

Министерство образования и науки Российской Федерации

Федеральное государственное бюджетное образовательное
учреждение высшего образования
«Пермский национальный исследовательский
политехнический университет»

Д.А. Истомин, В.Ю. Столбов

ИНТЕЛЛЕКТУАЛЬНЫЕ СИСТЕМЫ И ТЕХНОЛОГИИ

Лабораторный практикум

Утверждено

Редакционно-издательским советом университета

Издательство

Пермского национального исследовательского
политехнического университета

2017

УДК 004.8
И89

Рецензенты:
доктор технических наук, профессор
Пермского государственного национального исследовательского
университета
Л.Н. Ясницкий;
доктор физико-математических наук, профессор
Пермского национального исследовательского политехнического
университета
М. Б. Гитман

Истомин, Д.А.

И89 Интеллектуальные системы и технологии: лаб. практикум /
Д.А. Истомин, В.Ю. Столбов. — Пермь : Изд-во Перм. нац. ис-
след. политехн. ун-та, 2017. — 38 с.
ISBN 978-5-398-01885-1

Приведены краткие сведения о технологиях интеллектуализа-
ции информационных систем, а также пять лабораторных работ
для практической реализации рассматриваемых технологий.

Предназначено для студентов высших учебных заведений,
обучающихся по направлению «Информационные системы и тех-
нологии».

УДК 004.8

ISBN 978-5-398-01885-1

© ПНИПУ, 2017

Оглавление

Введение	4
1. Экспертные системы	6
1.1. Продукционные правила	8
1.1.1. Машина логического вывода	8
1.2. Фреймы	12
1.2.1. Пример использования фреймов	13
1.3. Семантические сети	16
1.4. Лабораторная работа №1	20
1.5. Лабораторная работа №2	22
1.6. Лабораторная работа №3	24
2. Экспертные системы поддержки принятия коллектив-	
ных решений	26
2.1. Модели голосования при принятии коллективных	
решений	26
2.2. Лабораторная работа №4	29
3. Нейронные сети	31
3.1. Нейросетевые технологии распознавания образов .	32
3.2. Лабораторная работа №5	35
Список литературы	37

Введение

Интеллектуальная информационная система (ИИС) – информационная система, для которой характерны следующие признаки [1]:

1. Развитые коммуникативные способности – взаимодействие конечного пользователя с системой при помощи естественно-языкового интерфейса, когнитивной графики и пр.
2. Умение решать сложные плохо формализуемые задачи, которые требуют построения оригинального алгоритма решения в зависимости от конкретной ситуации, характеризующейся неопределенностью и динамичностью исходных данных и знаний.
3. Способность к самообучению – умение системы автоматически извлекать знания из накопленного опыта и применять их для решения задач.
4. Адаптивность – способность системы к развитию в соответствии с объективными изменениями области знаний.

Каждому из перечисленных признаков условно соответствует свой класс ИИС. Различные системы могут обладать одним или несколькими признаками интеллектуальности с различной степенью проявления.

Можно выделить три основные стратегии создания ИИС [2]:

- 1) высокоуровневая стратегия, основанная на технологиях экспертных систем (ЭС);
- 2) среднеуровневая стратегия, основанная на технологиях эволюционного моделирования;
- 3) низкоуровневая стратегия, основанная на нейросетевых и нейрокомпьютерных технологиях.

Технологии экспертных систем основаны на явных знаниях о предметной области, получаемых от экспертов или других источников информации и формализованных в ИИС с помощью моделей представления знаний. С помощью алгоритмов логического вывода ЭС делает обоснованное заключение, моделируя интеллект человека-эксперта.

Вторая стратегия базируется на теории возникновения и эволюции жизни на Земле. Здесь искусственный интеллект закладывается в самом процессе эволюции и взаимодействии особей различных популяций. Знания при эволюционных технологиях приобретаются в ходе конкурентной борьбы между особями, лучшие из которых остаются и изменяются, следуя поставленной цели, при этом наследуя положительные свойства своих родителей. Яркими представителями этой стратегии являются технологии, объединенные под названием «генетический алгоритм», примеры применения которого для решения различных интеллектуальных задач приведены в [1].

На нижнем уровне абстракции находятся нейросетевые технологии [2], предполагающие построение моделей, имитирующих работу мозга человека и состоящих из большого числа нейронов и связей между ними. Здесь знания приобретаются в ходе обучения модели на специально подобранных примерах из исследуемой предметной области.

В данном пособии будут рассмотрены только технологии экспертных систем и нейросетевые технологии, получившие в настоящее время наибольшее применение для решения широкого класса задач в различных областях предметной деятельности человека.

Экспертные системы

Экспертные системы (ЭС) как самостоятельное направление в искусственном интеллекте сформировалось в конце 1970-х годов. История ЭС началась с сообщения японского комитета по разработке ЭВМ пятого поколения, в котором основное внимание уделялось развитию таких «интеллектуальных способностей» компьютеров, которые могли бы оперировать не только данными, но и знаниями, как это делают специалисты (эксперты) при выработке умозаключений. Группа по экспертным системам при Комитете British Computer Society определила ЭС как «воплощение в ЭВМ компоненты опыта эксперта, основанной на знаниях, в такой форме, что машина может дать интеллектуальный совет или принять решение относительно обрабатываемой функции». Одним из важных свойств ЭС является способность объяснить ход своих рассуждений понятным для пользователя образом [1].

Главное отличие ЭС и систем искусственного интеллекта от систем обработки данных состоит в том, что в них используется символьный, а не числовой способ представления данных, а в качестве методов обработки информации применяются процедуры логического вывода и эвристического поиска решений.

Рассмотрим подробно архитектуру ЭС (рис. 1.1). Важно отметить, что архитектура не зависит от способа представления знаний, которыми оперирует ЭС.

Архитектура ЭС включает следующие компоненты [3]:

1. **Диалоговый компонент.** Компонент, с помощью которого происходит общение пользователя и экспертной системы.
2. **Компонент объяснения.** Компонент, позволяющий объяснить пользователю ход рассуждений системы.
3. **Рабочая память.** Глобальная база фактов, используемых в правилах.

