have indicated is a(n) ",Mygoal,"."),nl,
    write(" Is a dog you would like to have (y/n) ?"),nl,
    readchar(R),
    eval_reply(R).

```

/\* МЕХАНИЗМ ВЫВОДА \*/

```

go(Mygoal) :-
    rule(Rno,Mygoal,Ny,Cond),
    check(Rno,Cond),
    go(Ny).
check(Rno,[Bno|Rest]) :-
    yes(Bno),!,
    check(Rno,Rest).
check(_,[Bno|_]) :- no(Bno),!,fail.
check(Rno,[Bno|Rest]) :-
    cond(Bno,Text),
    inpo(Rno,Bno,Text),
    check(Rno,Rest).
check(_,[]).
do_answer(_,_,0) :- exit.
do_answer(_,_,Bno,1) :-
    assert(yes(Bno)),
    shiftwindow(1),
    write(yes),nl.
do_answer(_,_,Bno,2) :-
    assert(no(Bno)),
    write(no),nl,
    fail.
erase :- retract(_),fail.
erase.
clear :-
    retract(yes(_)),

```

```
retract(no(_)),  
fail,!.  
clear.  
/* конец программы */
```

## ПРИЛОЖЕНИЕ 2

### Система правил для экспертной системы распознавания животных

#### ПРАВИЛО 1

ЕСЛИ животное имеет волосяной покров  
ТО это животное – млекопитающее

#### ПРАВИЛО 2

ЕСЛИ животное дает молоко  
ТО это животное – млекопитающее

#### ПРАВИЛО 3

ЕСЛИ животное имеет перья  
ТО это – птица

#### ПРАВИЛО 4

ЕСЛИ животное может летать  
И откладывает яйца  
ТО это животное – птица

#### ПРАВИЛО 5

ЕСЛИ животное ест мясо  
ТО это животное – хищник

#### ПРАВИЛО 6

ЕСЛИ животное имеет острые зубы  
И животное имеет когти  
И его глаза смотрят вперед  
ТО это животное – хищник

#### ПРАВИЛО 7

ЕСЛИ животное является млекопитающим  
И имеет копыта  
ТО это животное – парнокопытное

#### ПРАВИЛО 8

ЕСЛИ животное является млекопитающим  
И жует жвачку  
ТО это животное – парнокопытное

#### ПРАВИЛО 9

ЕСЛИ животное является млекопитающим  
И это животное – хищник  
И это животное желто-коричневого цвета  
И это животное имеет темные пятна  
ТО можно предположить, что это животное – гепард

#### ПРАВИЛО 10

ЕСЛИ животное является млекопитающим  
И это животное – хищник  
И это животное желто-коричневого цвета  
И это животное имеет темные полосы  
ТО можно предположить, что это животное – тигр

#### ПРАВИЛО 11

ЕСЛИ животное является парнокопытным  
И имеет длинную шею

И имеет длинные ноги  
 И имеет черные пятна  
 ТО можно предположить, что это животное – жираф

**ПРАВИЛО 12**  
 ЕСЛИ животное является парнокопытным  
 И имеет черные полосы  
 ТО можно предположить, что это животное – зебра

**ПРАВИЛО 13**  
 ЕСЛИ животное является птицей  
 И не может летать  
 И имеет длинную шею  
 И имеет длинные ноги  
 И имеет черно-белую окраску  
 ТО можно предположить, что это животное – страус

**ПРАВИЛО 14**  
 ЕСЛИ животное является птицей  
 И не может летать  
 И может плавать  
 И имеет черно-белую окраску  
 ТО можно предположить, что это животное – пингвин

**ПРАВИЛО 15**  
 ЕСЛИ животное является птицей  
 И хорошо летает  
 ТО можно предположить, что это животное – альбатрос

### **Практическая работа № 1, 2**

Системы ИАД применяются в научных исследованиях и образовании, в работе правоохранительных органов, производстве, здравоохранении и многих других областях. Особенно широко технология ИАД используется в деловых приложениях. В таблице 1 приведено несколько примеров приложений в области финансов и маркетингового анализа.

