

**Министерство науки и высшего образования РФ
ФГБОУ ВО «Ульяновский государственный университет»
Факультет математики, информационных и авиационных технологий**

Кафедра телекоммуникационных технологий и сетей

Липатова Светлана Валерьевна

МЕТОДИЧЕСКИЕ РЕКОМЕНДАЦИИ

для семинарских (практических) занятий, лабораторного практикума
и самостоятельной работы
по дисциплинам

«Экспертные системы» и «Представление знаний»

для студентов направлений

09.03.02 «Информационные системы и технологии»,

11.03.02 «Инфокоммуникационные технологии и системы»,

09.03.03 « Прикладная информатика»,

*02.03.03 «Математическое обеспечение и администрирование
информационных систем»*



Ульяновск
2022

Методические рекомендации для семинарских (практических) занятий, лабораторного практикума и самостоятельной работы по дисциплинам «Экспертные системы» и «Представление знаний» / составитель: С.В. Липатова - Ульяновск: УлГУ, 2019 – 78 с.

Настоящие методические рекомендации предназначены для студентов направлений обучения 09.03.02 «Информационные системы и технологии», 11.03.02 «Инфокоммуникационные технологии и системы», 09.03.03 « Прикладная информатика», 02.03.03 «Математическое обеспечение и администрирование информационных систем». В работе приведены литература по дисциплине, темы дисциплины и вопросы в рамках каждой темы, рекомендации по изучению теоретического материала, контрольные вопросы для самоконтроля, задания для самостоятельной работы, задачи и упражнения для самостоятельной подготовки к семинарам или полностью самостоятельного освоения практических навыков, задания для лабораторного практикума и рекомендации по их выполнению.

Студентам всех форм обучения следует использовать данные методические рекомендации при подготовке к семинарам, самостоятельной подготовке, а также промежуточной аттестации по дисциплинам «Экспертные системы» и «представление знаний».

Рекомендованы к введению в образовательный процесс

Учёным советом факультета математики, информационных и авиационных технологий
УлГУ

протокол № 3/22 от «19» апреля 2022 г.

СОДЕРЖАНИЕ

ОБЩИЕ ВОПРОСЫ	5
РЕКОМЕНДАЦИИ ПО ОТДЕЛЬНЫМ ТЕМАМ ДИСЦИПЛИНЫ	7
<i>Тема 1. Экспертные системы</i>	7
Основные вопросы темы	7
Рекомендации по изучению темы	7
Вопросы для самоподготовки	7
Контрольные тесты	7
<i>Тема 2. Методы извлечения и структурирования знаний</i>	8
Основные вопросы темы	8
Рекомендации по изучению темы	8
Вопросы для самоподготовки	9
Задачи для практических занятий и самоподготовки	9
<i>Тема 3: Методы формализации (представления и кодирования) знаний</i>	10
Основные вопросы темы	10
Рекомендации по изучению темы	10
Вопросы для самоподготовки	11
Контрольные тесты	11
Задачи для практических занятий и самоподготовки	13
<i>Тема 4: Методы вывода на основе знаний</i>	28
Основные вопросы темы	28
Рекомендации по изучению темы	28
Вопросы для самоподготовки	28
Контрольные тесты	28
Задачи для практических занятий и самоподготовки	29
ЛАБОРАТОРНЫЙ ПРАКТИКУМ	45
<i>Тема 2: Методы извлечения и структурирования знаний</i>	45
Задание лабораторной работы	45

Методические указания по выполнению лабораторной работы	45
Варианты для выполнения лабораторной работы.....	48
<i>Тема 3: Методы формализации (представления и кодирования) знаний</i>	<i>49</i>
Задание лабораторной работы	49
Методические указания по выполнению лабораторной работы	49
Варианты для выполнения лабораторной работы.....	57
<i>Темы 1, 4: Методы вывода на основе знаний Экспертные системы</i>	<i>57</i>
Задание лабораторной работы SWI-Prolog.....	57
Методические указания по выполнению лабораторной работы в SWI-Prolog.....	57
Варианты для выполнения лабораторной работы.....	61
<i>Темы 1, 4: Методы вывода на основе знаний Экспертные системы</i>	<i>63</i>
Задание лабораторной работы по рекомендательным системам.....	63
Методические указания по использованию Surprise для построения рекомендательных систем.....	64
Варианты для выполнения лабораторной работы.....	73
РЕКОМЕНДУЕМАЯ ЛИТЕРАТУРА И ИНФОРМАЦИОННОЕ ОБЕСПЕЧЕНИЕ	77
Список рекомендуемой литературы	77
Программное обеспечение	78

ОБЩИЕ ВОПРОСЫ

В результате изучения дисциплин «Экспертные системы» и «Представление знаний» студенты должны:

знать:

- о нечеткости знаний, ее природе и разновидностях,
- основные модели нейронных сетей, методы и алгоритмы их обучения,
- проблемах и основных методах представления и обработки знаний,
- о подходах к построению интеллектуальных систем – логическом и нейрокибернетическом, эволюционном,
- этапы построения экспертных систем,
- языках программирования искусственного интеллекта;
- о структуру экспертных систем и их архитектурные особенности в зависимости от особенностей решаемой задачи,

уметь:

- ориентироваться в различных типах интеллектуальных систем,
- ориентироваться в различных методах представления знаний,
- ставить задачу построения экспертной системы для решения задачи выбора вариантов в плохо формализуемой предметной области,

владеть:

- методами представления и обработки знаний,
- навыками формализации знаний экспертов с применением различных методов представления знаний,
- навыками разработки продукционные базы знаний для решения задач задачи выбора вариантов в плохо формализуемой предметной области,
- навыками логического программирования.

Методические рекомендации для семинарских (практических) занятий, лабораторного практикума и самостоятельной работы по дисциплинам «Экспертные системы» и «Представление знаний» направлены на повышение эффективности освоения знаний, умений, навыков и компетенций, связанных с:

- извлечением, структурированием и формализацией знаний,
- построением баз знаний,
- разработкой экспертных систем,
- разработкой интеллектуальных систем и др.

Методические рекомендации предлагают указания по всем темам дисциплин «Экспертные системы» и «Представление знаний». Методические рекомендации разбиты по темам и содержат набор вопросов для систематизации теоретического материала, полученного на лекционных занятиях, и самостоятельного изучения теории, вопросы (тесты) для текущего контроля на практических занятиях (семинарах), задачи для усвоения практических навыков. Для лабораторного практикума приведены задания, варианты и рекомендации по выполнению лабораторных работ.

Список литературы и информационного обеспечения, приведённый в конце методических указаний, может служить основой для изучения всех рассматриваемых тем. Дополнительная и учебно-методическая литература могут быть использованы обучающимися для закрепления изучаемого материала.

РЕКОМЕНДАЦИИ ПО ОТДЕЛЬНЫМ ТЕМАМ ДИСЦИПЛИНЫ

Тема 1. Экспертные системы

Основные вопросы темы

1. Классические экспертные системы: META-DENDRAL, MYCIN-EMYCIN-TEIREIAS-PUFF-NEOMYCIN, PROSPECTOR-KAS. Современные экспертные системы: G2 (Gensym, США), RTWorks (Talarian, США), COMDALE/C (Comdale Techn, Канада), COGSYS (SC, США), ILOG Rules (ILOG, Франция), Watson (IBM, США). Зарубежные и российские исследователи в области представления знаний и построения экспертных систем.
2. Понятие экспертной системы, её структура, классификация экспертных систем, назначение, основные типы пользователей. Системы для разработки экспертных систем, оболочки экспертных систем. Этапы прототипирования экспертной системы.

Рекомендации по изучению темы

Вопрос 1 изложен в учебнике [3] на с. 94-97.

Вопрос 2 изложен в учебнике [3] на с. 97-105 и [1] в главах 1,2.

Дополнительные материалы представлены в [2] и в соответствующем разделе лабораторного практикума.

Вопросы для самоподготовки

Рекомендуется после изучения материалов лекций и специальной литературы подготовить ответы на вопросы:

1. Что такое экспертная система?
2. Как можно классифицировать экспертные системы?
3. Из каких блоков состоит экспертная система?
4. Какие этапы создания экспертных систем существуют?
5. Приведите примеры экспертных систем?

Контрольные тесты

1. **Как называлась одна из первых медицинских экспертных систем?**
 - a) MACSYMA
 - b) MYCIN
 - c) PROSPECTOR
 - d) нет правильного ответа

2. Какую задачу решала экспертная система PROSPECTOR?

- a) определение наиболее вероятной структуры химического соединения
- b) поиск месторождений на основе геологических анализов
- c) диагностика глазных заболеваний
- d) распознавание слитной человеческой речи
- e) нет правильного ответа

3. Какие подсистемы являются для экспертной системы обязательными?

- a) база знаний
- b) интерфейс системы с внешним миром
- c) алгоритмические методы решений
- d) интерфейс когнитолога
- e) контекст предметной области

4. Какая экспертная система имеет базу знаний размером от 1000 до 10000 структурированных правил?

- a) простая
- b) средняя
- c) сложная

Тема 2. Методы извлечения и структурирования знаний

Основные вопросы темы

1. Источники получения знаний. Процесс извлечения, основные проблемы извлечения знаний. Аспекты извлечения знаний. Типы экспертов. Методы: текстологические и коммуникационные (активные и пассивные), - их классификация, область применения, достоинства и недостатки.
2. Поле знаний. Процесс структурирования знаний. Подходы к структурированию знаний: структурный, объектный, объектно-структурный, онтологический. Методы структурирования знаний, их классификация, область применения, достоинства и недостатки.

Рекомендации по изучению темы

Вопрос 1 изложен в учебнике [1] на параграфе 1.3 .

Вопрос 2 изложен в учебнике [2] в разделе 2.

Дополнительные материалы представлены в [5] и в соответствующем разделе лабораторного практикума.

Вопросы для самоподготовки

Рекомендуется после изучения материалов лекций и специальной литературы подготовить ответы на вопросы:

1. В чем заключается метод «мозговой штурм»?
2. В чем заключается метод «круглый стол»?
3. В чем заключается метод «Дельфи»?
4. В чем заключается метод анализа иерархий?
5. Основные принципы SWOT-анализа?
6. Что такое ментальная карта?
7. Достоинства и недостатки диаграммы Ганта?
8. Что позволяет отображать диаграмма Исикавы?

Задачи для практических занятий и самоподготовки

Задание: проведите подготовку для использования одного из методов извлечения экспертных знаний.

	Метод	Задача	Выполнить
1)	анкетирование	Подарок на 8 марта	1) Сформировать анкеты (со всеми видами вопросов) (анкета должна учитывать разные категории объектов, имеющих отношение к задаче и результат, должен быть практически применим). 2) Повести анкетирование. 3) Провести анализ заполненных анкет и представить результат в наглядном виде.
2)		Подарок на Новый год	
3)		Лучшая компьютерная игра	
4)		Лучшая мобильная игра	
5)		Лучший фильм	
6)		Лучшая книга	
7)	интервьюирование	План на будущие 5 лет	1) Подготовка примерных вопросов для интервью. 2) Проведение интервью. 3) Обработка результатов интервью и представление их в наглядном виде.
8)		Какие бы изменения внести в организацию учебного процесса в ВУЗе	
9)		Какие навыки в сфере IT считаете наиболее востребованными	

Задание: Используйте методы предметной области для структурирования и постройте схему в программном продукте Edraw (или его аналоги).

	Метод	Задача
1)	SWOT-анализ	для открытия своего дела (например стартап в области электронной коммерции)
2)		для занятия должности менеджера по продажам в магазине по продаже электронных товаров
3)		для занятия должности ведущего программиста в web-компании
4)		для занятия должности программиста на промышленном предприятии
5)	диаграмма Ганта	Нарисуйте диаграмму Ганта по оставшемуся семестру вашего обучения в ВУЗе
6)		Нарисуйте диаграмму Ганта на пятилетний карьерный план
7)	диаграмма «рыбий скелет» Исикавы (схема причинно-следственных связей)	Постройте причинно-следственную диаграмму для качества вашей дипломной работы
8)		Постройте причинно-следственную диаграмму для качества программного продукта
9)		Постройте причинно-следственную диаграмму для качества программной документации

Тема 3: Методы формализации (представления и кодирования) знаний

Основные вопросы темы

1. Модели представления знаний, определение, классификация, достоинства и недостатки.
2. Продукционная, сетевая, фреймовая модели.

Рекомендации по изучению темы

Вопросы 1,2 изложен в учебнике [3] на с. 34-48 и [1] в параграфе 1.4, [4] в главах 4, 5,6, 10,11, 12.

Вопрос 2 изложен в учебнике [4] в главах 4,7,8, 9.

Дополнительные материалы представлены в [5] и в соответствующем разделе лабораторного практикума.

Вопросы для самоподготовки

Рекомендуется после изучения материалов лекций и специальной литературы подготовить ответы на вопросы:

1. Достоинства и недостатки детерминированных моделей представления знаний?
2. Каковы возможности использования нейронных сетей для построения экспертных систем?
3. Как можно применять нечёткие модели при построении баз знаний?
4. Что из себя представляют гибридные базы знаний?

Контрольные тесты

- 1. Что понимается под представлением знаний?**
 - a) кодирование информации на каком-либо формальном языке
 - b) знания, представленные в программе на языке C++
 - c) знания, представленные в учебниках по математике
 - d) моделирование знаний специалистов-экспертов
- 2. Какие определения, представленные ниже, не являются моделями представления знаний?**
 - a) продукционные модели
 - b) фреймы
 - c) имитационные модели
 - d) семантические сети
 - e) формально-логические модели
- 3. Что представляет собой семантическая сеть?**
 - a) сетевой график, вершины которого – сроки выполнения работ
 - b) нейронная сеть, состоящая из нейронов
 - c) ориентированный граф, вершины которого – понятия, а дуги – отношения между ними
- 4. Какой из основных типов отношений семантической сети, представленных ниже, может быть назван как АКО (A - Kind - Of)?**
 - a) это
 - b) элемент класса
 - c) имеет частью
 - d) принадлежит
 - e) функциональная связь
- 5. Чем отличаются семантические сети и фреймы?**

- a) элемент модели состоит из множества незаполненных значений некоторых атрибутов, именуемых «слотами»
- b) наследование по АКО-связям
- c) элемент модели – структура, используемая для обозначения объектов и понятий

6. Что объединяет семантические сети и фреймы?

- a) организация процедуры вывода
- b) наследование свойств
- c) множества незаполненных значений некоторых атрибутов, именуемых слотами
- d) структуры, используемые для обозначения объектов и понятий

7. Какие из выражений, представленных ниже, являются структурной частью фрейма?

- a) значение N-го слота
- b) шаблон
- c) примитивные типы данных

8. На каком формализме не основаны логические модели?

- a) исчисление высказываний
- b) пропозициональная логика
- c) силлогизмы Аристотеля
- d) правильно построенные формулы
- e) нечёткие системы (fuzzy set)

9. Что такое факт?

- a) это логическая модель знаний;
- b) это утверждение общего характера;
- c) это утверждение правила;
- d) это частное утверждение;
- e) нет правильного ответа.

10. Что такое правило?

- a) это утверждение факта;
- b) это частное утверждение;
- c) это утверждение общего характера;
- d) это логическая модель знаний;
- e) нет правильного ответа.