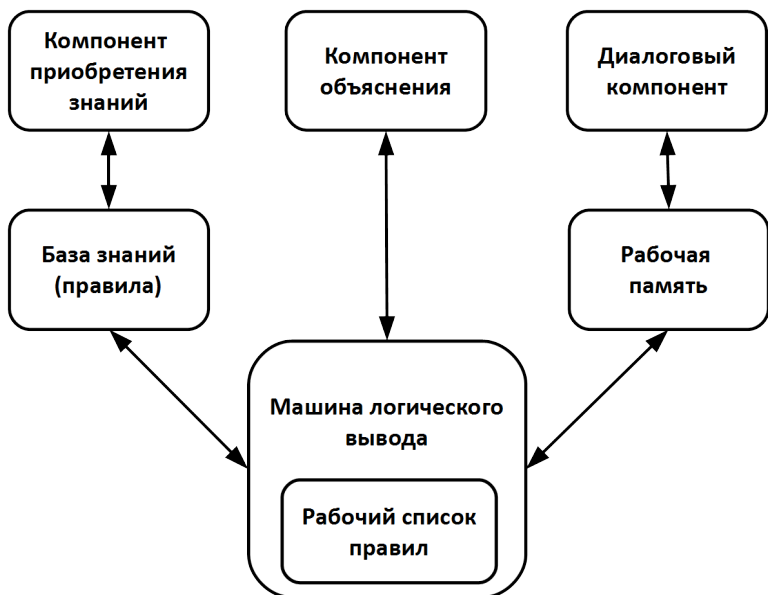


Рис. 1.1. Архитектура ЭС

4. **Машина логического вывода.** Программный компонент, который обеспечивает формирование логического вывода (принимая решение о том, каким правилам удовлетворяют факты или объекты), располагает выполняемые правила по приоритетам и выполняет правило с наивысшим приоритетом.
5. **Рабочий список правил.** Созданный машиной логического вывода и расположенный по приоритетам список правил, шаблоны которых удовлетворяют фактам или объектам, находящимся в рабочей памяти.
6. **Компонент приобретения знаний.** Автоматизированный способ, позволяющий пользователю вводить знания в систему, а не привлекать к решению задачи инженера по знаниям.

Далее рассмотрим три различных способа представления знаний: продукции, фреймы, семантические сети.

1.1. Продукционные правила

Больше всего распространены системы, основанные на правилах. В таких системах знания представлены не с помощью декларативного представления знаний, а в форме многочисленных правил, которые указывают, какие заключения должны быть сделаны или не сделаны в различных ситуациях.

Каждое правило обозначается именем (листинг 1.1). Вслед за именем находится часть IF правила. Участок правила между частями IF и THEN правила упоминается под разными именами, такими как *антецедент*, условная часть, и т.п. Вслед за частью THEN правила находится список действий, которые должны быть выполнены после запуска правила. Эта часть правила называется *консеквентом*, или правой частью.

Листинг 1.1. Общий вид правила

Правило: "имя правила" IF "условие" THEN "заключение"
--

Для более наглядного представления продукционных знаний можно использовать деревья И/ИЛИ. Такие деревья могут быть преобразованы в так называемый граф И/ИЛИ (рис. 1.2).

1.1.1. Машина логического вывода

В системе, основанной на правилах, машина логического вывода (МЛВ) определяет, какие антецеденты правил (если таковые вообще имеются) выполняются согласно фактам. В качестве стратегий решения задач в экспертных системах обычно используются два общих метода логического вывода: прямой логический вывод и обратный логический вывод.

Прямой логический вывод представляет собой метод формирования рассуждений от фактов к заключениям, которые следуют из этих фактов. Обратный логический вывод предусматривает формирование рассуждений в обратном направлении —

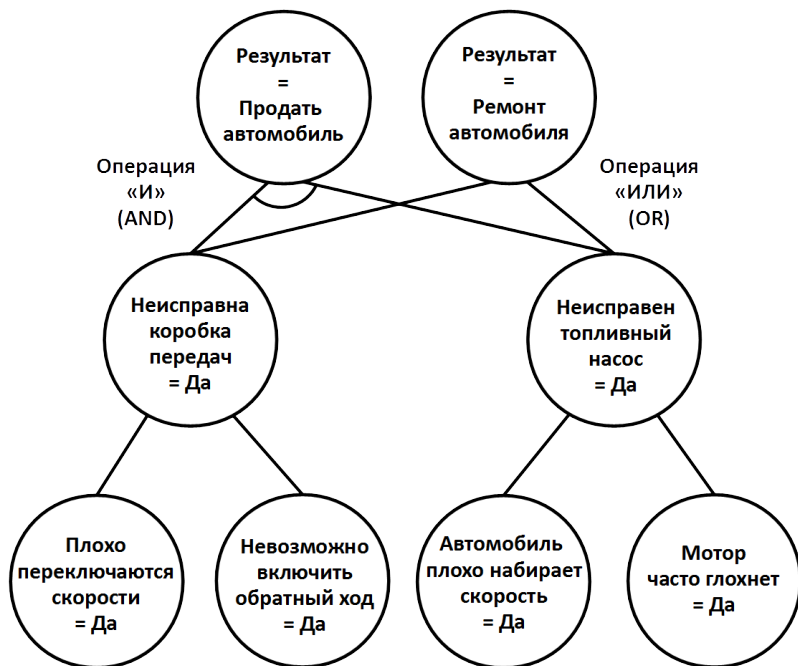


Рис. 1.2. Пример графа И/ИЛИ

от гипотезы (потенциального заключения, которое должно быть доказано) к фактам, которые подтверждают гипотезу.

Рассмотрим пример прямого и обратного логического вывода (листинг 1.2). При прямом логическом выводе на вход подается факт («Цвет неба» = «Голубой») и выводится заключение («Действие» = «Прогулка»). При обратном логическом выводе движение идет от гипотезы («Действие» = «Прогулка») и факту «Прогноз» присваивается значение «Солнечно». Далее срабатывает присваивание «Цвет неба» = «Голубой».