<b>Таблица 1. Примеры использования методов интеллектуального анализа данных в финансовых приложениях и маркетинговом анализе</b>	
<b>Приложение (организация)</b>	<b>Описание</b>
FALCON (HNC Software, Inc.)	Инструментальное средство для оперативного выявления злоупотреблений с кредитными карточками; более 100 организаций-пользователей отмечают сокращение числа нарушений на 20-30%.
Классификатор дебиторских счетов (Internal Revenue Service)	Выявление счетов потенциально платежеспособных дебиторов на основе анализа больших объемов архивных данных по уплате налогов.
Повышение качества архивной финансовой информации (Lockheed)	Выявление закономерностей (в виде правил вывода) в архивных финансовых данных для использования в моделях прогнозирования, системах поддержки принятия решений по инвестированию и т.д.
Верификация данных по курсам валют (Reuters)	Система выявления ошибок в оперативно поступающих данных по курсам валют. С помощью нейронных сетей и индук-

	тивного вывода правил строятся приблизительные прогнозы, которые сравниваются с поступающими данными. Большие отклонения рассматриваются как возможные ошибки.
Прогнозирование невыплат в сделках с недвижимостью (Leeds)	Анализ архивных данных по сделкам с недвижимостью и выявление паттернов, соответствующих проблемным сделкам, заканчивающимся невыплатами. Выявленные закономерности используются для оценки риска при заключении новых сделок.
Маркетинговые исследования (Dickinson Direct)	Определение характеристик типичных покупателей продукции компании для выявления новых потенциальных клиентов (профилирование клиентов).
Маркетинговые исследования (Reader's Digest Canada)	Выявление основных сегментов рынка и наиболее благоприятных подмножеств, а также исследование зависимостей между основными показателями и характеристиками сегментов.
Установка лотерейных автоматов (Automated Wagering, Inc.)	Объединение методов ИАД с географическим анализом для определения наилучших мест для установки лотерейных автоматов в штате Флорида.
Выявление потенциальных покупателей автомобильных стерео систем (Washington Auto Audio, Inc.)	Анализ демографической базы данных, содержащей информацию о 14000 реальных и потенциальных клиентов, позволил за 90 секунд получить 3 довольно надежных индикатора для прогноза спроса на продукцию и услуги компании. Аналогичные результаты были получены в результате традиционного исследования, выполненного одной из консалтинговых компаний, причем это исследование обошлось фирме на порядок дороже, чем автоматизированная система интеллектуального анализа данных.

**Таблица 2. Основные технологии интеллектуального анализа данных**

Технология	Достоинства	Недостатки
Правила вывода	Удобны в тех случаях, когда данные связаны отношениями, представляемыми в виде правил "если то "	При большом количестве правил теряется наглядность; не всегда удается выделить отношения "если то "
Нейронные сети	Удобны при работе с нелинейными зависимостями, зашумленными и неполными данными	"Черный ящик": модель не может объяснить выявленные знания; данные обязательно должны быть преобразованы к числовому виду
Нечеткая логика	Ранжируют данные по степени близости к желаемым результатам; нечет-	Технология новая, поэтому сейчас известно ограниченное число специализированных приложений

	кий поиск в базах данных	
Визуализация	Многомерное графическое представление данных, по которому пользователь сам выявляет закономерности - образцы и отношения между данными	Модели не исполняются, и их интерпретация полностью зависит от аналитика
Статистика	Есть большое число алгоритмов и опыт их применения в научных и инженерных приложениях	Больше подходят для проверки гипотез, чем для выявления новых закономерностей в данных
К-ближайший сосед	Выявление кластеров, обработка целостных источников данных	Большие затраты памяти, проблемы с чувствительностью
Интегрированные	Можно выбирать подходы, адекватные задачам, или сравнивать	Сложные средства поддержки; высокая стоимость;
технологии	результаты применения разных подходов.	для каждой отдельно взятой технологии не всегда реализуется наилучшее решение

### **ТЕМАТИКА НАУЧНЫХ ДОКЛАДОВ ПО ДИСЦИПЛИНЕ «ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ»**

1. Тема: Базы и банки данных, их роль в построении и функционировании экономических информационных систем.