11. Что такое база знаний?

- a) это компьютерная модель знаний специалиста в определённой предметной области;
- b) это компьютерная модель логических рассуждений специалиста в определённой предметной области;
- c) это компьютерная модель фактов;
- d) это компьютерная модель правил;
- e) все ответы правильные.

12. Что такое механизм вывода?

- a) нет правильного ответа;
- b) это модель алгоритма вывода ответов на экран монитора;
- c) это вывод ответов на внешние запоминающие устройства компьютера;
- d) это модель алгоритма создания ответов ;
- e) это модель логических рассуждений, на основе базы знаний.

Задачи для практических занятий и самоподготовки

Задача 1. Построить продукционную модель представления знаний в предметной области «Ресторан» (посещение ресторана).

Описание процесса решения. Для построения продукционной модели представления знаний необходимо выполнить следующие шаги:

- 1) Определить целевые действия задачи (являющиеся решениями).
- 2) Определить промежуточные действия или цепочку действий, между начальным состоянием и конечным (между тем, что имеется, и целевым действием).
- 3) Определять условия для каждого действия, при котором его целесообразно и возможно выполнить. Определить порядок выполнения действий.
- 4) Добавить конкретики при необходимости, исходя из поставленной задачи.
- 5) Преобразовать полученный порядок действий и соответствующие им условия в продукции.
- 6) Для проверки правильности построения продукции записать цепочки продукции, явно проследив связи между ними.

Этот набор шагов предполагает движение при построении продукционной модели от результата к начальному состоянию, но возможно и движение от начального состояния к результату (шаги 1 и 2).

Решение.

- 1) Обязательное действие, выполняемое в ресторанах – поглощение пищи и ее оплата. Значит, есть уже два целевых действия «съесть пищу» и «оплатить», которые взаимосвязаны и следуют друг за другом.
- 2) Прежде чем что-либо съесть в ресторане, туда нужно придти, дождаться официанта и сделать заказ. Кроме того, нужно выбрать, в какой именно ресторан пойти. Значит, цепочка промежуточных действий: «выбор ресторана и путь туда», «сделать заказ официанту».
- 3) Прежде чем идти в ресторан, необходимо убедиться, что есть необходимая сумма денег. Выбор ресторана может обуславливаться многими причинами, выберем территориальный признак – к какому ближе в тот и идем. В разных ресторанах работают разные люди, поэтому в зависимости от выбора ресторана, официанты будут разные. Кроме того, разные рестораны специализируются на разных кухнях, поэтому заказанные блюда будут в разных ресторанах отличаться. Значит вначале идут действия, позволяющие выбрать ресторан, затем характеризующие рестораны, а уже после заказ, еда, и оплата заказа.
- 4) Пусть в задаче будут рассматриваться два ресторана: «Вкусная еда» и «Вкуснятина». Первый – паб и заказы приносят быстрее, чем во втором, второй – пиццерия. В первом работает официант Сергей, а во втором официантка Марина. Петр – это клиент.
- 5) Выше описанное можно преобразовать в следующие предложения типа «Если, то»:
 - *Если субъект хочет есть и у субъекта есть достаточная сумма денег, то субъект может пойти в ресторан.*
 - *Если субъект ближе к ресторану «Вкусная еда», чем к ресторану «Вкуснятина» и субъект может пойти в ресторан, то субъект идет в ресторан «Вкусная еда».*
 - *Если субъект ближе к ресторану «Вкуснятина», чем к ресторану «Вкусная еда» и субъект может пойти в ресторан, то субъект идет в ресторан «Вкуснятина».*
 - *Если субъект идет в ресторан «Вкуснятина» и в ресторане «Вкуснятина» работает официант Марина, то у субъекта принимает заказ Марина.*
 - *Если субъект идет в ресторан «Вкусная еда» и в ресторане «Вкусная еда» работает официант Сергей, то у субъекта принимает заказ Сергей.*
 - *Если субъект выбрал блюда и у субъекта принимает заказ Марина, то заказ принесут через 20 мин.*

- Если субъект выбрал блюда и у субъекта принимает заказ Сергей, то заказ принесут через 10 мин.
- Если заказ принесут через 20 мин. или заказ принесут через 10 мин., то субъект может есть.
- Если субъект может есть, то после еды субъект должен оплатить заказ.

Введем обозначения для фактов (Ф), действий (Д) и продукций (П), тогда:

Субъект = Петр;

Ф1= субъект хочет есть;

Ф2= у субъекта есть достаточная сумма денег;

Ф3= субъект ближе к ресторану «Вкусная еда», чем к «Вкуснятина»;

Ф4=в ресторане «Вкуснятина» работает официант Марина;

Ф5=в ресторане «Вкусная еда» работает официант Сергей;

Ф6= субъект выбрал блюда;

Д1= субъект может пойти в ресторан;

Д2=субъект идет в ресторан «Вкусная еда»;

Д3=субъект идет в ресторан «Вкуснятина»;

Д4= у субъекта принимает заказ Марина;

Д5=у субъекта принимает заказ Сергей;

Д6=заказ принесут через 20 мин.

Д7=заказ принесут через 10 мин.

Д8=после еды субъект должен оплатить заказ.

Для продукций установим приоритет (в скобках перед запятой, чем выше приоритет, чем раньше проверяется правило).

П1(4 , Ф1 и Ф2)= Д1;

П5(3 , Д2 и Ф5)= Д5;

П2(5 , Ф3 и Д1)= Д2;

П6(2 , Д4)= Д6;

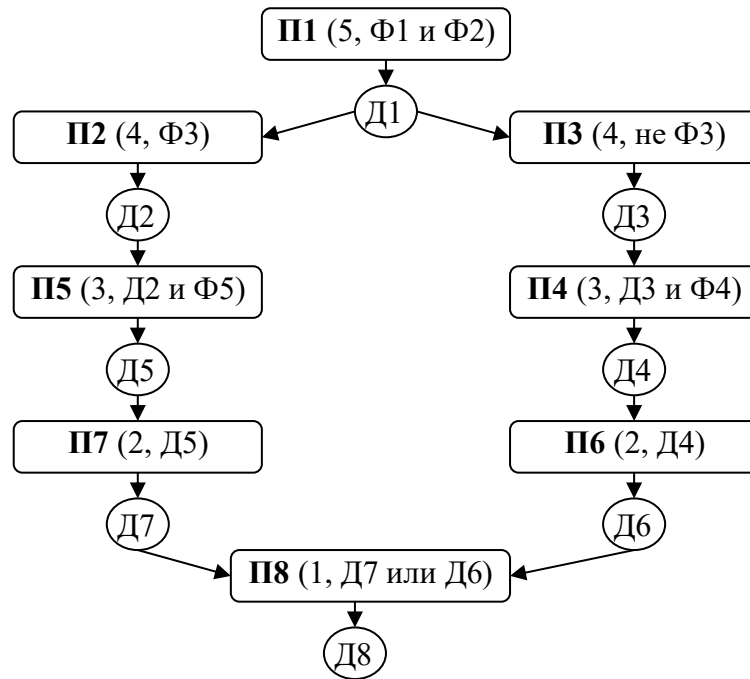
П3(4 , не Ф3 и Д1)= Д3;

П7(2 , Д5)= Д7;

П4(3 , Д3 и Ф4)= Д4;

П8(1 , Д6 или Д7)= Д8;

- б) Для отображения взаимосвязи продукций построим граф (рис. 1).



Варианты задачи 1

1. Построить продукционную модель представления знаний в предметной области «Аэропорт» (диспетчерская).
2. Построить продукционную модель представления знаний в предметной области «Железная дорога» (продажа билетов).
3. Построить продукционную модель представления знаний в предметной области «Торговый центр» (организация).
4. Построить продукционную модель представления знаний в предметной области «Автозаправка» (обслуживание клиентов).
5. Построить продукционную модель представления знаний в предметной области «Автопарк» (пассажирские перевозки).
6. Построить продукционную модель представления знаний в предметной области «Компьютерные сети» (организация).
7. Построить продукционную модель представления знаний в предметной области «Университет» (учебный процесс).
8. Построить продукционную модель представления знаний в предметной области «Компьютерная безопасность» (средства и способы ее обеспечения).
9. Построить продукционную модель представления знаний в предметной области «Компьютерная безопасность» (угрозы).
10. Построить продукционную модель представления знаний в предметной области «Интернет-кафе» (организация и обслуживание).

11. Построить продукционную модель представления знаний в предметной области «Разработка информационных систем» (ведение информационного проекта).
12. Построить продукционную модель представления знаний в предметной области «Туристическое агентство» (работа с клиентами).
13. Построить продукционную модель представления знаний в предметной области «Зоопарк» (организация).
14. Построить продукционную модель представления знаний в предметной области «Кухня» (приготовление пищи).
15. Построить продукционную модель представления знаний в предметной области «Больница» (прием больных).
16. Построить продукционную модель представления знаний в предметной области «Кинопрокат» (ассортимент и работа с клиентами).
17. Построить продукционную модель представления знаний в предметной области «Прокат автомобилей» (ассортимент и работа с клиентами).
18. Построить продукционную модель представления знаний в предметной области «Операционные системы» (функционирование).
19. Построить продукционную модель представления знаний в предметной области «Информационные системы» (виды и функционирование).
20. Построить продукционную модель представления знаний в предметной области «Предприятие» (структура и функционирование).

Задача 2. Построить сетевую модель представления знаний в предметной области «Ресторан» (посещение ресторана).

Описание процесса решения. Для построения сетевой модели представления знаний необходимо выполнить следующие шаги:

- 1) Определить абстрактные объекты и понятия предметной области, необходимые для решения поставленной задачи. Оформить их в виде вершин.
- 2) Задать свойства для выделенных вершин, оформив их в виде вершин, связанных с исходными вершинами атрибутивными отношениями.
- 3) Задать связи между этими вершинами, используя функциональные, пространственные, количественные, логические, временные, атрибутивные отношения, а также отношения типа «являться наследником» и «являться частью».
- 4) Добавить конкретные объекты и понятия, описывающие решаемую задачу. Оформить их в виде вершин, связанных с уже существующими отношениями типа «являться экземпляром», «есть».

- 5) Проверить правильность установленных отношений (вершины и само отношение при правильном построении образуют предложение, например «Двигатель является частью автомобиля»).

Решение.

- 1) Ключевые понятия данной предметной области – ресторан, тот, кто посещает ресторан (клиент) и те, кто его обслуживают (повара, метрдотели, официанты, для простоты ограничимся только официантами). У обслуживающего персонала и клиентов есть общие характеристики, поэтому целесообразно выделить общее абстрактное понятие – человек. Продукцией ресторана являются блюда, которые заказывают клиенты.

Исходя из этого, вершины графа будут следующими: «Ресторан», «Человек», «Официант», «Клиент», «Заказ» и «Блюдо».

- 2) У этих объектов есть определенные свойства и атрибуты. Например, рестораны располагаются по определенным адресам, каждое блюдо из меню имеет свою цену. Поэтому добавим вершины «Адрес» и «Цена».

- 3) Определим для имеющихся вершин отношения и их типы, используя таблицу 2.

- 4) Добавим знание о конкретных фактах решаемой задачи. Пусть имеется два ресторана: «Вкуснятина» и «Вкусная еда», в первом работает официантка Марина, а во втором официант Сергей. Пётр решил пойти в ресторан «Вкусная еда» и сделал заказ официанту на 2 блюда: картофель фри за 30 р., бифштекс за 130 р. Также известны адреса этих ресторанов и их специфика.

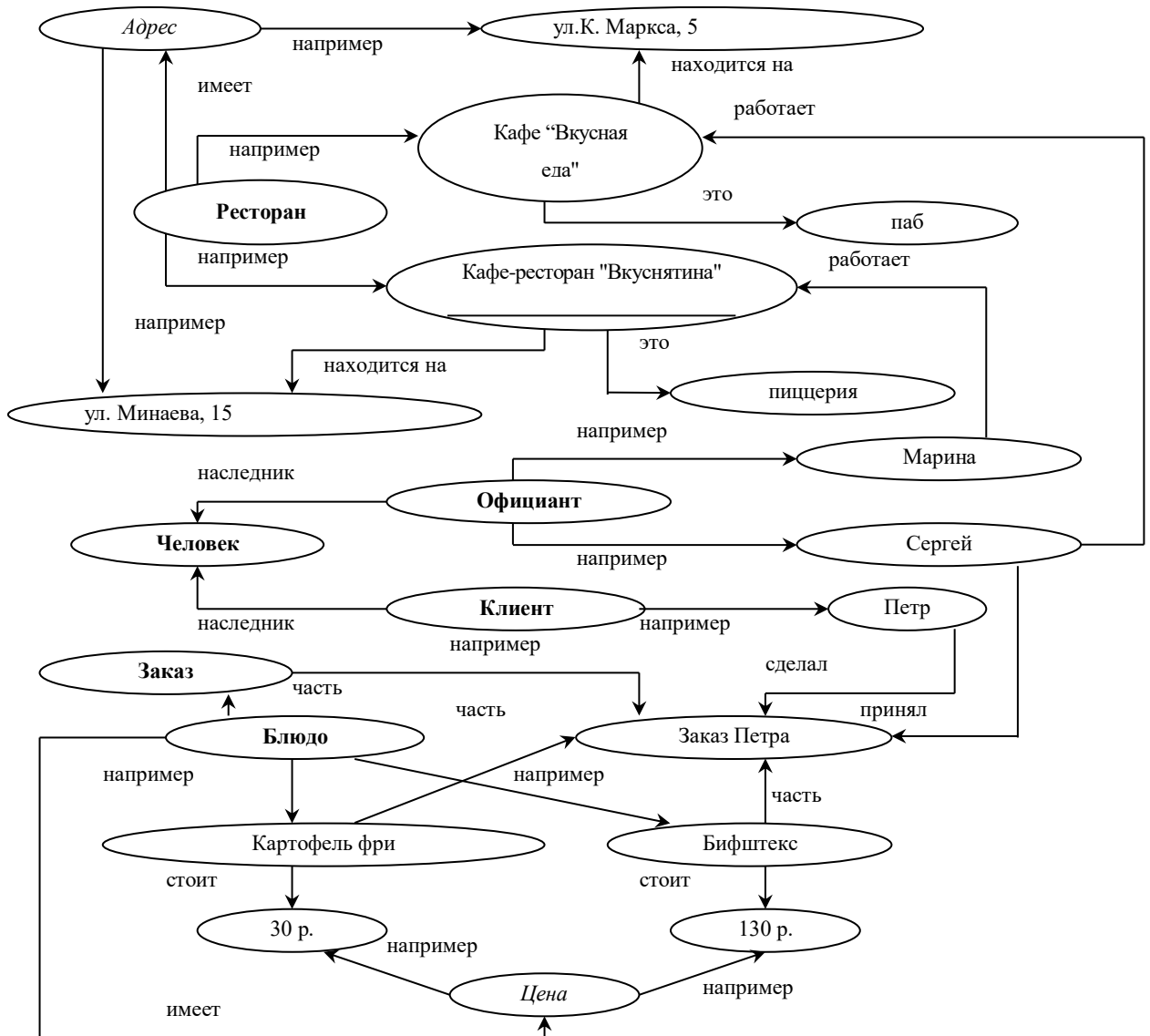
Исходя из этого, добавим соответствующие вершины в граф и соединим их функциональными отношениями и отношениями типа «например или являются экземпляром». Полученный в результате граф изображен на рис. 2.