При реализации логического вывода важно помнить следующий алгоритм:

1. В рабочую память попадает факт. Факт может попасть как и из диалогового компонента (т.е. от пользователя), так

Листинг 1.2. Пример правил для логического вывода

Правило: солнечно

IF "Цвет неба" = "Голубой" THEN "Прогноз" = "Солнечно"

Правило: прогулка

IF "Прогноз" = "Солнечно" THEN "Действие" = "Прогулка"

и как результат срабатывания другого правила.

2. МЛВ выбирает то правило, которое может сработать в данный момент. В простейшем случае – полный перебор всех правил.

В ходе логического вывода может оказаться так, что в один момент времени могут сработать одновременно несколько правил. Для решения данной проблемы нужно осуществить операцию «разрешение конфликта». Существует множество способов разрешения конфликтов:

1. Приоритет отдается тем правилам, у которых больше явно указанный приоритет.
2. Приоритет отдается тем правилам, у которых антецедент является более конкретным (т.е. меньше условий в левой части правила).
3. Приоритет отдается тем правилам, которые в ходе логического вывода еще не срабатывали.

И другие способы разрешения конфликтов.

На рис. 1.3 представлены возможные способы устранения данных неопределенностей [3, гл. 5].

Безусловно, полный перебор правил неэффективен. Для осуществления более быстрого вывода было разработано множество алгоритмов, например алгоритм RETE. При использовании алгоритма Rete строится своего рода префиксное дерево, в котором узлы соответствуют части условий правил. Тогда путь от корня до листа дерева образует полное условие некоторой продукции. Алгоритм Rete жертвует объемом памяти ради скорости. Rete стал основой многих популярных экспертных систем, включая CLIPS, Jess, Drools, BizTalk Rules Engine и Soar.

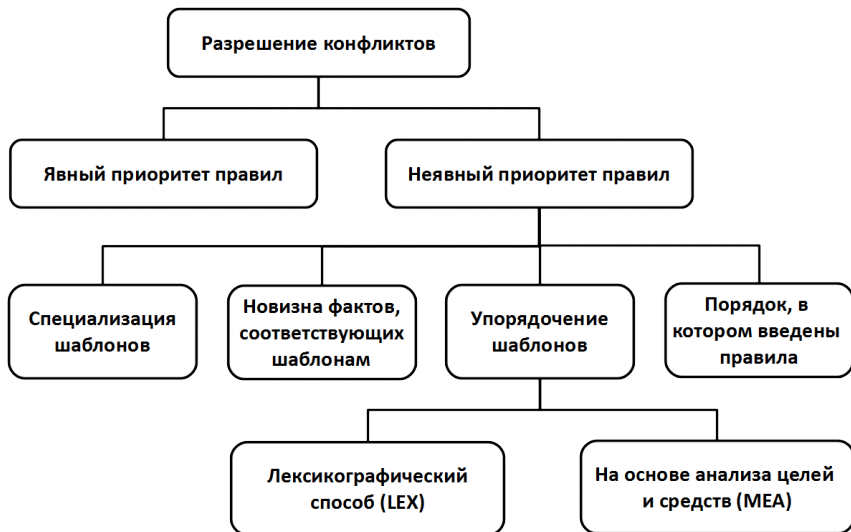


Рис. 1.3. Способы устранения неопределенностей

Для реализации базы знаний (БЗ) можно воспользоваться существующими оболочками экспертных систем. Например, системой CLIPS (C Language Integrated Production System) [3, гл. 7].

1.2. Фреймы

Фрейм является структурой данных для представления стереотипной ситуации [6].

Фреймовая модель представления знаний основана на теории фреймов Марвина Минского [6].

Отправным моментом для данной теории служит тот факт, что человек, пытаясь познать новую для себя ситуацию или по-новому взглянуть на уже привычные вещи, выбирает из своей памяти некоторую структуру данных (образ), называемую нами *фреймом*, с таким расчетом, чтобы путем изменения в ней отдельных деталей сделать ее пригодной для понимания более широкого класса явлений или процессов [6].

Фреймы могут быть связаны друг с другом и тем самым образовывать иерархию фреймов. Визуально фрейм можно представить в виде таблицы (табл. 1.1).

Таблица 1.1

Структура фрейма

Имя фрейма				
Имя слота	Указатель наследования	Указатель типа данных	Значение слота	Присоединенная процедура
Слот 1				
Слот 2				
...				
Слот N				

Рассмотрим более подробно структуру фрейма:

1. **Указатель наследования.** Показывает, какую информацию об атрибутах слотов во фрейме верхнего уровня наследуют слоты-потомки. Возможные значения: U (Unique), S (Same), R (Range), O (Override).

2. **Указатель типа данных.** Возможные значения: INTEGER, TEXT, LISP, FRAME
3. **Значение слота.** Непосредственное значение, которое должно удовлетворять требованиям указанного типа данных.
4. **Присоединенная процедура.** Процедуры, которые активируются при обращении к слотам. Возможные значения: «IF-NEEDED», «IF-ADDED», «IF-REMOVED» и т.д.

У каждого фрейма выделяют системные слоты, например, слот для указания на родительский фрейм для организации иерархии. Данный слот чаще всего имеет имя АКО (A Kind Of).

Существует множество языков, позволяющих описать сеть фреймов, более подробно можно ознакомиться с языком FRL [8].

Процесс вывода по сети фреймов можно кратко сформулировать следующим образом:

1. Создается новый незаполненный фрейм (протофрейм).
2. Создается ссылка от незаполненного фрейма на уже заполненный фрейм (экзофрейм) из БЗ.
3. Для каждого слота осуществляется процедура получения значения путем обхода иерархии фреймов.
4. Процесс заканчивается, когда всем слотам присвоено значение.

1.2.1. Пример использования фреймов

Рассмотрим пример сети фреймов на рис. 1.4. Пусть задана некоторая предметная область – книжное издательство. Считается, что в данном издательстве все материалы изготавливаются за 10 дней и брошюруются скобкой, если другого не задано. Однако для типа материала «Методическое пособие» время изготовления будет исчисляться, исходя из количества страниц, а изготовление типа «Книга» всегда занимает 100 дней.

Необходимо узнать, как долго будет изготавливаться издание и какой будет тип переплета у методического пособия автора «Иванов И. И.» с названием «Лаб. практикум» размером в 38 страниц.