2. Тема: Основные направления развития автоматизации управления.
3. Тема: Особенности автоматизации производственных и административных систем управления.
4. Тема: Системы автоматизации проектных работ (САПР).
5. Тема: Экспертные системы, их применение для решения задач различных предметных областей.
6. Тема: Системы искусственного интеллекта, классификация, особенности применения для решения экономических и управленческих задач.  
Пояснение: при освящении данной темы необходимо рассмотреть экспертные системы, системы поддержки принятия решений и нейросетевые (нейронные) системы.
7. Тема: Роль автоматизированных систем поддержки принятия решений в управлении экономическими объектами.
8. Тема: Системы автоматизированной обработки информации, их классификация области применения.
9. Тема: Автоматизация обработки текстовых документов.
10. Тема: Инструментарий технологий программирования. Средства для создания приложений.
11. Тема: Инструментарий технологий программирования. Средства для создания информационных систем.
12. Тема: Применение информационных технологий в предпринимательской деятельности.

13. Тема: Эффективные средства организации и разработки пользовательского интерфейса в программах с пакетным и диалоговым режимами обработки информации.
14. Тема: Состояние и тенденции развития прикладного программного обеспечения для решения экономических проблем.
15. Тема: Информационные технологии: классификация, особенности, тенденции развития.
16. Тема: Информационные технологии, их роль в проектировании и функционировании информационных систем.
17. Тема: Информационные технологии в управлении деятельностью предприятия.
18. Тема: Формализация и структурирование данных при проектировании баз данных. Модели данных.
19. Тема: Формализация и структурирование знаний при проектировании баз знаний. Модели знаний.
20. Тема: Роль автоматизированных информационных технологий в поддержке процесса принятия решений.
21. Тема: Роль сетевых информационных технологий в глобализации экономики.
22. Тема: Информационные технологии в маркетинге и рекламе.
23. Тема: Технология обработки информации посредством табличных процессоров, её прикладное значение в экономике.
24. Тема: Бухгалтерские автоматизированные информационные системы, их прикладное значение.
25. Тема: Программный продукт и его жизненный цикл
26. Тема: Автоматизированные рабочие места как способ повышения эффективности деятельности различных специалистов.

### **МЕТОДИЧЕСКИЕ УКАЗАНИЯ ДЛЯ САМОСТОЯТЕЛЬНОЙ РАБОТЫ СТУДЕНТОВ**

Самостоятельная работа студентов предполагает самостоятельное изучение отдельных тем, дополнительную подготовку студентов к каждому практическому занятию.

В процессе изучения дисциплины «Экспертные системы» обучающиеся должны выполнить следующие виды самостоятельной работы:

самоподготовку к учебным занятиям по конспектам, учебной литературе и с помощью электронных ресурсов;

подготовку к тестированию по темам дисциплины.

Формой самостоятельной работы является работа с литературой. Овладение методическими приемами работы с литературой - одна из важнейших задач студента. Работа с литературой включает следующие этапы: предварительное знакомство с содержанием; углубленное изучение текста с преследованием следующих целей: усвоить основные положения; усвоить фактический материал; логическое обоснование главной мысли и выводов.

По выбранной теме студенты выполняют реферативную работу.

Работа включает следующие разделы:

1. Обоснование актуальности выбранной тематики и описание целей выполнения работы.
2. Систематизация и анализ найденных в научной печати, в сети Интернет и других источниках материалов.
3. Выводы.
4. Предложения по использованию результатов работы в конкретных областях и возможные направления дальнейших исследований.
5. Составление и отладка программы.
6. Составление отчета.