- 5) Осуществим проверку установленных связей. Например, возьмем вершину «Блюдо» и пройдем по установленным связям. Получаем следующую информацию: блюдо является частью заказа, примерами блюд могут служить картофель фри и бифштекс.

Для получения ответа на какой-либо вопрос по этой задаче, необходимо найти соответствующий участок сети и, используя связи, получить результат.

Например, вопрос «Какова цена заказа Петра (сколько Петр заплатил за заказ)?» Из запроса понятно, что необходимо найти следующие вершины: «Цена», «Петр» и «Заказ» или «Заказ Петра». Часть семантической сети, находящаяся между этими вершинами, содержит ответ, а именно, частью заказа Петра являются картофель фри и

бифштекс, которые стоят 30 и 130 р. соответственно. Больше информации о заказе Петра в модели нет, поэтому делаем вывод – Петр заплатил 160 р.



Семантическая сеть предметной области «Ресторан».

Варианты задачи 2

1. Построить семантическую модель (сеть) представления знаний в предметной области «Аэропорт» (диспетчерская).
2. Построить семантическую модель (сеть) представления знаний в предметной области «Железная дорога» (продажа билетов).
3. Построить семантическую модель (сеть) представления знаний в предметной области «Торговый центр» (организация).
4. Построить семантическую модель (сеть) представления знаний в предметной области «Автозаправка» (обслуживание клиентов).

5. Построить семантическую модель (сеть) представления знаний в предметной области «Автопарк» (пассажирские перевозки).
6. Построить семантическую модель (сеть) представления знаний в предметной области «Компьютерные сети» (организация).
7. Построить семантическую модель (сеть) представления знаний в предметной области «Университет» (учебный процесс).
8. Построить семантическую модель (сеть) представления знаний в предметной области «Компьютерная безопасность» (средства и способы ее обеспечения).
9. Построить семантическую модель (сеть) представления знаний в предметной области «Компьютерная безопасность» (угрозы).
10. Построить семантическую модель (сеть) представления знаний в предметной области «Интернет-кафе» (организация и обслуживание).
11. Построить семантическую модель (сеть) представления знаний в предметной области «Разработка информационных систем» (ведение информационного проекта).
12. Построить семантическую модель (сеть) представления знаний в предметной области «Туристическое агентство» (работа с клиентами).
13. Построить семантическую модель (сеть) представления знаний в предметной области «Зоопарк» (организация).
14. Построить семантическую модель (сеть) представления знаний в предметной области «Кухня» (приготовление пищи).
15. Построить семантическую модель (сеть) представления знаний в предметной области «Больница» (прием больных).
16. Построить семантическую модель (сеть) представления знаний в предметной области «Кинопрокат» (ассортимент и работа с клиентами).
17. Построить семантическую модель (сеть) представления знаний в предметной области «Прокат автомобилей» (ассортимент и работа с клиентами).
18. Построить семантическую модель (сеть) представления знаний в предметной области «Операционные системы» (функционирование).
19. Построить семантическую модель (сеть) представления знаний в предметной области «Информационные системы» (виды и функционирование).
20. Построить семантическую модель (сеть) представления знаний в предметной области «Предприятие» (структура и функционирование).

Задача 3. Построить фреймовую модель представления знаний в предметной области «Ресторан» (посещение ресторана).

Описание процесса решения. Для построения фреймовой модели представления знаний необходимо выполнить следующие шаги:

- 1) Определить абстрактные объекты и понятия предметной области, необходимые для решения поставленной задачи. Оформить их в виде фреймов-прототипов (фреймов-объектов, фреймов-ролей).
- 2) Задать конкретные объекты предметной области. Оформить их в виде фреймов-экземпляров (фреймов-объектов, фреймов-ролей).
- 3) Определить набор возможных ситуаций. Оформить их в виде фреймов-ситуаций (прототипы). Если существуют прецеденты по ситуациям в предметной области, добавить фреймы-экземпляры (фреймы-ситуации).
- 4) Описать динамику развития ситуаций (переход от одних к другим) через набор сцен. Оформить их в виде фреймов-сценариев.
- 5) Добавить фреймы-объекты сценариев и сцен, которые отражают данные конкретной задачи.

Решение.

- 1) Ключевые понятия данной предметной области – ресторан, тот, кто посещает ресторан (клиент) и те, кто его обслуживают (повара, метрдотели, официанты, для простоты ограничимся только официантами). У обслуживающего персонала и клиентов есть общие характеристики, поэтому целесообразно выделить общее абстрактное понятие – человек. Тогда фреймы «Ресторан» и «Человек» являются прототипами-образцами, а фреймы «Официант» и «Клиент» - прототипами-ролями. Также нужно определить основные слоты фреймов – характеристики, имеющие значения для решаемой задачи.

ЧЕЛОВЕК			
Имя слота	Значение слота	Способ получения значения	Демон
пол	Мужской или	из внешних источников	
возраст	От 0 до 120 лет	из внешних источников	

РЕСТОРАН			
Имя слота	Значение слота	Способ получения значения	Демон
Название		из внешних источников	
Адрес		из внешних источников	
Часы работы		из внешних источников	
Специализация		из внешних источников	
Класс	Средний или высший	из внешних источников	

Фреймы-наследники содержат все слоты своих родителей, они явно прописываются только в случае изменения какого-либо параметра.

ОФИЦИАНТ (АКО ЧЕЛОВЕК)

Имя слота	Значение слота	Способ получения значения	Демон
возраст	От 18 до 55 лет	из внешних источников	
стаж работы		из внешних источников	
зарплата		из внешних источников	
график работы		из внешних источников	
место работы	Фрейм-объект	из внешних источников	

КЛИЕНТ (АКО ЧЕЛОВЕК)

Имя слота	Значение слота	Способ получения значения	Демон
Вид оплаты	Наличные или карточка	По умолчанию (наличные)	
Статус	Обычный или Vip	По умолчанию (обычный)	
Форма заказа	Заказ есть или нет	По умолчанию (заказа нет)	
Чаевые		Из внешних источников	

- 2) Фреймы-образцы описывают конкретную ситуацию: какие рестораны имеются в городе, как именно организовывается посещение, кто является посетителем, кто работает в выбранном ресторане и т.д. Поэтому определим следующие фреймы-образцы, являющиеся наследниками фреймов-прототипов:

КАФЕ-РЕСТОРАН "ВКУСНЯТИНА" (АКО РЕСТОРАН)

Имя слота	Значение слота	Способ получения значения	Демон
Название	Вкуснятина	из внешних источников	
Адрес	г. Ульяновск, улица Минаева, 15	из внешних источников	
Часы работы	9:00-00:00	из внешних источников	
Специализация	Пиццерия	из внешних источников	
Класс	Средний или высший	из внешних источников	

КАФЕ "ВКУСНАЯ ЕДА" (АКО РЕСТОРАН)

Имя слота	Значение слота	Способ получения значения	Демон
Название	Вкусная еда	из внешних источников	
Адрес	г. Ульяновск, улица Карла Маркса, 5	из внешних источников	
Часы работы	9:00-00:00	из внешних источников	
Специализация	Паб	из внешних источников	
Класс	Средний	из внешних источников	

СЕРГЕЙ (АКО ОФИЦИАНТ)

Имя слота	Значение слота	Способ получения значения	Демон
возраст	27	из внешних источников	
пол	мужской	из внешних источников	
стаж работы	5	из внешних источников	
зарплата	7 000	из внешних источников	
график работы	Через день с 18:00 до 00:00	из внешних источников	
место работы	КАФЕ "ВКУСНАЯ	из внешних источников	

МАРИНА (АКО ОФИЦИАНТ)			
Имя слота	Значение слота	Способ получения значения	Демон
возраст	24	из внешних источников	
Пол	женский	из внешних источников	
стаж работы	2	из внешних источников	
зарплата	8 200	из внешних источников	
график работы	Каждый день с 9:00 до 14:00	из внешних источников	
место работы	КАФЕ-РЕСТОРАН "ВКУСНЯТИНА"	из внешних источников	

ПЁТР (АКО КЛИЕНТ)			
Имя слота	Значение слота	Способ получения значения	Демон
пол	мужской	из внешних источников	
возраст	19	из внешних источников	
Вид оплаты	Наличные	По умолчанию (наличные)	
Статус	Обычный	По умолчанию (обычный)	
Форма заказа	Заказа нет	По умолчанию (заказа нет)	
Чаевые	7 % от суммы заказа	Из внешних источников	

- 3) Фреймы-ситуации описывают возможные ситуации. В ресторане клиент попадает в несколько типичных ситуаций: заказ и оплата. Возможны и другие не типичные ситуации: клиент подавился, у клиента нет наличности для оплаты счета и т.д. Рассмотрим типичные ситуации (их может быть больше):

ЗАКАЗ			
Имя слота	Значение слота	Способ получения значения	Демон
Перечень блюд		из внешних источников	IF-ADDED (изменяет слот «Перечень цен»)
Перечень цен		Присоединенная процедура	IF-ADDED (изменяет слот «Сумма заказчик»)
Сумма заказа		Присоединенная	
Принял заказ	Фрейм-образец	из внешнего источника	
Сделал заказ	Фрейм-образец	из внешнего источника	

ОПЛАТА			
Имя слота	Значение слота	Способ получения значения	Демон
Вид платежа		из внешних источников	IF-ADDED (изменяет слот «Чаевые»)
Чаевые		Присоединенная	

Оплатил	Фрейм-образец	Присоединенная процедура	
Заказ	Фрейм-образец	из внешних источников	IF-ADDED (изменяет слот «Оплатил»)

- 4) Ситуации возникают после наступления каких-то событий, выполнения условий и могут следовать одна за другой. Динамику предметной области можно отобразить в фреймах-сценариях. Их может быть множество, опишем наиболее общий и типичный сценарий посещения ресторана:

ПОСЕЩЕНИЕ РЕСТОРАНА			
Имя слота	Значение слота	Способ получения значения	Демон
Посетитель	Фрейм-объект	из внешних источников	
Ресторан	Фрейм-объект	из внешних источников	IF-ADDED, IF-REMOVED (изменяют слот «Официант»)
Официант	Фрейм-объект	присоединенная процедура (определяет по выбранному)	
Сцена 1	Вход, выбор	из внешних источников	
Сцена 2	Заказ	из внешних источников	
Сцена 3	Еда	из внешних источников	
Сцена 4	Оплата	из внешних источников	
Сцена 5	Выход	из внешних источников	

- 5) Пусть в рамках нашей задачи Пётр посетил ресторан «Вкусная еда». Тогда фреймы будут заполнены следующим образом:

ПОСЕЩЕНИЕ «Вкусной еды» (АКО ПОСЕЩЕНИЕ РЕСТОРАНА)			
Имя слота	Значение слота	Способ получения значения	Демон
Посетитель	ПЁТР	из внешних источников	
Ресторан	КАФЕ "ВКУСНАЯ ЕДА"	из внешних источников	IF-ADDED, IF-REMOVED (изменяют слот «Официант»)
Официант	СЕРГЕЙ	присоединенная процедура (определяет по выбранному ресторану)	
Сцена 1	Вход, выбор	из внешних источников	
Сцена 2	ЗАКАЗ ПЕТРА	из внешних источников	
Сцена 3	Еда	из внешних источников	
Сцена 4	ОПЛАТА ПЕТРА	из внешних источников	
Сцена 5	Выход	из внешних источников	

ЗАКАЗ ПЕТРА (АКО ЗАКАЗ)			
Имя слота	Значение слота	Способ получения значения	Демон
Перечень блюд	Отбивная, темное пиво	из внешних источников	IF-ADDED (изменяет слот «Перечень цен»)
Перечень цен	250, 75	Присоединенная процедура	IF-ADDED (изменяет слот «Сумма заказк»)
Сумма заказа	325	Присоединенная	
Принял заказ	СЕРГЕЙ	из внешнего источника	
Сделал заказ	ПЕТР	из внешнего источника	

ОПЛАТА ПЕТРА (АКО ОПЛАТА)			
Имя слота	Значение слота	Способ получения значения	Демон
Вид платежа	Наличные	из внешних источников	IF-ADDED (изменяет слот «Чаевые»)
Чаевые	30	Присоединенная процедура	
Оплатил	ПЕТР	из внешних источников	
Заказ	ЗАКАЗ ПЕТРА	из внешних источников	IF-ADDED (изменяет слот «Оплатил»)

Взаимосвязь различных видов фреймов отображается графически в виде графа (рис. 3).

Использование фреймовой модели аналогично семантической, только в процессе получения ответа кроме вершин учитываются и слоты.

Например, получить ответ на вопрос «Кто работает официантом в ресторане “Вкусная еда”?» можно следующим образом: из запроса понятно, что необходимо найти фрейм «Ресторан “Вкусная еда”» и проследить связь с фреймом «Сергей», являющимся наследником фрейма «Официант». Также можно найти слот «Место работы» и проверив его значение во фреймах наследниках фрейма «Официант» определить, что официантом в ресторане “Вкусная еда” работает Сергей.

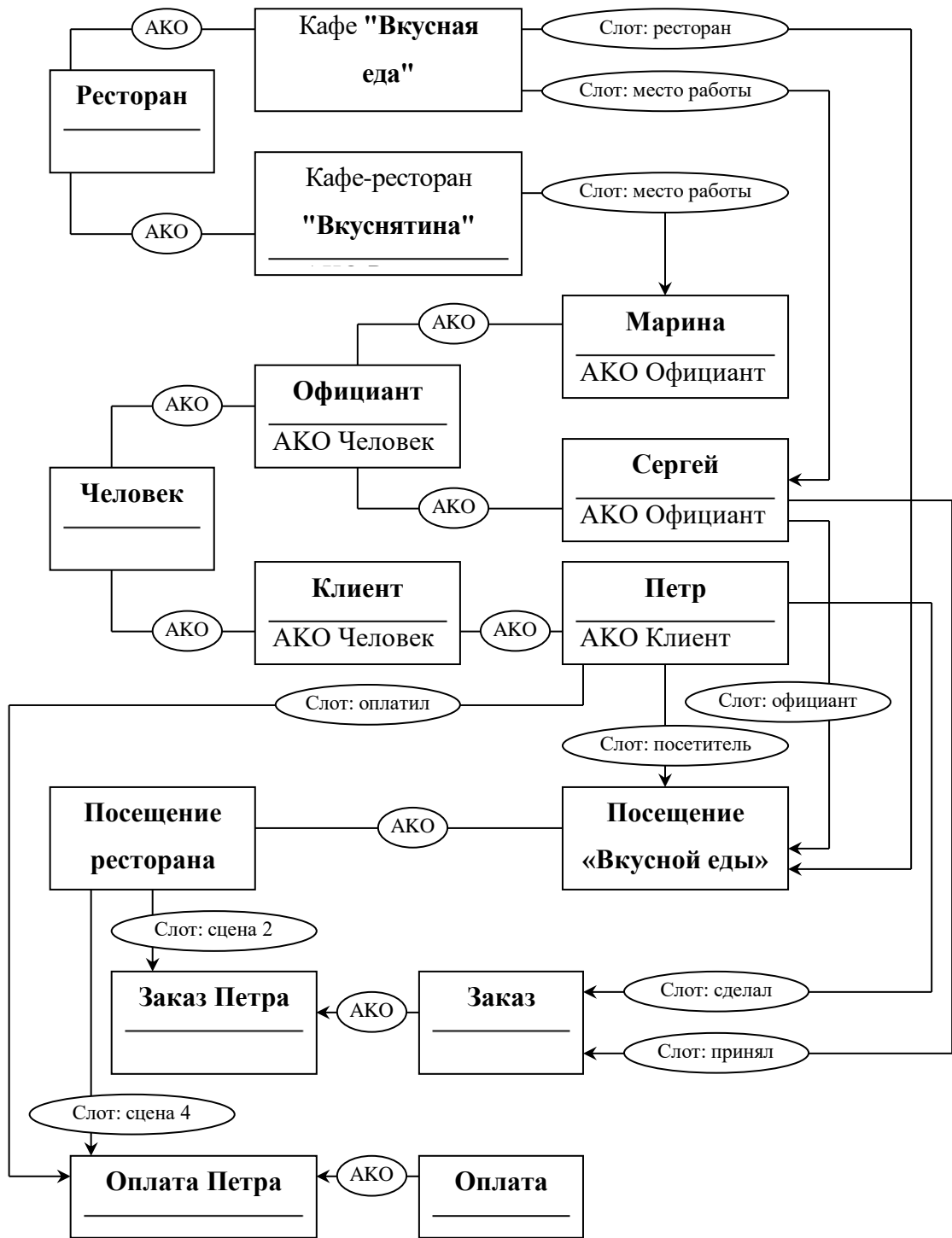


Рис. 3. Схема фреймов для предметной области «Ресторан».