После создания пустого слота с названием «Методическое пособие Иванова И. И.» поочередно всем пустым слотам будет присваиваться значение:

1. Слоту «Автор» присваивается значение, полученное из запроса.
2. Слоту «Название» присваивается значение, полученное из запроса.
3. Слоту «Количество страниц» присваивается значение, полученное из запроса.
4. Слоту «Тип переплета» присваивается значение, исходя из значения «по-умолчанию» слота «Учебный материал для печати».
5. Слоту «Время изготовления» присваивается значение, исходя из присоединенной процедуры у родительского слота «Методическое пособие».

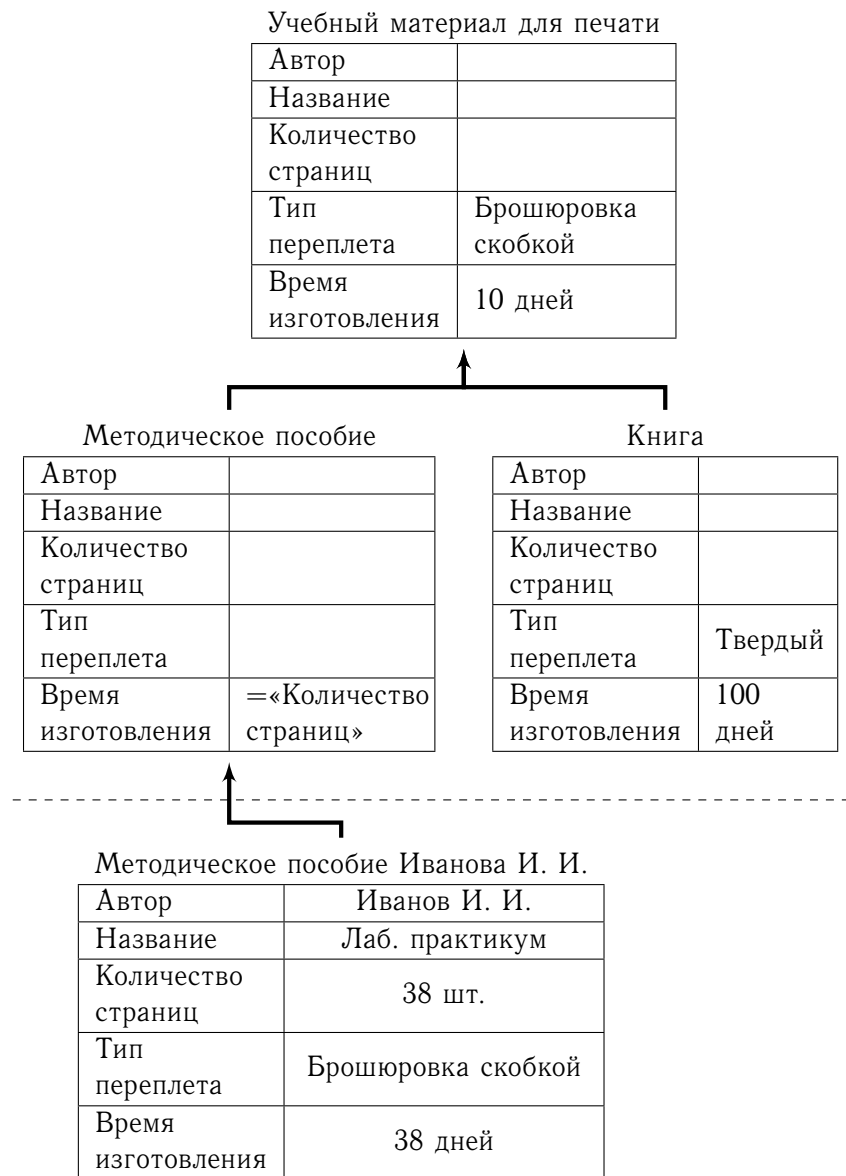


Рис. 1.4. Краткий пример сети фреймов

При реализации структуры хранения можно воспользоваться объектно-ориентированным подходом (рис. 1.5).

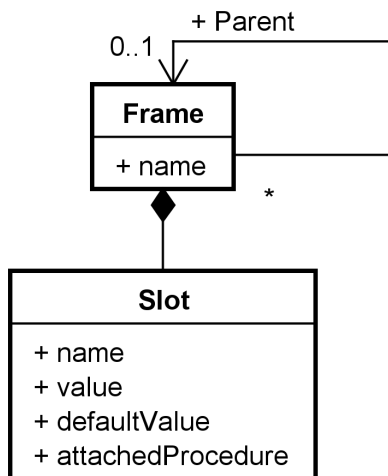


Рис. 1.5. Краткое UML описание структуры фреймов

1.3. Семантические сети

Структура семантической сети отображается графически с помощью узлов и соединяющих их дуг. Узлы иногда именуются объектами а дуги – связями, или ребрами. Связи в семантической сети применяются для представления отношений, а узлы, как правило, служат для представления физических объектов, концепций или ситуаций.

Наиболее часто в семантических сетях встречаются следующие виды отношений:

1. Иерархические («род – вид», «элемент – множество», «часть – целое» и т.п.);
2. Функциональные («аргумент – функция», а также связи, определяемые глаголами, например, «влияет», «создает» и

- другие);
3. Количественные («больше», «меньше»);
 4. Пространственные («далеко от», «близко от», «над», «под» и т.д.);
 5. Временные («раньше», «позже», «в течение»);
 6. Атрибутивные («иметь свойство», «иметь значение»);
 7. Каузальные (причинно-следственные);
- И другие.

Семантические сети могут быть использованы для представления текста на естественном языке. Рассмотрим пример, описывающий следующий текст: «Иван прочитал книгу днем в сквере» (рис. 1.6).

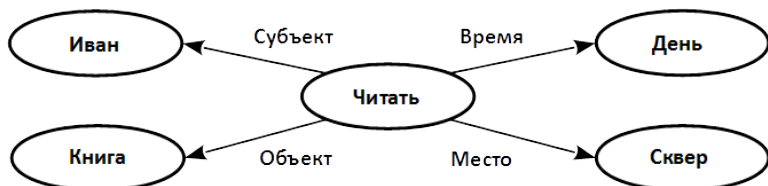


Рис. 1.6. Модель текста на естественном языке

Рассмотрим подробно две широко распространенные иерархические связи: IS-A и A-KIND-OF (связь последнего типа иногда записывается как АКО). На рис. 1.7 показана семантическая сеть, в которой используются такие связи. На данном рисунке связь IS-A означает «является экземпляром» данного класса и указывает на конкретный экземпляр некоторого класса. В данном случае связь АКО используется для обозначения отношения одного класса с другим. Связь АКО не может служить для обозначения отношений конкретных объектов; для этого предназначена связь IS-A. Связь АКО определяет отношение между отдельным классом и родительским классом (или классами), применительно к которому этот отдельный класс является дочерним.

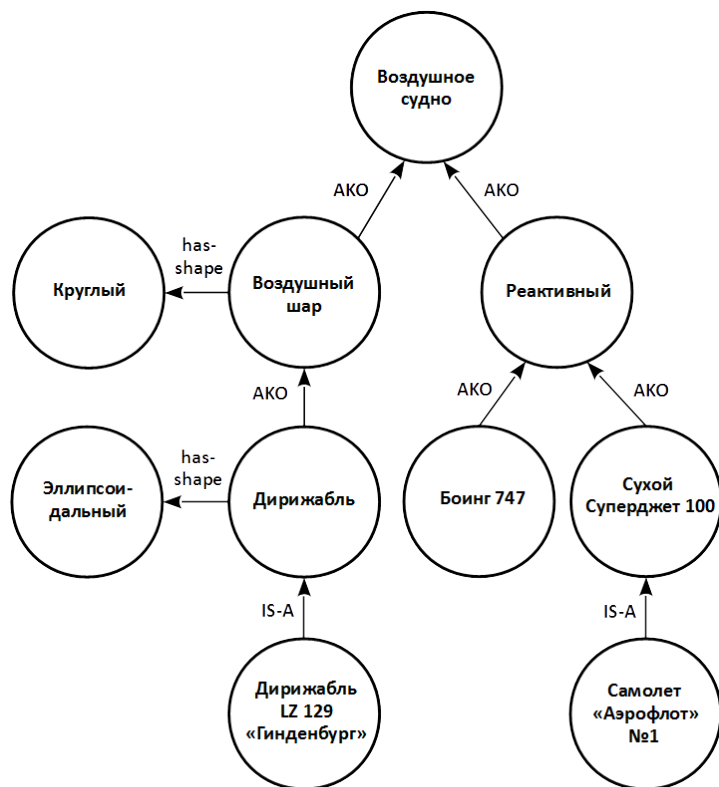


Рис. 1.7. Семантическая сеть с иерархическими связями

Еще одной широко применяемой связью является HAS-A, которая устанавливает отношение между классом и подклассом. Связь HAS-A направлена в сторону, противоположную по отношению к связи АКО, и часто используется для обозначения отношения между одним объектом и частью другого объекта.

Такие три понятия, как объект, атрибут и значение, встречаются вместе настолько часто, что иногда обнаруживается возможность создать упрощенную семантическую сеть с использованием только этих понятий. Чтобы охарактеризовать все зна-

ния, представленные в семантической сети, можно воспользоваться тройкой «объект – атрибут – значение» (object – attribute – value).

Типы запросов определяются типами связей между вершинами. Рассмотрим примеры запросов, на которые может способна экспертная система с использованием знаний на рис. 1.7:

1. **Является ли** «Самолет „Аэрофлот“ №1» «воздушным судном»?
2. **Имеет ли** «воздушный шар» «круглую форму»?
3. **Какие есть** типы «воздушных судов»?

Для реализации хранения и реализации вывода на семантических сетях можно воспользоваться абстрактной структурой данных под названием «Граф». Логический вывод можно реализовать, используя алгоритм обхода в глубину/ширину [7, гл. 22].

1.4. Лабораторная работа №1

Проектирование и реализация продукционной базы знаний

Цель: сформировать у студентов способность проектирования и разработки базы знаний, основанной на продукционной модели их представления.

Описание: в рамках данной работы необходимо спроектировать продукционную БЗ, а также реализовать данную БЗ с использованием существующих инструментальных средств разработки ЭС (CLIPS). Также работа позволит ознакомиться с тем, как функционально работают существующие оболочки ЭС для того, чтобы облегчить в дальнейшем работы по созданию собственной оболочки ЭС.

Вариативность задания заключается в различной предметной области:

1. Диагностика заболевания человека или животного.
 2. Прогнозирование получения студентом оценки на экзамене – какую оценку может получить студент при влиянии различных факторов (посещение занятий, строгость преподавателя, характеристики аудитории и т.д.).
 3. Выбор места для проведения отдыха (наличие средств, характеристики места прибытия и т.д.).
- И другие.

Последовательность выполнения:

1. Выбрать предметную область, в рамках которой планируется построить БЗ. Должна решаться задача, например, диагностики или прогнозирования.
2. Визуально оформить граф И/ИЛИ.
3. Описать правила БЗ в текстовом виде в нотации IF-THEN.
4. Ознакомиться с тестовыми примерами, которые идут совместно с поставкой среды CLIPS.
5. В инструменте разработки ЭС CLIPS реализовать сформированные правила.

Требования:

1. Граф И/ИЛИ должен иметь высоту не менее 4, т.е., как минимум, от одного факта до заключения должно быть 4 правила.
2. Количество правил должно быть не менее 15.
3. Скрипт в CLIPS должен соответствовать требованиям:
 - Часть фактов должно инициализироваться при запуске, остальные – в интерактивном режиме работы с пользователем.
 - Должны использоваться функции (deffunction).
 - Код должен быть единообразно отформатирован.
 - Должны присутствовать комментарии.
 - Должны присутствовать не только логические факты, но и целочисленные.

В результате выполнения лабораторной работы необходимо, чтобы были получены следующие результаты:

- 1) описание БЗ в нотации IF-THEN;
- 2) визуально оформленный граф И/ИЛИ;
- 3) скрипт правил в системе CLIPS.

Все результаты должны быть оформлены в виде отчета по лабораторной работе.

1.5. Лабораторная работа №2

Проектирование и реализация оболочки экспертной системы, использующей продукционный способ представления знаний

Цель: сформировать у студентов способность построения учебного прототипа экспертной системы, основанной на знаниях, включающего основные компоненты ИИС.