Варианты задачи 3

1. Построить фреймовую модель представления знаний в предметной области «Аэропорт» (диспетчерская).
2. Построить фреймовую модель представления знаний в предметной области «Железная дорога» (продажа билетов).
3. Построить фреймовую модель представления знаний в предметной области «Торговый центр» (организация).

4. Построить фреймовую модель представления знаний в предметной области «Автозаправка» (обслуживание клиентов).
5. Построить фреймовую модель представления знаний в предметной области «Автопарк» (пассажирские перевозки).
6. Построить фреймовую модель представления знаний в предметной области «Компьютерные сети» (организация).
7. Построить фреймовую модель представления знаний в предметной области «Университет» (учебный процесс).
8. Построить фреймовую модель представления знаний в предметной области «Компьютерная безопасность» (средства и способы ее обеспечения).
9. Построить фреймовую модель представления знаний в предметной области «Компьютерная безопасность» (угрозы).
10. Построить фреймовую модель представления знаний в предметной области «Интернет-кафе» (организация и обслуживание).
11. Построить фреймовую модель представления знаний в предметной области «Разработка информационных систем» (ведение информационного проекта).
12. Построить фреймовую модель представления знаний в предметной области «Туристическое агентство» (работа с клиентами).
13. Построить фреймовую модель представления знаний в предметной области «Зоопарк» (организация).
14. Построить фреймовую модель представления знаний в предметной области «Кухня» (приготовление пищи).
15. Построить фреймовую модель представления знаний в предметной области «Больница» (прием больных).
16. Построить фреймовую модель представления знаний в предметной области «Кинопрокат» (ассортимент и работа с клиентами).
17. Построить фреймовую модель представления знаний в предметной области «Прокат автомобилей» (ассортимент и работа с клиентами).
18. Построить фреймовую модель представления знаний в предметной области «Операционные системы» (функционирование).
19. Построить фреймовую модель представления знаний в предметной области «Информационные системы» (виды и функционирование).
20. Построить фреймовую модель представления знаний в предметной области «Предприятие» (структура и функционирование).

Тема 4: Методы вывода на основе знаний

Основные вопросы темы

1. Понятие продукции. Архитектура продукционной системы. Цикл распознавание-действие. Типы вывода: монотонный и немонотонный, прямой и обратный. Символическая логика Поста. Методы поиска в пространстве состояний, представление знаний в исчислении предикат.
2. Метод резолюций. Марковский алгоритм. Rete-алгоритм.

Рекомендации по изучению темы

Вопрос 1 изложен в учебнике [4] в главах 2-5.

Вопрос 2 изложен в учебнике [2] в параграфе 2.6, 2.7, 2.10.

Дополнительные материалы представлены соответствующем разделе лабораторного практикума.

Вопросы для самоподготовки

Рекомендуется после изучения материалов лекций и специальной литературы подготовить ответы на вопросы:

1. Какие виды поиска в пространстве состояний существуют?
2. Какой алгоритм вывода на базе метода резолюций?
3. Возможности применения байесовских сетей доверия?
4. Сущность нормального алгоритма Маркова?
5. Каковы отличия теории Демстера-Шеффера от теории вероятности?
6. Где применяется Rete-алгоритм?
7. Из каких элементов строится сеть в Rete-алгоритм?
8. Основные положения метода Марковские цепи.
9. Описание нормального марковского алгоритма.

Контрольные тесты

- 1. Процесс работы нормального алгоритма считается завершённым, если на данном шаге**
 - a) применена последняя формула в списке формул марковских подстановок, задающих данный алгоритм
 - b) применена заключительная формула подстановки
 - c) ни одна подстановка схемы не подходит
 - d) понятно, что процесс подстановок не сможет остановиться

2. Результат применения нормального алгоритма $ab \rightarrow bd, db \rightarrow ba, bba \rightarrow abb, c$

$\rightarrow \Delta$ к слову $R = abbc$

- a) Алгоритм не применим к этому слову
- b) bb
- c) aa
- d) cc

3. Результат марковской подстановки $21 \rightarrow 3$ в слово 521421

- a) 52143
- b) 5343
- c) 531421
- d) 533433

4. Марковская подстановка $21 \rightarrow 3$ не применима к словам

- a) 521421
- b) 5241
- c) 21
- d) 12

Задачи для практических занятий и самоподготовки

Задание: Постройте дерево решений для приведенной ниже задачи (отображать и тупиковые ветви). Выделите ветку, являющуюся решением. Используйте для отображения программном продукте Edraw (или его аналоги).

Варианты:

- 1) Крестьянину нужно перевезти через реку волка, козу и капусту. Но лодка такова, что в ней может поместиться только крестьянин, а с ним или один волк, или одна коза, или одна капуста. Но если оставить волка с козой, то волк съест козу, а если оставить козу с капустой, то коза съест капусту. Как перевез свой груз крестьянин?
- 2) Отец, два его сына и лодка находятся по одну сторону реки. Отец весит 80 кг, сыновья - по 40 кг. Как переправить эту семью на другую сторону, если лодка выдерживает только 80 кг?
- 3) Вам нужно переправить через реку с помощью одного плота семью (мать, отца, 2-х дочерей и 2-х сыновей) и полицейского с заключенным.

Правила:

1. На плоту могут одновременно перемещаться максимум 2 человека.
2. Папе не разрешается находиться с дочерьми без присутствия матери.
3. Маме не разрешается находиться с сыновьями без присутствия отца.
4. Заключённого нельзя оставлять без полицейского ни с одним из членов семьи.
5. Управлять плотом могут только полицейский и родители.

- 4) Туристы (отец, мать и два брата-близнеца) должны переправиться через реку. В их распоряжении есть маленькая лодка, которая вмещает только одного взрослого или двоих детей. Как организовать переправу, если и взрослые и дети умеют грести?
- 5) Три человека, одна большая и две маленькие обезьяны должны переправиться через реку. Есть одна лодка, в которой может поместиться не больше двоих. Только люди и большая обезьяна умеют грести. Нельзя, чтобы оставались вместе больше обезьян, чем людей, иначе обезьяны нападут на людей или сбегут. Обезьяны могут выпрыгивать на берег, когда лодка причаливает. Как им переправиться через реку?
- 6) Одной семье надо пройти на другую сторону длинного, узкого и очень тёмного тоннеля. Отец может пройти сквозь тоннель за 1 минуту, мать – за 2, сын – за 4 и дочь за 5 минут. У них есть один факел, которого хватит ровно на 12 минут. В тоннеле могут идти не больше двух человек с факелом. Как всей семье перебраться на другую сторону тоннеля, если все боятся темноты?
- 7) Три рыцаря, каждый в сопровождении оруженосца, съехались на берегу реки, намереваясь переправиться на другую сторону. Им удалось найти маленькую двухместную лодку, и переправа произошла бы легко, ведь лошади могли перебраться вплавь. Но одно затруднение чуть было не помешало этому предприятию. Все оруженосцы, словно сговорившись, наотрез отказались оставаться в обществе незнакомых рыцарей без своих хозяев. Не помогли ни уговоры, ни угрозы. Трусливые оруженосцы упорно стояли на своём. И всё же переправа состоялась, все шесть человек благополучно перебрались на другой берег с помощью одной двухместной лодки. При этом соблюдалось условие, на котором настаивали оруженосцы. Как это было сделано?
- 8) Три ревнивых мужа, пришедши с жёнами своими к берегу реки, нашли при оном лодку, в которую по её малости более двух человек вмещаться не могло. Почему спрашивается, как бы через реку переехать сим шести человекам так, чтобы ни одна жена с чужим мужем не переезжала и ни на котором берегу не оставалась.

Задание: Используя метод резолюций, решите задачи 1, 2.

Задача 1

Варианты:

- 1) Доказать с помощью метода резолюций, что формула G есть логическое следствие формул F_1, \dots, F_n : $F_1 = X \square Y$, $F_2 = X \rightarrow Z$, $G = (Y \rightarrow Z) \rightarrow Z$;
- 2) Доказать с помощью метода резолюций, что формула G есть логическое следствие формул F_1, \dots, F_n : $F_1 = X$, $F_2 = X \& Y \rightarrow Z$, $G = Y \rightarrow Z$;

- 3) Доказать с помощью метода резолюций, что формула G есть логическое следствие формул F_1, \dots, F_n : $F_1 = X \rightarrow Y \square Z$, $F_2 = Z \rightarrow W$, $F_3 = \emptyset W$, $G = X \rightarrow Y$;
- 4) Доказать с помощью метода резолюций, что формула G есть логическое следствие формул F_1, \dots, F_n : $F_1 = X \square Y \square \emptyset Z$, $F_2 = X \rightarrow X_1$, $F_3 = Y \rightarrow Y_1$, $F_4 = Z$, $G = X_1 \square Y_1$;
- 5) Доказать с помощью метода резолюций, что формула G есть логическое следствие формул F_1, \dots, F_n : $F_1 = X \& Y \rightarrow \emptyset X \& Z$, $F_2 = \emptyset(X \& \emptyset Y) \square Z$, $G = X \rightarrow Z$;
- 6) Доказать с помощью метода резолюций, что формула G есть логическое следствие формул F_1, \dots, F_n : $F_1 = X \rightarrow [\emptyset Y \& (\emptyset Y \rightarrow Z)]$, $F_2 = (X \rightarrow \emptyset Y) \& \emptyset(\emptyset X \& \emptyset W)$, $G = W \square Z$.

Задача 2

- 1) Записать формально следующее рассуждение на языке логики высказываний и доказать его справедливость, используя метод резолюций.

Посылки: Если идет дождь, то прохладно. Если светит солнце, то жарко. Идет дождь.

Заключение: Прохладно и не светит солнце.

- 2) Записать формально следующее рассуждение на языке логики высказываний и доказать его справедливость, используя метод резолюций.

Посылки: Экзамен сдан вовремя или сессия продлена. Если сессия продлена, то не сдана курсовая работа или не зачтены лабораторные работы. Курсовая работа сдана. Экзамен вовремя не сдан.

Заключение: Неверно, что если курсовая работа сдана, то лабораторные работы зачтены.

- 3) Записать формально следующее рассуждение на языке логики высказываний и доказать его справедливость, используя метод резолюций.

Посылки: Если имеет место денежная эмиссия, то растет курс доллара. Если эмиссии нет и инфляция не растет, то курс доллара не растет. Инфляция не растет.

Заключение: Имеет место эмиссия и растет курс доллара или нет эмиссии и курс доллара не растет.

- 4) Записать формально следующее рассуждение на языке логики высказываний и доказать его справедливость, используя метод резолюций.

Посылки: Заработная плата возрастет только, если будет инфляция. Если будет инфляция, то увеличится стоимость жизни. Заработная плата возрастет.

Заключение: Стоимость жизни увеличится.

- 5) Записать формально следующее рассуждение на языке логики высказываний и доказать его справедливость, используя метод резолюций.

Посылки: Если 2 - простое число, то это наименьшее простое число. Если 2 - наименьшее простое число, то 1 не есть простое число. Число 1 не есть простое число.

Заключение: 2 - простое число.

б) Записать формально следующее рассуждение на языке логики высказываний и доказать его справедливость, используя метод резолюций.

Посылки: Джон или переутомился или он болен. Если он переутомился, то он раздражается. Он не раздражается.

Заключение: Джон болен.

7) Записать формально следующее рассуждение на языке логики высказываний и доказать его справедливость, используя метод резолюций.

Посылки: Если завтра будет холодно, я надену теплое пальто, если рукав будет починен.

Завтра будет холодно, а рукав не будет починен.

Заключение: Я не надену теплое пальто.

8) Записать формально следующее рассуждение на языке логики высказываний и доказать его справедливость, используя метод резолюций.

Посылки: Если исход скачек будет предрешен сговором или в игорных домах будут орудовать шулеры, то доходы от туризма упадут и город пострадает. Если доходы от туризма упадут, полиция будет довольна. Полиция никогда не бывает довольна.

Заключение: Исход скачек не предрешен сговором.

9) Записать формально следующее рассуждение на языке логики высказываний и доказать его справедливость, используя метод резолюций.

Посылки: Или Сэлли и Боб одного возраста, или Сэлли старше Боба. Если Сэлли и Боб одного возраста, то Нэнси и Боб не одного возраста. Если Сэлли старше Боба, то Боб старше Уолтера.

Заключение: Или Нэнси и Боб не одного возраста, или Боб старше Уолтера.

Задание: Постройте байесовскую сеть доверия согласно полученному варианту (предположите несколько причин, определите если связь между ними, постройте сеть и сами назначьте вероятности по своему усмотрению). Используя построенную сеть, определите наиболее вероятную причину наступления события (событие наступило, т.е. например, человек уже страдает от насморка – вариант 1).

Варианты:

- 1) В чем причина насморка?
- 2) Почему не горит лампочка?
- 3) Почему сработала противопожарная сигнализация?
- 4) Почему болит живот?
- 5) В чем причина неурожая?
- 6) Почему не работает чайник?

- 7) Почему Перезагрузился компьютер?
- 8) Почему неустойчивый радиосигнал?
- 9) Почему студент не сдал лабораторную работу?

Задание: Используя метод цепей Маркова, решите задачи 1-4.

Задача 1.

- 1) P – матрица вероятностей перехода из состояния в состояние за один шаг:

$$P = \begin{matrix} & \begin{matrix} 1 & 2 & 3 \end{matrix} \\ \begin{matrix} 1 \\ 2 \\ 3 \end{matrix} & \begin{bmatrix} 1 & 0 & 0 \\ 1/2 & 0 & 1/2 \\ 2/3 & 0 & 1/3 \end{bmatrix} \end{matrix}$$

Построить граф, соответствующий матрице P.