Описание: в рамках данной работы необходимо реализовать оболочку ЭС, которая была бы способна работать с продукционной БЗ. Работа позволит углубленно познакомиться со способами взаимодействия компонентов ЭС, а также реализовать отдельные функциональные блоки:

1. База знаний.
2. Механизм логического вывода.
3. Компонента объяснения.

Таким образом, будет создан минимальный прототип оболочки ЭС, позволяющий осуществить логический вывод. Вариативность задания может заключаться в различном способе хранения правил:

1. Хранение в формате XML.
2. Хранение в формате JSON.
3. Хранение в формате YAML.
4. Хранение в формате Protocol Buffers.

Последовательность выполнения:

1. Взять предметную область и БЗ из лабораторной работы №1.
2. В текстовом виде описать БЗ в выбранной нотации.
3. Спроектировать интерфейсы модулей реализуемой оболочки ЭС
4. Начать разработку в системе хранения контроля версий.

Требования:

1. Должно быть использовано несколько стратегий разрешения конфликтов
2. Каждая компонента оболочки ЭС должна быть представлена в виде отдельного модуля системы.
3. Программный код должен соответствовать требованиям:
 - код должен быть единообразно отформатирован;
 - должны присутствовать комментарии.
4. Компонента объяснения не должна быть сильно связана с МЛВ, т.е. должна присутствовать так называемая «Рабочая память».

В результате выполнения лабораторной работы необходимо, чтобы были получены следующие результаты:

- программный код, хранящийся в системе контроля версий.

Все результаты должны быть оформлены в виде отчета по лабораторной работе.

1.6. Лабораторная работа №3

Проектирование базы знаний и реализация ЭС на основе семантических сетей и фреймов

Цель: сформировать у студентов способность построения прототипа ЭС, использующего различные модели представления знаний.

Описание: в рамках данной работы необходимо спроектировать БЗ с представлением знаний в виде фреймов или семантических сетей, а также реализовать оболочку экспертной системы, работающую с данной БЗ. Основную часть программного кода необходимо взять из лабораторной работы №2. Оболочка должна состоять из следующих функциональных блоков:

1. БЗ – возможность хранения знаний.
2. МЛВ – возможность логического вывода.
3. Компонента объяснения – возможность объяснения логического вывода.

Вариативность задания заключается в различном способе представления знаний:

1. Нечетный номер варианта – семантические сети.
2. Четный номер варианта – фреймы.

Последовательность выполнения:

1. В сформированных группах распределить подсистемы между участниками.
2. Предметную область и описание БЗ необходимо взять из задания №1.
3. Спроектировать БЗ.
4. Начать разработку в системе контроля версий.

Требования:

1. Каждый функциональный блок должен быть представлен в виде отдельного модуля системы.
2. При использовании фреймового способа представления знаний иерархия фреймов должна содержать не менее 3 уров-

ней, а также некоторые фреймы должны содержать присоединенные процедуры.

3. При использовании семантической сети должна быть реализована поддержка не менее 3 типов запросов к БЗ, например, «является ли что-то чем-то» и т.п.
4. Программный код должен соответствовать требованиям:
 - код должен быть единообразно отформатирован;
 - должны присутствовать комментарии.

В результате выполнения лабораторной работы необходимо, чтобы были получены следующие результаты:

- 1) визуальное представление спроектированной семантической сети (или сети фреймов);
- 2) программный код, хранящийся в системе контроля версий.

Все результаты должны быть оформлены в виде отчета по лабораторной работе.

Экспертные системы поддержки принятия коллективных решений

Поддержка принятия решений является одной из распространенных интеллектуальных задач, суть которой заключается в формировании рекомендаций лицу, принимающему решение, позволяющих облегчить выбор наилучшего решения на заданном множестве альтернатив [4]. Знания экспертов в ЭС задаются с помощью одной или нескольких моделей, рассмотренных ниже и предполагающих формализацию предпочтений экспертов на множестве возможных решений.

2.1. Модели голосования при принятии коллективных решений

Начиная с политической философии Просвещения, выбор правил голосования являлся главной этической проблемой, связанной с далеко идущими приложениями для функционирования большинства политических институтов. Дебаты о справедливости различных методов голосования начались с исследований де Борда и Кондорсе. В 1952 году Эрроу предложил формальную модель, которая в течение трех десятилетий анализировалась в многочисленных работах математической ориентации по так называемому коллективному выбору [5, гл. 9].

Правило голосования решает задачу принятия коллективного решения: несколько агентов (выборщиков) осуществляют выбор из нескольких исходов (кандидатов).

Если кандидатов только два, то обычное правило голосования большинством голосов является наиболее адекватным методом. Однако если кандидатов больше двух, то возникает ряд проблем.

ЭКСПЕРТНЫЕ СИСТЕМЫ ПОДДЕРЖКИ ПРИНЯТИЯ КОЛЛЕКТИВНЫХ РЕШЕНИЙ

Рассмотрим пример обычного голосования людьми за одного кандидата (табл. 2.1).

Таблица 2.1

Пример голосования

№ альтернативы	1	2	3
Количество голосов «За»	50	40	10
Количество голосов «Очень против»	60	20	20

Получается, что победил кандидат №1 (за него большинство голосов), но он не подходит более чем половине выборщиков.

Для решения данной проблемы нужно учитывать предпочтения, т.е. каждый выборщик расставляет альтернативы по приоритету.

Наиболее популярные модели:

1. **Модель Кондорсе.** Победителем по Кондорсе называется кандидат a , который побеждает любого другого кандидата при попарном сравнении по правилу большинства (т.е. число выборщиков, считающих, что a лучше b , больше, чем число выборщиков, считающих, что b лучше a). Однако победитель по Кондорсе может не существовать (табл. 2.2).

Таблица 2.2

Профиль голосования с парадоксом по Кондорсе

8	7	6
a	b	c
b	c	a
c	a	b

Для определения победителя может применяться одно из возможных правил:

- **Правило Копленда.** Сравним кандидата a с любым другим кандидатом x . Начислим ему $+1$, если для большинства a лучше x , -1 , если для большинства

x лучше a , и 0 при равенстве. Суммируя общее количество очков по всем x , $x \neq a$, получаем оценку Копленда для a . Избирается кандидат с *наивысшей из таких оценок*.