- 2) P – матрица вероятностей перехода из состояния в состояние за один шаг:

$$P = \begin{pmatrix} 0,1 & 0,4 & 0,3 & 0,2 \\ 0,3 & 0,2 & 0,4 & 0,1 \\ 0,2 & 0,3 & 0,2 & 0,3 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Построить граф, соответствующий матрице P.

- 3) P – матрица вероятностей перехода из состояния в состояние за один шаг:

$$P = \begin{pmatrix} \frac{1}{2} & \frac{1}{2} & 0 \\ \frac{1}{3} & 0 & \frac{2}{3} \\ \frac{1}{5} & \frac{2}{5} & \frac{2}{5} \end{pmatrix}$$

Построить граф, соответствующий матрице P.

- 4) P – матрица вероятностей перехода из состояния в состояние за один шаг:

$$P = \begin{pmatrix} \frac{1}{10} & \frac{1}{10} & 0 & \frac{4}{5} \\ \frac{1}{5} & \frac{1}{5} & 0 & \frac{3}{5} \\ \frac{2}{5} & 0 & \frac{1}{10} & \frac{1}{2} \\ \frac{3}{5} & \frac{1}{10} & \frac{2}{10} & \frac{1}{10} \end{pmatrix}$$

Построить граф, соответствующий матрице P.

- 5) P – матрица вероятностей перехода из состояния в состояние за один шаг:

$$P = \begin{pmatrix} \frac{1}{10} & \frac{2}{5} & \frac{1}{5} & \frac{3}{10} \\ \frac{1}{5} & \frac{2}{5} & 0 & \frac{2}{5} \\ \frac{2}{5} & \frac{1}{5} & \frac{1}{5} & \frac{1}{5} \\ \frac{1}{2} & 0 & \frac{1}{2} & 0 \end{pmatrix}$$

Построить граф, соответствующий матрице P.

- 6) P – матрица вероятностей перехода из состояния в состояние за один шаг:

$$P = \begin{pmatrix} \frac{1}{4} & \frac{1}{4} & \frac{1}{2} \\ 0 & 1 & 0 \\ \frac{1}{3} & \frac{1}{3} & \frac{1}{3} \end{pmatrix}$$

Построить граф, соответствующий матрице P.

7) P – матрица вероятностей перехода из состояния в состояние за один шаг:

$$G = \begin{pmatrix} 0.1 & 0.9 & 0 & 0 \\ 0.3 & 0.6 & 0.1 & 0 \\ 0.2 & 0.3 & 0.3 & 0.2 \\ 0 & 0.3 & 0.3 & 0.4 \end{pmatrix}$$

Построить граф, соответствующий матрице P.

8) P – матрица вероятностей перехода из состояния в состояние за один шаг:

$$P = \begin{bmatrix} 0.9 & 0.075 & 0.025 \\ 0.15 & 0.8 & 0.05 \\ 0.25 & 0.25 & 0.5 \end{bmatrix}$$

Построить граф, соответствующий матрице P.

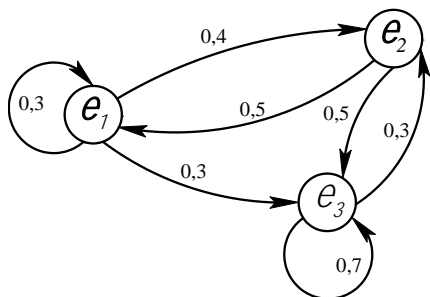
9) P – матрица вероятностей перехода из состояния в состояние за один шаг:

$$P = \begin{bmatrix} 0.8 & 0.1 & 0.1 \\ 0 & 0.7 & 0.3 \\ 0 & 0 & 1 \end{bmatrix}$$

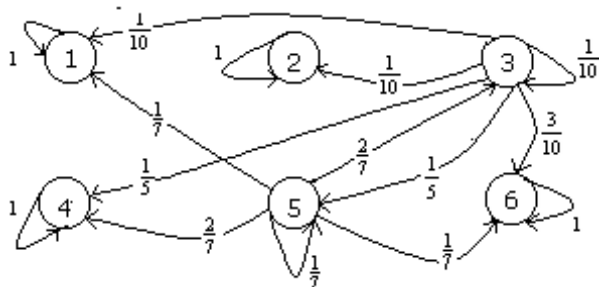
Построить граф, соответствующий матрице P.

Задача 2

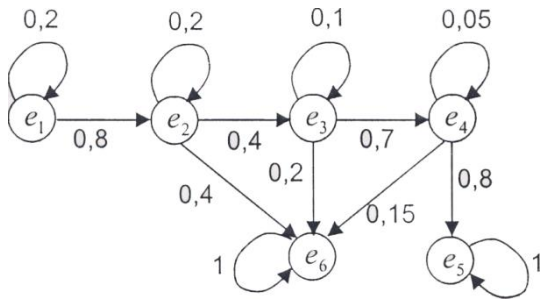
1) Дан граф. Составить переходную матрицу.



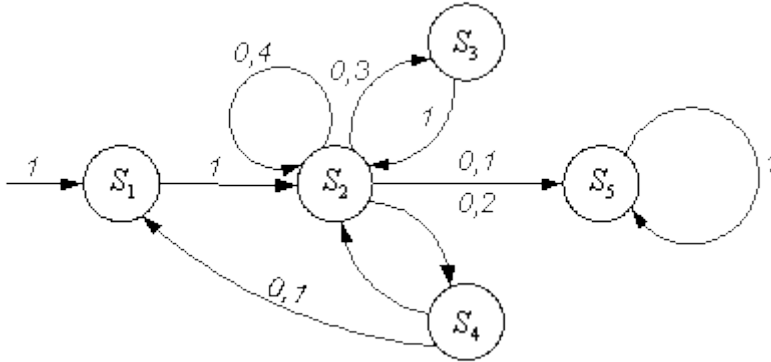
2) Дан граф. Составить переходную матрицу.



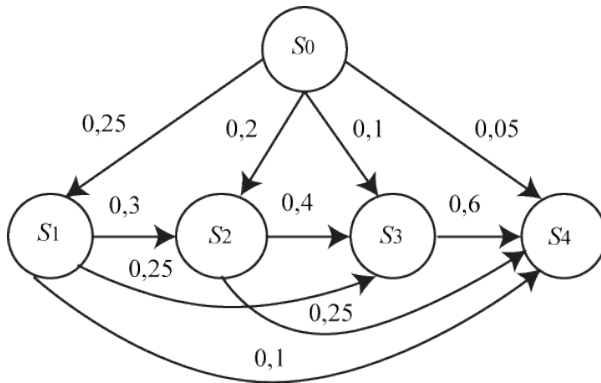
3) Дан граф. Составить переходную матрицу.



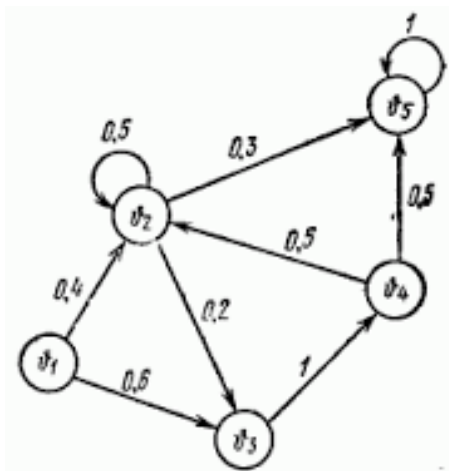
4) Дан граф. Составить переходную матрицу.



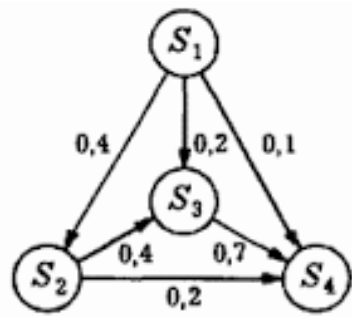
5) Дан граф. Составить переходную матрицу.



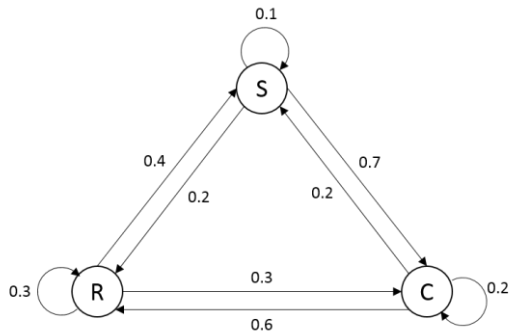
6) Дан граф. Составить переходную матрицу.



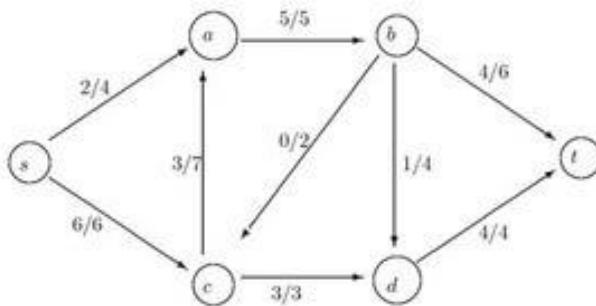
7) Дан граф. Составить переходную матрицу.



8) Дан граф. Составить переходную матрицу.



9) Дан граф. Составить переходную матрицу.



Задача 3

1) Дана матрица вероятностей перехода

$$P = \begin{bmatrix} 1/3 & 2/3 \\ 1/5 & 4/5 \end{bmatrix}$$

и вектор начальных вероятностей $\vec{a} \left(\frac{3}{4}, \frac{1}{4} \right)$.

Найти распределение

по состояниям в момент $t = 2$.

2) Матрица вероятностей перехода цепи Маркова имеет вид

$$P = \begin{bmatrix} 0,4 & 0,5 & 0,1 \\ 0,2 & 0,3 & 0,5 \\ 0,3 & 0,4 & 0,3 \end{bmatrix}$$

Вектор начальных вероятностей $\vec{a}(0, 1; 0, 2; 0, 7)$.

Найти

вероятность того, что через два шага система будет находиться в состоянии E_3 .

3) Частица находится в одном из трех состояний E_1 , E_2 , E_3 , которые выбираются в начальный момент с вероятностями $1/2$, $1/3$, $1/6$ соответственно. Под воздействием случайных толчков система переходит в другое состояние с вероятностью $1/4$ и остается на месте с вероятностью $1/2$. Найти вероятность того, что через два толчка система будет находиться в состоянии E_1 .

4) Рассмотрим марковскую цепь с двумя состояниями 1 и 2 с матрицей вероятностей перехода:

$$P = \begin{bmatrix} 1/3 & 2/3 \\ 1/2 & 1/2 \end{bmatrix}.$$

С помощью особого устройства случайного выбора мы выбираем состояние, с которого начинается процесс. Это устройство выбирает состояние 1 с вероятностью $1/2$ и состояние 2 с вероятностью $1/2$. Требуется найти вероятность того, что после первого шага этот процесс перейдет в состояние E_1 ;

5) Рассмотрим марковскую цепь с двумя состояниями 1 и 2 с матрицей вероятностей перехода:

$$P = \begin{bmatrix} 1/3 & 2/3 \\ 1/2 & 1/2 \end{bmatrix}.$$

С помощью особого устройства случайного выбора мы выбираем состояние, с которого начинается процесс. Это устройство выбирает состояние 1 с вероятностью $1/3$ и состояние 2 с вероятностью $2/3$.

6) Матрица вероятностей перехода цепи Маркова равна

$$P = \begin{pmatrix} 0,1 & 0,4 & 0,3 & 0,2 \\ 0,3 & 0,2 & 0,4 & 0,1 \\ 0,2 & 0,3 & 0,2 & 0,3 \\ 0 & 0 & 0 & 1 \end{pmatrix}. \text{ Начальное распределение дается вектором } \begin{pmatrix} 0,2 \\ 0,3 \\ 0,5 \\ 0 \end{pmatrix} \text{ Найти}$$

распределение вероятностей состояний в моменты $n = 2$.

7) Матрица вероятностей перехода цепи Маркова равна

$$P = \begin{pmatrix} 0,1 & 0,4 & 0,3 & 0,2 \\ 0,3 & 0,2 & 0,4 & 0,1 \\ 0,2 & 0,3 & 0,2 & 0,3 \\ 0 & 0 & 0 & 1 \end{pmatrix}. \text{ Начальное распределение дается вектором } \begin{pmatrix} 0,2 \\ 0,3 \\ 0,5 \\ 0 \end{pmatrix} \text{ Найти}$$

вероятность того, что через 3 шага состояние цепи будет 3.

8) Рассмотрим марковскую цепь с двумя состояниями E_1 и E_2 с матрицей вероятностей перехода:

$$P = \begin{bmatrix} 1/3 & 2/3 \\ 1/2 & 1/2 \end{bmatrix}$$

С помощью особого устройства случайного выбора мы выбираем состояние, с которого начинается процесс. Это устройство выбирает E1 с вероятностью 1/2 и E2 с вероятностью 1/2. Требуется:

- а) найти вероятность того, что после первого, шага этот процесс перейдет в состояние E1;
 - б) то же самое для случая, когда это устройство выбивает E1 с вероятностью 1/3 и E2 с вероятностью 2/3.
- 9) Погода на некотором острове через длительные периоды времени становится то дождливой (Д), то сухой (С). Вероятности ежедневных изменений заданы матрицей:

$$P = \begin{matrix} & \begin{matrix} Д & С \end{matrix} \\ \begin{matrix} Д \\ С \end{matrix} & \begin{bmatrix} 0,7 & 0,3 \\ 0,4 & 0,6 \end{bmatrix} \end{matrix}$$

- а) Если в среду погода дождливая, то какова вероятность, что она будет дождливой в ближайшую пятницу?
- б) Если в среду ожидается дождливая погода с вероятностью 0,3, то какова вероятность, что она будет дождливой в ближайшую пятницу?

Задача 4

- 1) Автомашина может находиться в одном из четырех состояний: - исправна, - неисправна, -осматривается, - ремонтируется, - списана.

Если машина исправна, то с вероятностью 0,8 она может сломаться, если машина неисправна, то она с вероятностью 0,7 ремонтируется или с вероятностью 0,3 списывается; если же машина ремонтируется, то она с вероятностью 0,6 становится исправной, либо с вероятностью 0,4 продолжает ремонтироваться. Остальные переходы считать невозможными. Найти вероятность того, что машина будет исправна в субботу, если известно, что она была исправна в среду.

- 2) Рассматриваются следующие состояния телефона-автомата: телефон свободен, телефон занят и нет очереди, телефон занят и в очереди один человек, телефон занят и в очереди 2 человека. Предполагается, что третьим в очередь никто не встанет,

предпочитая искать другой телефон. В каждую минуту с вероятностью $\frac{1}{11}$ может

подойти один человек (больше одного подойти не может), а с вероятностью $\frac{1}{5}$

разговор в данную минуту заканчивается. С какой вероятностью через 3 минуты в очереди будет один человек, если в настоящий момент времени телефон свободен?