- **Правило Симпсона.** Рассмотрим кандидата a , любого другого кандидата x и обозначим через $N(a, x)$ число выборщиков, для которых a лучше x . Оценкой Симпсона для a называется **минимальное** из чисел $N(a, x)$ по всем x , $x \neq a$. Избирается кандидат с *наивысшей из таких оценок*.

2. **Модель Борда.** Каждый выборщик объявляет свои предпочтения, ранжируя p кандидатов от лучшего к худшему (безразличие запрещается). Кандидат не получает очков за последнее место, получает $p - 1$ очков за первое место. Побеждает кандидат с *наибольшей суммой очков*.

2.2. Лабораторная работа №4

Реализация моделей принятия коллективных решений

Цель: сформировать у студентов способность построения экспертной системы поддержки принятия коллективных решений.

Описание: в рамках данной работы необходимо реализовать систему, включающую следующие модели принятия коллективных решений:

- 1) относительного большинства;
- 2) модель Кондорсе:
 - явный победитель,
 - правило Копленда,
 - правило Симпсона;
- 3) модель Борда;
- 4) линейная многокритериальная модель выбора [4, с. 16–18];
- 5) многокритериальная модель выбора с применением специальных нечетких множеств [4, с. 22–25].

Последние две модели (пункты 4 и 5) — модели повышенной сложности, реализация которых не является обязательной для выполнения задания.

Вариативность задания заключается в различной предметной области:

1. Выбор победителя на конкретных соревнованиях.
2. Выбор экспертами оптимального профиля ценных бумаг.
3. Выбор совместного места отдыха.
4. Выбор старосты группы,
и другие.

Требования:

1. Интерфейс должен быть максимально адаптирован к предметной области, т.е.:
 - результаты модели должны быть визуализированы в императивном виде: т.е. объяснять почему данный победитель был выбран;

- должна быть возможность голосования, используя графический интерфейс;
 - должна быть возможность использования «справочников» при добавлении вариантов для голосования. (Например, список видов ценных бумаг при добавлении конкретного кандидата).
2. Программный код должен соответствовать требованиям:
- код должен быть единообразно отформатирован;
 - должны присутствовать комментарии.

В результате выполнения лабораторной работы необходимо, чтобы были получены следующие результаты:

- 1) демонстрационный пример;
- 2) программный код, хранящийся в системе контроля версий.

Все результаты должны быть оформлены в виде отчета по лабораторной работе.

Нейронные сети

Традиционным подходом к построению механизмов рассуждения является использование дедуктивного логического вывода на правилах (rule-based reasoning), который применяется в экспертных системах продукционного и логического типа. При таком подходе необходимо заранее сформулировать весь набор закономерностей, описывающих предметную область. Альтернативный подход основан на концепции обучения по примерам (case-based reasoning). В этом случае при построении интеллектуальной системы не требуется заранее знать обо всех закономерностях исследуемой области, но необходимо располагать достаточным количеством примеров для настройки разрабатываемой адаптивной системы, которая после обучения будет способна получать требуемые результаты с определенной степенью достоверности. В качестве таких адаптивных систем применяются искусственные нейронные сети (ИНС) [1, гл. 5].

Серьезное развитие нейрокибернетики получила в трудах американского нейрофизиолога Ф. Розенблатта, который предложил свою модель нейронной сети в 1958 году и продемонстрировал созданное на ее основе электронное устройство, названное персептроном. Ф. Розенблатт ввел возможность модификации межнейронных связей, что сделало ИНС обучаемой. Первые персептроны были способны распознавать некоторые буквы латинского алфавита. Впоследствии модель персептрона была значительно усовершенствована, а наиболее удачным ее применением стали задачи автоматической классификации. Алгоритм обучения персептрона включает следующие шаги:

1. Системе предъявляется эталонный образ.
2. Если результат распознавания совпадает с заданным, весовые коэффициенты связей не изменяются.

3. Если ИНС неправильно распознает результат, то весовым коэффициентам дается приращение в сторону повышения качества распознавания.

3.1. Нейросетевые технологии распознавания образов

Нейросетевые технологии нашли широкое применение в таких направлениях, как распознавание печатного текста, контроль качества продукции на производстве, медицинские и военные приложения, управление и оптимизация, финансовый анализ, прогнозирование и др. В дальнейшем будет рассмотрено применение нейронных сетей в задачах распознавания образов с использованием архитектуры многослойного персептрона. Однако существуют архитектуры нейронных сетей, решающие задачу распознавания изображений наиболее эффективно, например сверточные нейронные сети. Данная архитектура ИНС была предложена Яном Лекуном, а также входит в состав технологий глубокого обучения (англ. *deep learning*).

Для того чтобы передать изображение в ИНС, нужно преобразовать его в матрицу размером $N \times N$, где каждый элемент матрицы имеет значение 1 или 0, исходя из контуров распознаваемого символа (рис. 3.1). А далее каждый пиксель изображения надо передать на вход сети (рис. 3.2).

Для того чтобы ИНС могла распознавать образы, ее нужно соответствующим образом обучить. Рассмотрим популярный метод обучения нейронной сети — метод обратного распространения ошибки [1, гл. 5.4.]. Данный метод является алгоритмом обучения «с учителем». Он является обобщением процедуры обучения простого персептрона с использованием дельта-правила на многослойные сети (алгоритм 1).

Для реализации нейронной сети можно воспользоваться принципами объектно-ориентированного программирования и выделить следующие классы:

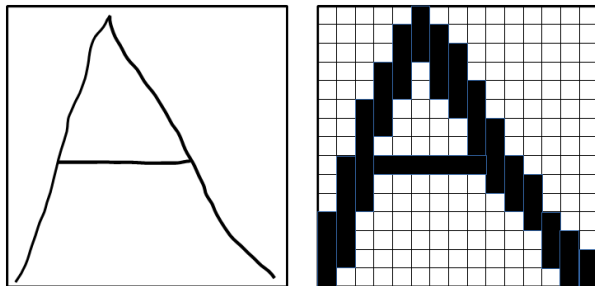


Рис. 3.1. Пример преобразования изображения для ИНС

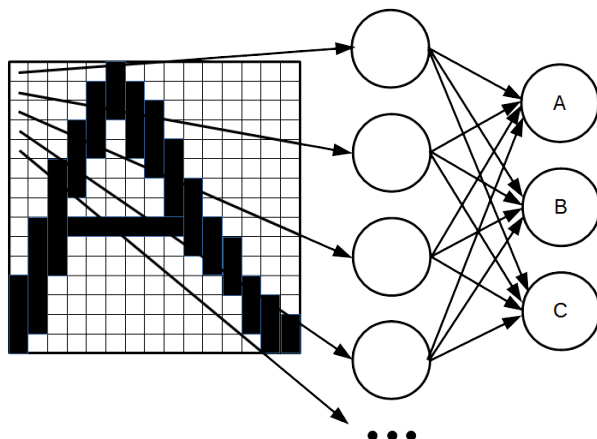


Рис. 3.2. Пример передачи изображения в ИНС

- 1) нейрон;
- 2) слой (с реализованным алгоритмом обратного распространения ошибки);
- 3) синапс – связь между слоями (так как рассматриваемые связи имеют вид «от каждого нейрона к каждому»).

Алгоритм 1. Алгоритм обратного распространения ошибки

- 1: Инициализировать веса связей малыми произвольными значениями;
 - 2: **Повторять**
 - 3: **Выполнить для всех** элементов обучающей выборки **выполнять**
 - 4: Подать образ на вход;
 - 5: // Прямое распространение.
 - 6: **Выполнить для всех** слоя в сети **выполнять**
 - 7: **Выполнить для всех** нейронов в слое **выполнять**
 - 8: Вычислить сумму произведения весов на значение входных сигналов;
 - 9: Добавить смещение к сумме;
 - 10: Вычислить значение функции активации;
 - 11: **Конец цикла**
 - 12: **Конец цикла**
 - 13: // Обратное распространение ошибки выходного слоя.
 - 14: **Выполнить для всех** нейронов в выходном слое **выполнять**
 - 15: Вычислить ошибку сигнала нейрона;
 - 16: **Конец цикла**
 - 17: // Обратное распространение ошибки скрытых слоев.
 - 18: **Выполнить для всех** скрытого слоя **выполнять**
 - 19: **Выполнить для всех** нейронов в скрытом слое **выполнять**
 - 20: Вычислить ошибку сигнала нейрона;
 - 21: Обновить вес связи;
 - 22: **Конец цикла**
 - 23: **Конец цикла**
 - 24: Вычислить глобальную ошибку;
 - 25: **Конец цикла**
 - 26: **Пока выполняется** (Количество итераций < Максимум итераций)
 И (Ошибка > Указанная точность)
-

3.2. Лабораторная работа №5

Реализация нейронных сетей для решения задачи распознавания образов

Цель: сформировать у студентов способность построения нейросетевой интеллектуальной системы для решения задач распознавания образов.

Описание: в рамках данной работы необходимо реализовать приложение, решающее задачу распознавание образов с использованием искусственных нейронных сетей. Возможный вариант реализации – многослойный персептрон с методом обратного распространения ошибки, где каждый входной сигнал соответствует пикселю образа.

Вариативность задания заключается в различной предметной области:

- 1) распознавание геометрических фигур (не менее 8);
- 2) распознавание знаков валют (не менее 8);
- 3) распознавание направления стрелки, указывающей на 8 направлений сторон света (север, северо-восток и т.д.);
- 4) распознавание римских цифр.

Требования:

1. Должна быть возможность добавления, удаления, изменения элементов обучающей выборки.
2. Должна быть возможность рисования распознаваемого образа мышью компьютера.
3. Не допускается непосредственное копирование кода из открытых библиотек, реализующих алгоритмы с искусственными нейронными сетями.
4. Программный код должен соответствовать требованиям:
 - код должен быть единообразно отформатирован;
 - должны присутствовать комментарии.

В результате выполнения лабораторной работы необходимо, чтобы были получены следующие результаты:

- 1) визуальное представление архитектуры нейронной сети;
- 2) программный код, хранящийся в системе контроля версий.

Все результаты должны быть оформлены в виде отчета по лабораторной работе.

Список литературы

1. Андрейчиков А.В., Андрейчикова О.Н. Интеллектуальные информационные системы : учебник. — М.: Финансы и статистика, 2004. — 424 с.
2. Ясницкий Л.Н. Интеллектуальные системы : учебник. — М.: Лаборатория знаний, 2016. — 221 с.
3. Джарратано Д., Райли Г. Экспертные системы: принципы разработки и программирование : пер. с англ. — 4-е изд. — М.: Вильямс, 2007. — 1152 с.
4. Гитман М.Б, Столбов В.Ю. Экспертные системы поддержки принятия коллективных решений : учеб. пособие. — Пермь: Изд-во Перм. нац. исслед. политехн. ун-та, 2017. — 38 с.
5. Мулен Э. Кооперативное принятие решений: аксиомы и модели. — М.: Мир, 1991. — 464 с.
6. Минский М. Фреймы для представления знаний: пер. с англ. — М.: Энергия, 1979. — 152 с.
7. Кормен Т. и др. Алгоритмы: построение и анализ: пер. с англ. — 3-е изд. — М.: Вильямс, 2013. — 1328 с.
8. Roberts R.B., Goldstein I.P. The FRL Primer. — Massachusetts Institute of Technology, Cambridge, MA, 1977. — № AI-M-408.

Учебное издание

Истомин Денис Андреевич
Столбов Валерий Юрьевич

ИНТЕЛЛЕКТУАЛЬНЫЕ СИСТЕМЫ И ТЕХНОЛОГИИ

Лабораторный практикум

Редактор и корректор *И.Н. Жеганина*

Подписано в печать 30.10.17. Формат 60х90/16.
Усл. печ. л. 2,5. Тираж 100 экз. Заказ № 264/2017.

Издательство
Пермского национального исследовательского
политехнического университета.
Адрес: 614990, г. Пермь, Комсомольский пр., 29, к. 113
Тел. (342) 219-80-33.