3) В любой данный день человек здоров или болен. Если человек здоров сегодня, то вероятность того, что он будет здоров и завтра оценивается в 98%. Если человек сегодня болен, то завтра он будет здоров лишь в 30% случаев. Описать последовательность состояний здоровья как Марковскую цепь. Определить:

а) вероятность того, что человек выздоровеет завтра, послезавтра и на третий день, если сегодня он болен;

б) ожидаемое число дней, в течение которых больной на сегодняшний день человек остается больным.

4) Предположим, что некая фирма осуществляет доставку оборудования по Москве: в северный округ (обозначим А), южный (В) и центральный (С). Фирма имеет группу курьеров, которая обслуживает эти районы. Понятно, что для осуществления следующей доставки курьер едет в тот район, который на данный момент ему ближе. Статистически было определено следующее:

- после осуществления доставки в А следующая доставка в 30 случаях осуществляется в А, в 30 случаях - в В и в 40 случаях - в С;

- после осуществления доставки в В следующая доставка в 40 случаях осуществляется в А, в 40 случаях - в В и в 20 случаях - в С;

- после осуществления доставки в С следующая доставка в 50 случаях осуществляется в А, в 30 случаях - в В и в 20 случаях - в С.

Таким образом, район следующей доставки определяется только предыдущей доставкой. Допустим, что каждая доставка с последующим перемещением в следующий район занимает 15 минут. Если курьер стартует из С, какова вероятность того, что осуществив две доставки, он будет в В, т.е. как можно достичь В в 2 шага?

5) Некоторая совокупность рабочих семей поделена на три группы: ε_1 – семьи, не имеющие автомашины и не намеревающиеся ее приобрести; ε_2 – семьи, не имеющие автомашины, но собирающиеся ее приобрести, и, наконец, ε_3 – семьи, имеющие автомашину. Статистические обследования дали возможность оценить вероятность перехода семей из одной группы на протяжении года в другую. При этом матрица перехода оказалась такой:

$$P = \begin{bmatrix} 0,8 & 0,1 & 0,1 \\ 0 & 0,7 & 0,3 \\ 0 & 0 & 1 \end{bmatrix}.$$

Из этой матрицы следует, что вероятность того, что семьи, которые в предшествующем году имели машину и в следующем за ним году, наверняка, будут иметь ее, равна $p_{33} = 1$. В матрице элемент $p_{23} = 0,3$ означает вероятность того, что семья, не имевшая в предыдущем году машины, но решившая приобрести ее, осуществит свое намерение в этом году. Далее, $p_{21} = 0$ означает вероятность того, что семья, намеревавшаяся купить автомашину в прошлом году, в следующем за ним году от этого намерения вообще отказалась. Вычислить вероятность того, что семья, не имевшая машины и не собиравшаяся ее приобрести, будет находиться в той же ситуации через 2 года

- б) В учениях участвуют два корабля, которые одновременно производят выстрелы друг в друга и через равные промежутки времени. При каждом обмене выстрелами корабль А поражает корабль В с вероятностью $1/2$, корабль В поражает корабль А с вероятностью равной $3/8$. Предполагается, что при любом попадании корабль выходит из строя. Рассматриваются результаты серии выстрелов.

Требуется:

а) построить матрицу вероятностей перехода, вычислив переходные вероятности p_{ij} , если состояниями цепи являются комбинации кораблей, оставшихся в строю: e_1 – оба корабля в строю; e_2 – в строю корабль А; e_3 – в строю корабль В; e_4 – оба корабля поражены.

б) построить граф этой системы.

- 7) Некоторая совокупность рабочих семей поделена на три группы: e_1 – семьи, не имеющие автомашины и не намеревающиеся ее приобрести; e_2 – семьи, не имеющие автомашины, но собирающиеся ее приобрести, и, наконец, e_3 – семьи, имеющие автомашину. Статистические обследования дали возможность оценить вероятность перехода семей из одной группы на протяжении года в другую. При этом матрица перехода оказалась такой:

$$P = \begin{bmatrix} 0,8 & 0,1 & 0,1 \\ 0 & 0,7 & 0,3 \\ 0 & 0 & 1 \end{bmatrix}.$$

Из этой матрицы следует, что вероятность того, что семьи, которые в предшествующем году имели машину и в следующем за ним году, наверняка, будут иметь ее, равна $p_{33} = 1$. В матрице элемент $p_{23} = 0,3$ означает вероятность того, что семья, не имевшая в предыдущем году машины, но решившая приобрести ее, осуществит свое намерение в этом году. Далее, $p_{21} = 0$ означает вероятность того, что семья, намеревавшаяся купить автомашину в прошлом году, в следующем за ним году

от этого намерения вообще отказалась. Вычислить вероятность того, что семья, не имевшая машины и не собиравшаяся ее приобрести, будет находиться в той же ситуации через 2 года. Вычислить вероятность того, что семья, не имевшая автомашины и намеревающаяся ее приобрести, будет иметь автомашину через 2 года.

- 8) Две машины А и В сдаются в аренду по одной и той же цене. Эти машины имеют следующие матрицы переходных вероятностей:

$$P_A = \begin{bmatrix} 0,8 & 0,2 \\ 0,7 & 0,3 \end{bmatrix}; \quad P_B = \begin{bmatrix} 0,7 & 0,3 \\ 0,6 & 0,4 \end{bmatrix}.$$

где ε_1 – состояние, когда машина работает хорошо;

ε_2 – состояние, когда машина требует регулировки.

Определить вероятности для обеих машин. Какую машину стоит арендовать?

- 9) Погода на некотором острове через длительные периоды времени становится то дождливой (состояние ε_1), то сухой (состояние ε_2). Вероятности ежедневных изменений заданы матрицей:

$$P = \begin{pmatrix} 0,7 & 0,3 \\ 0,4 & 0,6 \end{pmatrix}$$

Вычислить:

- матрицы прогноза погоды на данном острове на три дня вперед;
- вероятность солнечной погоды в ближайшую субботу, если в среду погода была дождливой.

Задание: Используя НАМ, решите задачи 1-4.

Варианты:

Задача 1

- 1) Пусть для слов в алфавите $A=\{a,b,c,d\}$ и заданы следующие марковские подстановки:

$ab \rightarrow dc$;	$cb \rightarrow d$;	$dac \rightarrow acd$;
$ac \rightarrow dc$;	$abc \rightarrow \Delta$;	$b \rightarrow a$;
$dd \rightarrow bb$;	$cba \rightarrow \Delta$;	$a \rightarrow bd$;
$bc \rightarrow a$;	$da \rightarrow \Delta$;	

Применить каждую из них к слову $abcdcdacba$.

- 2) Пусть для слов в алфавите $A=\{a,b,c,d\}$ и заданы следующие марковские подстановки:

$ab \rightarrow dc$;	$cb \rightarrow d$;	$dac \rightarrow acd$;
$ac \rightarrow dc$;	$abc \rightarrow \Delta$;	$b \rightarrow a$;
$dd \rightarrow bb$;	$cba \rightarrow \Delta$;	$a \rightarrow bd$;
$bc \rightarrow a$;	$da \rightarrow \Delta$;	

Применить каждую из них к слову $ddacbabс$.

3) Пусть для слов в алфавите $A=\{a,b,c,d\}$ и заданы следующие марковские подстановки:

$ab \rightarrow dc$;	$cb \rightarrow d$;	$dac \rightarrow acd$;
$ac \rightarrow dc$;	$abc \rightarrow \Delta$;	$b \rightarrow a$;
$dd \rightarrow bb$;	$cba \rightarrow \Delta$;	$a \rightarrow bd$;
$bc \rightarrow a$;	$da \rightarrow \Delta$;	

Применить каждую из них к слову $ababсdас$.

4) Пусть для слов в алфавите $A=\{a,b,c,d\}$ и заданы следующие марковские подстановки:

$b \rightarrow a$;	$сab \rightarrow \Delta$;
$bc \rightarrow са$;	$a \rightarrow b$;
$бса \rightarrow \Delta$;	$ab \rightarrow bc$;
$bcab \rightarrow \Delta$;	$abc \rightarrow \Delta$;
$c \rightarrow b$;	$abса \rightarrow a$.
$са \rightarrow ab$	

Применить каждую из них к слову $abcabсabсab$ максимально возможное число раз.

5) Пусть для слов в алфавите $A=\{a,b,c,d\}$ и заданы следующие марковские подстановки:

$b \rightarrow a$;	$сab \rightarrow \Delta$;
$bc \rightarrow са$;	$a \rightarrow b$;
$бса \rightarrow \Delta$;	$ab \rightarrow bc$;
$bcab \rightarrow \Delta$;	$abc \rightarrow \Delta$;
$c \rightarrow b$;	$abса \rightarrow a$.
$са \rightarrow ab$;	

Применить каждую из них к слову $бсабсabсabса$ максимально возможное число раз.

6) Пусть для слов в алфавите $A=\{a,b,c,d\}$ и заданы следующие марковские подстановки:

$b \rightarrow a$;	$сab \rightarrow \Delta$;
$bc \rightarrow са$;	$a \rightarrow b$;
$бса \rightarrow \Delta$;	$ab \rightarrow bc$;
$bcab \rightarrow \Delta$;	$abc \rightarrow \Delta$;
$c \rightarrow b$;	$abса \rightarrow a$.
$са \rightarrow ab$;	

Применить каждую из них к слову $сabсabсabсab$ максимально возможное число раз.

7) Пусть для слов в алфавите $A=\{a,b,c,d\}$ и заданы следующие марковские подстановки:

$b \rightarrow a$;	$бсab \rightarrow \Delta$;
$bc \rightarrow са$;	$c \rightarrow b$;
$бса \rightarrow \Delta$;	$са \rightarrow ab$;

$cab \rightarrow \Delta$;

$abc \rightarrow \Delta$;

$a \rightarrow b$;

$abca \rightarrow a$.

$ab \rightarrow bc$;

Применить каждую из них к слову $abcdacba$.

Задача 2

- 1) В произвольном слове, состоящем из букв $\{a, b, c\}$, все подряд стоящие одинаковые буквы заменить одной буквой (например, слово « $abbbcaaa$ » преобразовать в « $abca$ »).
- 2) Удвоить слово, состоящее из одинаковых символов (для определенности — « x »). Т.е. слово « x » надо преобразовать в « xx », слово « xx » — в « $xxxx$ » и т.д. Нельзя написать $x \rightarrow xx$, т.к. в этом случае на каждом шаге НАМ к слову будет добавляться символ « x » и этот процесс будет бесконечным. Необходимо контролировать удвоение каждого символа слова так, чтобы каждый символ удвоился только один раз. (Для это введём маркер, с помощью которого будем обеспечивать контекст применения удваивающего правила).
- 3) Дано слово в алфавите $\{a, b, c\}$. Упорядочить буквы входного слова в лексикографическом порядке.
- 4) Дан алфавит $A = \{a, b, c\}$. Если во входное слово P входит буква b , то заменить все вхождения букв b на c .
- 5) Дан алфавит $A = \{a, b, c\}$. Заменить во входном слове P все вхождения ba , на a .
- 6) Дан алфавит $A = \{a, b\}$. Заменить во входном слове P все стоящие на чётных позициях вхождения b на bab . Пример: $abaababb \rightarrow ababaababbab$.
- 7) Дан алфавит $A = \{a, b\}$. Удалить во входном слове P два последних вхождения буквы a . Пример: $aabaababb \rightarrow aababbb$.

Задача 3

- 1) Дан алфавит $A = \{a, b\}$, входное слово P чётной длины. Стереть правую половину слова P . Пример: $babbaa \rightarrow bab$.
- 2) Дан алфавит $A = \{a, b\}$. Обратить входное слово P . Пример: $babbaa \rightarrow aabbbab$.
- 3) $A = \{a, b\}$. Если в непустом слове P совпадают первый и последний символы, то удалить оба этих символа, а иначе слово не менять.
- 4) $A = \{a, b\}$. Определить, является ли слово P палиндромом (перевёртышем, симметричным словом).
- 5) $A = \{a, b\}$. Пусть слово P имеет нечётную длину. Удалить из него средний символ.

- 6) Дан алфавит $A = \{a, b, c\}$. Приписать к концу P столько палочек, сколько раз буква b входит в P . Пример: $abcbb \rightarrow abcbb|||$.
- 7) Дан алфавит $A = \{0, 1\}$. Входное слово P представляет собой двоичную запись некоторого числа n . Написать число $n + 1$ в двоичной системе счисления. Пример: $1011 \rightarrow 1100$.

ЛАБОРАТОРНЫЙ ПРАКТИКУМ

Порядок выполнения лабораторных работ может быть произвольным и определяется уровнем освоения компетенций обучающегося.

Тема 2: Методы извлечения и структурирования знаний

Задание лабораторной работы

Задание. Построить ментальную карту, используя сервис Coggle (<https://coggle.it/>) или аналогичный (см. <https://texterra.ru/blog/obzor-15-besplatnykh-programm-dlya-sozdaniya-intellekt-kart.html>) согласно варианту.

Отчет по лабораторной должен содержать описание ментальной карты (постановка задачи) и саму ментальную карту (отчет можно сдавать в электронном виде).

Справочное видео по сервису: <https://evgeniypopov.com/blog/coggleit-servis-dlya-sozdaniya-intellekt-kart-onlajn.html>.

Методические указания по выполнению лабораторной работы

Ментальные карты (интеллект-карты) — это техника визуализации мышления. Применения ментальных карт очень разнообразны — например, их можно использовать для того, чтобы зафиксировать, понять и запомнить содержание книги или текста, сгенерировать и записать идеи, разобраться в новой для себя теме, подготовиться к принятию решения.

Как нарисовать ментальную карту: инструкция

1. Возьмите лист бумаги и напишите в центре одним словом главную тему, которой посвящена карта. Заключите ее в замкнутый контур.
2. От центральной темы рисуйте ветви и располагайте на них ключевые слова, которые с ней связаны.
3. Продолжайте расширять карту, добавляя к уже нарисованным ветвям подветви с ключевыми словами, пока тема не будет исчерпана.

Правила работы с ментальными картами

Техника кажется интуитивно понятной, однако приведенные ниже правила помогают существенно повысить ее эффективность.

- Пишите *одно* слово на одной ветви. Такой подход значительно экономит время и место и способствует лучшей читаемости карты. Это сначала кажется непривычным, — у вас может возникнуть опасение, что вы забудете остальные

слова. На самом деле не забудьте, если выберете в качестве ключевых слов наиболее характерные, яркие, запоминаемые, «цепляющие» слова.

- Располагайте лист горизонтально — такую карту будет удобнее читать.
- Пишите ключевые слова печатными буквами, черным цветом, как можно яснее и четче.
- Ключевые слова размещайте прямо на линиях, отображающих их взаимосвязь. Не заключайте их в какие-либо рамки. Пишите на каждой линии только одно ключевое слово.
- Длина линии должна быть равна длине слова — не делайте линии длиннее слов. Не прерывайте линий.
- Используйте разные цвета для основных ветвей, чтобы они не сливались визуально.
- Варьируйте размер букв в надписях и толщину ветвей в зависимости от степени удаленности от главной темы.
- Располагайте ветви равномерно — не оставляйте пустого места и не размещать ветви слишком плотно.
- Используйте рисунки и символы (как минимум — для центральной темы, лучше — для всех основных ветвей).
- Если вы рисуете сложную карту, есть смысл вначале набросать мини-карту с основными ветвями, чтобы определить структуру будущей карты, поскольку выбор основных ветвей влияет на организацию и читаемость карты.

Ментальные карты — это способ записи, альтернативный по отношению к тексту, спискам и схемам (например, «деревьям» или диаграммам связей). Главное отличие ментальных карт от других способов визуализации прежде всего тем, что ментальные карты *активируют память*. Списки, сплошной текст, деревья и схемы **однообразны**. Ментальные карты, наоборот, используют все возможные способы, чтобы активировать восприятие посредством **разнообразия**: разная толщина линий, разные цвета ветвей, точно выбранные ключевые слова, которые лично для вас являются значимыми, использование образов и символов. Техника ментальных карт помогает не только организовать и упорядочить информацию, но и лучше воспринять, понять, запомнить и проассоциировать ее.

Когда использовать ментальные карты?

1. **В самообучении.** Изучение любого материала может идти быстрее, если фиксировать основные мысли в форме ментальной карты. Даже если вы просто читаете книгу, сделайте ментальную карту и вы увидите, насколько это удобный

способ осмыслить прочитанное (не говоря уже о том, что эта карта может вам очень пригодиться, если вы впоследствии захотите быстро освежить в памяти содержание книги).

2. **Для создания идей.** Напишите задачу в центре и располагайте на ветках идеи или ассоциации, от них — следующие ассоциации, все время задавая себе вопрос, как это может решить вашу задачу.
3. **Чтобы разобраться в новой области.** Бывают ситуации, когда вы чувствуете, что тема ускользает от понимания. Напишите ее в середине и располагайте на исходящих из центра ветках все, что к ней относится. Когда карта станет большой и хаотичной, перерисуйте ее, используя несколько основных ветвей в качестве структурных.
4. **Чтобы представить большое количество информации в емкой форме.** Вы можете делать это для большей наглядности, например в ходе презентаций, или для себя самого (например, чтобы не забыть что-то важное во время выступления).

Рекомендации

Старайтесь получать удовольствие от самого процесса рисования ментальной карты. Сделайте свою карту красивой, ведь для этого ведь нужно совсем немного — аккуратный текст печатными буквами, ровные линии той же длины, что и ключевые слова, разные цвета для ветвей, чтобы они не путались между собой, самые простые рисунки-символы, обозначающие важные для вас моменты... Разумеется, степень «красоты» может варьироваться в зависимости от задачи, однако не жалейте времени, особенно в начале освоения техники. После определенной практики его будет уходить совсем немного.

Интересный момент в ментальных картах — то, что они являются не только средством наглядного представления процесса нашего мышления, но и одновременно его диагностикой. Сама форма карты, то, как она визульно выглядит, многое говорит о вашем отношении к теме, о том, насколько понятны вам ее отдельные аспекты (представленные основными ветвями), о вашем способе восприятия этой информации.

Очень рекомендуем начать с рисования ментальных карт руками. Для начинающих кажется быстрее и легче воспользоваться компьютерными программами, но это не дает ни навыков самого майндмэппинга, ни полезного эффекта организации собственного мышления в процессе этого. Программа для построения ментальных карт никогда не позволит вам нарисовать неправильно закомпонированную карту, и это на самом деле проблема, потому что, думая за вас, она лишает вас возможности диагностировать свое мышление. Дело в том, что благодаря рисованию руками Вы можете наглядно увидеть, как и насколько эффективно организовано Ваше мышление по какой-то теме.

Принципы

Если обобщить приведенные выше рекомендации и правила, то в их основе лежат несколько принципов:

- Ментальная карта должна активировать восприятие и память.
- Стремление к большей четкости и удобочитаемости карты дает больше осознанности в мышлении.
- Анализ уже нарисованной вами карты дает подсказки к «устройству» вашего мышления по теме.

Идея ментальных карт основывается на сходствах между изображением процесса мышления при помощи ментальной карты и устройством человеческого мозга: во-первых, ментальная карта имеет, как и нейрон, радиальную структуру, а во-вторых, мысли на физическом уровне отображаются как «деревья» биохимических импульсов.

Варианты для выполнения лабораторной работы

- 1) Цифровая техника (виды устройств, гаджеты).
- 2) Виды электронно-вычислительных устройств.
- 3) Языки программирования.
- 4) Технологии создания дисплеев.
- 5) Запоминающие устройства.
- 6) Устройство компьютера.
- 7) Периферийные устройства.
- 8) Сетевые технологии.
- 9) Ресурсы интернет.
- 10) Направления в информационных технологиях.
- 11) Роботы.
- 12) Компьютерные игры (виды).
- 13) Операционные системы.
- 14) История развития электронной техники (этапы, архитектуры).
- 15) Форматы хранения информации.
- 16) Свободный вариант (студент предлагает тему и согласует ее с преподавателем).

Тема 3: Методы формализации (представления и кодирования) знаний

Задание лабораторной работы

Цель работы: получение практических навыков применения специализированных оболочек для построения продукционных экспертных систем.

Задание: На языке COOL оболочки экспертных систем CLIPS реализовать экспертную систему согласно полученному варианту. Использовать классы, объекты, продукции при написании программы.

Реализовать макет ЭС должен включать в себя:

- Организацию ввода информации (опрос системой пользователя).
- Структуру правил системы (2-3 реально работающих правила, для остальных – описание работы на естественном языке).
- Структуру функций системы (2-3 реально работающих функции, для остальных – описание работы на естественном языке).
- Организацию вывода информации на экран и в файл.

Общее количество правил, функций и хэндлеров в сумме должно составлять не менее 20.

Отчет по лабораторной работе должен содержать:

1. Фамилию и номер группы учащегося, задание
2. Краткое описание предметной области и решаемой задачи (каков результат работы системы, что является входными данными)
3. Описание примененного подхода к извлечению знаний с необходимыми рисунками, диаграммами (деревья И/ИЛИ, деревья решений) и т.д.
4. Полный текст базы знаний и протокол ее работы для трех консультаций

Методические указания по выполнению лабораторной работы

Экспертные системы – это сложные программные комплексы, аккумулирующие знания специалистов в конкретных предметных областях и тиражирующие эти знания для консультации менее квалифицированных специалистов.

CLIPS сочетает в себе 3 парадигмы программирования: логическую, процедурную и объектно-ориентированную. Также, в CLIPS предусмотрены 3 основных формата представления информации: факты, глобальные переменные и объекты.

Команды

Команда (run) запускает выполнение программы в среде (с машиной логического вывода), (reset) – обновляет рабочую память, а (clear) – полностью очищает среду CLIPS

(рабочую память). Опция главного меню Load позволяет загрузить файл с конструкциями, а Load Batch – файл с командами (программу).

Главный файл программы на CLIPS (например, main.bat) может иметь следующий вид:

```
(clear)
(load* "classes.clp") ; загрузка структуры классов
(load-instances "instances.clp") ; загрузка экземпляров классов
(load* "templates.clp") ; загрузка предопределённых фактов
(load* "functions.clp") ; загрузка функций
(load* "rules.clp") ; загрузка правил
(reset)
(run)
```

Вывод информации может осуществляться как на экран, так и в выходные файлы.

Например, нижеследующая функция может использоваться для запроса у пользователя значения, присвоения его переменной, а затем вывода его на экран:

```
(deffunction foo
  () ; функция не имеет аргументов
  (printout t crlf "Please input value:" crlf) ; t – ключ для вывода на экран
  (bind ?var (read)) ; bind – функция для присваивания значения
  (printout t crlf "Your value: " ?var crlf) ; crlf – символ конца строки
)
```

В свою очередь, команды (save-facts ?file+name) и (save-instances ?file+name) служат для сохранения всех имеющихся в CLIPS на текущий момент фактов и экземпляров классов соответственно. Значением переменной ?file+name может быть, например, "system_output.clp".

Обратите внимание, что в CLIPS в именах функций для обозначения пробела обычно используется знак «-», а в именах переменных и строковых значениях – знак «+». Имя переменной всегда начинается со знака «?».

Форматы представления данных в CLIPS

Факты являются одной из основных форм высокого уровня для представления информации в системе CLIPS. Факт (fact) – это список элементарных значений, на которые ссылаются либо позиционно (упорядоченные (ordered) факты), либо по имени (неупорядоченные (non-ordered) или шаблонные (template) факты).

Каждый факт представляет часть информации и помещается в текущий список фактов (fact-list). Факты могут быть добавлены в список фактов (используя команду

assert), удалены из него (используя команду retract), изменены (используя команду modify) или скопированы (используя команду duplicate) в результате явного воздействия пользователя или при исполнении программы CLIPS.

Глобальные переменные. Конструкция defglobal позволяет описывать переменные, которые являются глобальными в контексте окружения CLIPS. То есть глобальная переменная доступна в любом месте окружения CLIPS и сохраняет свое значение независимо от других конструкций. Напротив, некоторые конструкции (как, например, defrule или deffunction) могут иметь собственные локальные переменные, которые задаются в пределах описания конструкции. Обращение к таким переменным возможно только изнутри конструкции, где они описаны; за ее пределами они не имеют значения.

Функция bind используется, чтобы задать значения глобальным переменным. Значения глобальных переменных сбрасываются к начальным установочным значениям при выполнении команды окружения reset или если для глобальной переменной вызвана функция bind без соответствующего значения.

Объекты в CLIPS могут быть описаны как символьные, строковые, целые или вещественные числа, значения с множеством полей, внешние адреса или объекты определенного пользователем класса.

Для создания пользовательского класса используется конструкция defclass. Объект пользовательского класса создается посредством функции make-instance, и к созданному таким образом объекту можно обращаться по уникальному адресу. В пределах модульного контекста к объекту можно уникально обращаться и по имени.

; описание класса car (машина)

(defclass Car

(is-a USER) ; пользовательский класс

(single-slot model+name ; слот «название модели» (одно значение)

(type STRING)) ; тип данных – строка

(multislot producer ; слот «производитель» (0 или более значений)

(type STRING)) ; тип данных – строка

)

; создание экземпляра класса Car

(make-instance Toyota+Corolla of Car

(model+name "Corolla")

(producer "Toyota" "Kanto Auto Works")

)

Взаимодействие с классами осуществляется путём отправки им «сообщений». Для этого сначала определяется «хэндлер» (по смыслу близок к функциям), через конструкцию `defmessage-handler`:

(defmessage-handler Guideline print-tag () ; выводит содержимое слота tag класса Guideline

(printout t ?self:tag crlf)

)

Затем в нужном месте программы классу отправляется «сообщение»:

?guideline <- (object (is-a Guideline))

=>

(send ?guideline print-tag ())

Основные конструкторы COOL

Среда CLIPS имеет встроенные средства для осуществления логического вывода, но позволяет программисту управлять очередностью выполнения правил (посредством параметра `salience`).

Отличительной особенностью CLIPS являются конструкторы для создания баз знаний (БЗ):

defrule определение правил;

deffacts определение фактов;

deftemplate определение шаблона факта;

defglobal определение глобальных переменных;

deffunction определение функций;

defmodule определение модулей (совокупности правил);

defclass определение классов;

defintances определение объектов по шаблону, заданному *defclass* ;

defmessagehandler определение сообщений для объектов;

defgeneric создание заголовка родовой функции;

defmethod определение метода родовой функции.

Конструкторы не возвращают никаких значений, в отличие от функций, например:

(deftemplate person

(slot name)

(slot age)

(multislot friends))

Пример функции:

```
(deffunction factorial (?a)
  (if (or (not (integerp ? a)) (< ? a0)) then
      (printout t "Factorial Error!" crlf)
    else
      (if (= ? a0) then
          1
        else
          (* ? a (factorial ($-$ ? a1))))))
```

Правила в CLIPS состоят из предпосылок и следствия. Предпосылки также называют ЕСЛИ-частью правила, левой частью правила или LHS правила (left-hand side of rule). Следствие называют ТО-частью правила, правой частью правила или RHS правила (right-hand side of rule).

Пример правила представлен ниже:

```
(deftemplate data (slot x) (slot y))
(defrule twice
  (data (x ? x) (y>(*2 ? x)))
=>)
(assert (data (x2) (y4)); f-0
        (data (x3) (y9))); f-1
```

Здесь самая распространенная в CLIPS функция `assert` добавляет новые факты в список правил. В противоположность `assert` функция `retract` удаляет факты из списка фактов, например:

```
(defrule vis11
  ?doors < — (fit ? wdfit)
  (test (eq ? wdfit no))
=>
  (assert (EVIDENCE OF MAJOR ACCIDENT))
  (retract ? doors))
```

В этом правиле проверяется наличие факта `doors` и в случае его отсутствия факт `doors` удаляется из списка фактов задачи.

Функция `modify` является также весьма распространенной. Она позволяет в определенном факте поменять значение слота, например,

```
(deftemplate age (slot value))
```

(assert (age (value young)))

(modify 0 (value old))

Следующий пример описывает представление данных в виде фактов, объектов и глобальных переменных. Примеры фактов:

(voltage is 220 volt)

(meeting (subject "AI") (chief "Kuzin") (Room "3240"))

В первой строке приведен упорядоченный факт, во второй - неупорядоченный, в котором порядок слотов не важен.

CLIPS поддерживает следующие типы данных: integer, float, string, symbol, external-address, fact-address, instance-name, instance-address.

Пример integer: 594 23 +51 -17

Пример float: 594e2 23.45 +51.0 -17.5e-5

String — это строка символов, заключенная в двойные кавычки.

Пример string: "expert", "Phil Blake", "состояние \$-0\$", "quote=\"

CLIPS поддерживает следующие процедурные функции, реализующие возможности ветвления, организации циклов в программах и т.п.:

If оператор ветвления;

While цикл с предусловием;

loop-for-count итеративный цикл;

prong объединение действий в одной логической команде;

prong\$ выполнение набора действий над каждым элементом поля;

return прерывание функции, цикла, правила и т.д.;

break то же, что и *return*, но без возвращения параметров;

switch оператор множественного ветвления;

bind создание и связывание переменных.

Среди логических функций (возвращающих значения true или false) следует выделить следующие группы:

- функции булевой логики: and, or, not
- функции сравнения чисел: =, >, <
- предикативные функции для проверки принадлежности проверяемому типу: integerp, floatp, stringp, symbolp, pointerp (относится ли аргумент к xternal-address), numberp (относится ли аргумент к integer или float), lexemepr (относится ли аргумент к string или symbol), evenp (проверка целого на четность), oddp (проверка целого на нечетность), multifildp (является ли аргумент составным полем).
- Функции сравнения по типу и по значению: eq, neq

Среди математических функций следует выделить следующие группы:

- Стандартные: +, -, *, /, max, min, div (целочисленное деление), abs (абсолютное значение), float (преобразование в тип float), integer (преобразование в тип integer)
- Расширенные: sqrt (извлечение корня), round (округление числа), mod (вычисление остатка от деления)
- Тригонометрические: sin, sinh, cos, cosh, tan, tanh, acos, acosh, acot, acoth, asc, asch, asec, asech, asin, asinh, atan, atanh, cot, coth, csc, csch, sec, sech, deg-grad (преобразование из градусов в секторы), deg-rad (преобразование из градусов в радианы), grad-deg (преобразование из секторов в градусы), rad-deg (преобразование из радиан в градусы)
- Логарифмические: log, log10, exp, pi

Среди функций работы со строками следует назвать функции:

- str-cat объединение строк,
- sym-cat объединение строк в значение типа symbol,
- sub-string выделение подстроки,
- str-index поиск подстроки,
- eval выполнение строки в качестве команды CLIPS,
- build выполнение строки в качестве конструктора CLIPS,
- upcase преобразование символов в символы верхнего регистра,
- lowercase преобразование символов в символы нижнего регистра,
- str-compare сравнение строк,
- str-length определение длины строки,
- check-syntax проверка синтаксиса строки,
- string-to-field возвращение первого поля строки.
- Функции работы с составными величинами являются одной из отличительных особенностей языка CLIPS. В их число входят:
 - create\$ создание составной величины,
 - nth\$ получение элемента составной величины,
 - members поиск элемента составной величины,
 - subset\$ проверка одной величины на подмножество другой,
 - delete\$ удаление элемента составной величины,
 - explode\$ создание составной величины из строки,
 - implode\$ создание строки из составной величины,
 - subseq\$ извлечение подпоследовательности из составной величины,

- `replace$` замена элемента составной величины,
- `insert$` добавление новых элементов в составную величину,
- `first$` получение первого элемента составной величины,
- `rest$` получение остатка составной величины,
- `length$` определение числа элементов составной величины,
- `delete-member$` удаление элементов составной величины,
- `replace-member$` замена элементов составной величины.

Функции ввода-вывода используют следующие логические имена устройств:

- `stdin` устройство ввода,
- `stdout` устройство вывода,
- `wclips` устройство, используемое как справочное,
- `wdialog` устройство для отправки пользователю сообщений,
- `wdisplay` устройство для отображения правил, фактов и т.,п.,
- `werror` устройство вывода сообщений об ошибках,
- `wwarning` устройство для вывода предупреждений,
- `wtrase` устройство для вывода отладочной информации,
- Собственно функции ввода-вывода следующие:
- `open` открытие файла (виды доступа `r`, `w`, `r+`, `a`, `wb`),
- `close` закрытие файла,
- `printout` вывод информации на заданное устройство,
- `read` ввод данных с заданного устройства,
- `readline` ввод строки с заданного устройства,
- `format` форматированный вывод на заданное устройство,
- `rename` переименование файла,
- `remove` удаление файла.

Среди двух десятков команд CLIPS следует назвать основные команды при работе со средой CLIPS:

- `load` загрузка конструкторов из текстового файла,
- `load+` загрузка конструкторов из текстового файла без отображения,
- `reset` сброс рабочей памяти системы CLIPS,
- `clear` очистка рабочей памяти системы,
- `run` выполнение загруженных конструкторов,
- `save` сохранение созданных конструкторов в текстовый файл,
- `exit` выход из CLIPS.

Более подробно о CLIPS читайте в книге: Частиков А.П., Гаврилова Т.А., Бело Д.Л. Разработка экспертных систем. Среда clips.

Варианты для выполнения лабораторной работы

1. Разработка ПО (ИС)
2. Анализ требований к ПО
3. Проектирование архитектуры ПО
4. Программная реализация ПО
5. Тестирование и отладка ПО
6. Внедрение и поддержка (сопровождение) ПО
7. Проектирование компьютерных интерфейсов
8. CASE-средства
9. Контроль качества ПО
10. Моделирование бизнес-процессов
11. Свободный вариант, студент сам предлагает предметную область.

Темы 1, 4: Методы вывода на основе знаний Экспертные системы

Задание лабораторной работы SWI-Prolog

Задание 1. Построить базу фактов (не менее 50), используя сервис SWI-Prolog (<https://swish.swi-prolog.org/>) или аналогичный offline приложения согласно варианту. Сформулировать не менее 5 правил (хотя бы одно с использованием рекурсии). Сформулировать не менее 5 запросов с интерпретацией их на естественном языке.

Задание 2. Решить логическую задачу на прологе согласно варианту.

Отчет по лабораторной должен содержать описание программ, запросов и результатов (отчет можно сдавать в электронном виде).

Методические указания по выполнению лабораторной работы в SWI-Prolog

Наиболее известным языком декларативного (логического) программирования, реализующим модифицированную логику предикатов первого порядка, является PROLOG (англ. PROgramming LOGic - логическое программирование).

В 1965 году в работе «A machine oriented logic based on the resolution principle»³, опубликованной в 12 номере журнала «Journal of the ACM», Дж Робинсон представил метод автоматического поиска доказательства теорем в исчислении предикатов первого порядка, получивший название «**принцип резолюции**». На самом деле, идея данного метода была предложена Эрбраном в 1931 году, когда еще не было компьютеров (Herbrand, «Une methode de demonstration», These, Paris, 1931). Робинсон модифицировал

этот метод так, что он стал пригоден для автоматического (компьютерного) использования и разработал эффективный алгоритм унификации, составляющий базис его метода.

Идеи использования логики в качестве языка программирования зародилась в начале 1970-х годов. Первыми исследователями, которые занялись разработкой этой идеи, были Роберт Ковальски (Robert Kowalski) из Эдинбурга (теоретические основы, статьи 1971 и 1974 г.), Маартен ван Эмден (Maarten van Emden) из Эдинбурга (экспериментальная демонстрационная система) и Ален Колмероз (Alain Colmerauer) из Марселя (реализация, 1973 г.). В 1973 году «группа искусственного интеллекта» во главе с Аленом Колмероз создала в Марсельском университете программу, предназначенную для доказательства теорем. Эта программа использовалась при построении систем обработки текстов на естественном языке. Программа доказательства теорем получила название Prolog (фр. PROgrammation en LOGique) и послужила прообразом Пролога. Ходят легенды, что автором этого названия была жена Алена Колмероз. Программа была написана на Фортране и работала довольно медленно.

Популяризации Пролога во многом способствовали:

- эффективная реализация (интерпретатор/компилятор) этого языка для ЭВМ DEC-10 Дэвидом Д. Г. Уорреном (David D.H. Warren) из Эдинбурга в 1977 г. Послужила прототипом для многих последующих реализаций Пролога. Что интересно, компилятор был написан на самом Прологе. Эта реализация Пролога, известная как «эдинбургская версия», фактически стала первым и единственным стандартом языка;
- разработка Кларком и Маккейбом (Великобритания) в 1980 году версии для персональных ЭВМ;
- японский проект создания компьютеров V поколения. В конце 1978 г. Министерство внешней торговли и промышленности (МВТП) Японии поручило разработать проект интеллектуальных ЭВМ, специально созданному для этих целей Токийскому институту вычислительной техники (ICOT)⁴. Сердцем этих компьютеров должен был стать не арифметический процессор, а специально оптимизированный для работы с прологоподобными программами.

В 1995 году был опубликован официальный стандарт ISO⁵/IEC⁶ языка Пролог (ISO/IEC 13211-1 «Information technology - Programming languages - Prolog - Part 1: General core» - «Информационные технологии. Языки программирования. Пролог. Часть 1. Общее ядро»).

На сегодня существует довольно много реализаций Пролога. Наиболее известные из них следующие: BinProlog, AMZI-Prolog, Arity Prolog, CProlog, Micro Prolog, МПролог, Prolog-2, Quintus Prolog, SICTUS Prolog, Silogic iis Workbench, Strawberry Prolog, SWI-Prolog, Turbo Prolog (PDC Prolog, Visual Prolog), UNSW Prolog и т. д. В нашей стране были разработаны такие версии Пролога как Пролог-Д (С. Григорьев), Акторный Пролог (А. Морозов), а также Флэнг (А. Манцивода, В. Петухин).

SWI-Prolog является свободно распространяемой (англ. Free Software¹) реализацией (диалектом) языка программирования Пролог, которая практически полностью соответствует стандарту ISO²/IEC³ 13211-1 «Information technology - Programming languages - Prolog - Part 1: General core» (рус. «Информационные технологии. Языки программирования. Пролог. Часть 1. Общее ядро»).

SWI-Prolog развивается с 1987 года. Его создателем и основным разработчиком является Ян Вьелемакер (Jan Wielemaker). Название SWI происходит от Sociaal-Wetenschappelijke Informatica (гол. социально-научная информатика), первоначального названия группы в Амстердамском университете, где работает Вьелемакер.

SWI-Prolog позволяет разрабатывать приложения любой направленности, включая Web-приложения и параллельные вычисления, но основным направлением использования является разработка экспертных систем, программ обработки естественного языка, обучающих программ, интеллектуальных игр и т.п.

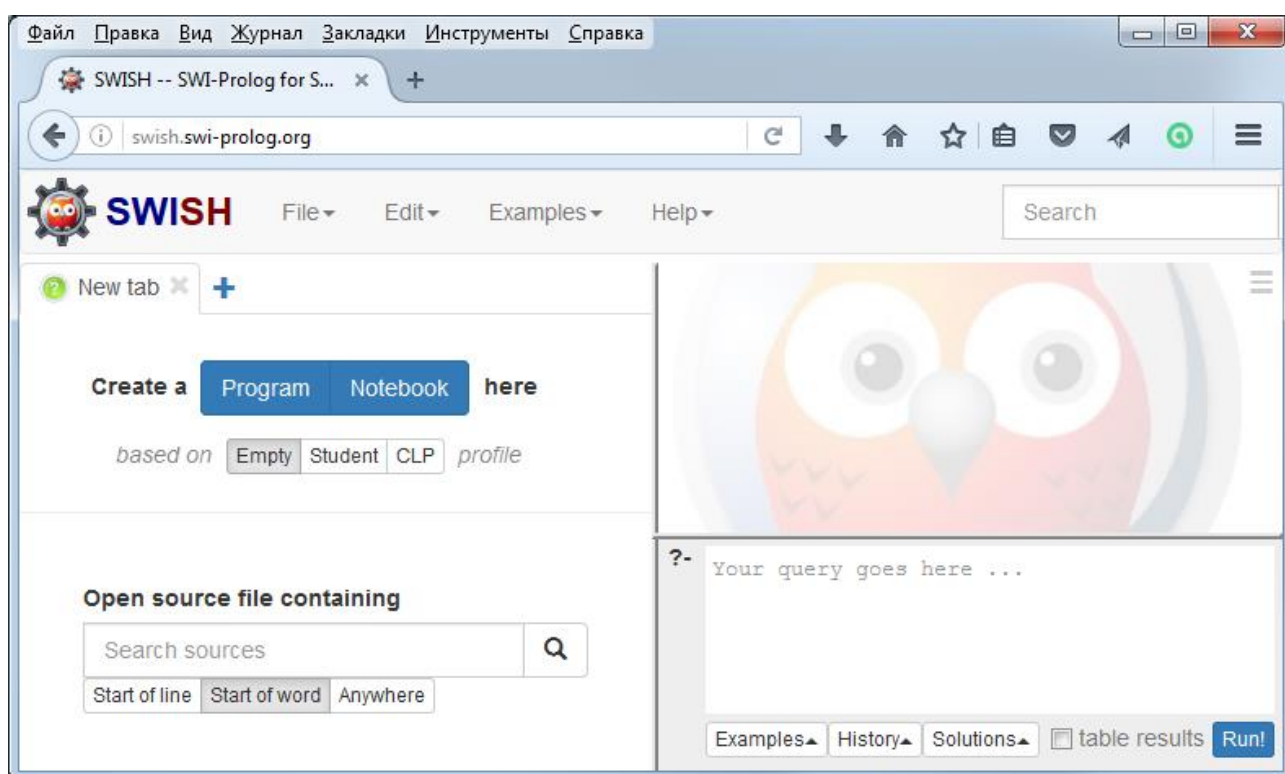


Рис.1. Стандартная online-среда программирования SWI-Prolog

Для перехода в режим редактирования и исполнения программ необходимо нажать на кнопку «Program» (см. рис.1).

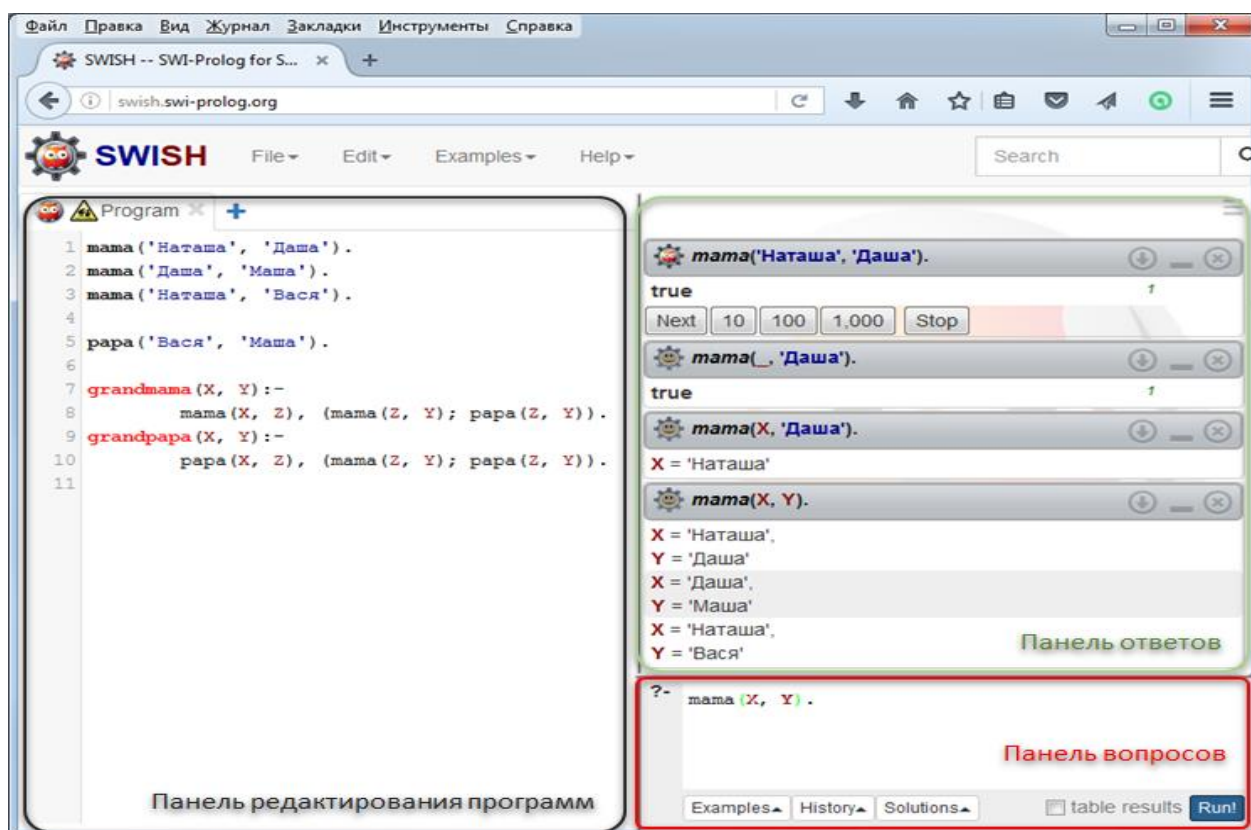


Рис.2. Режим редактирования и исполнения программ

В левой панели осуществляется редактирование программы, содержащей факты и правила.

В правой нижней панели выполняется набор вопросов и запуск их на исполнение с помощью кнопки «Run!».

В правой верхней панели интерпретатор SWI-Prolog выдает ответы на вопросы. В случае если на вопрос может быть получено более одного ответа, с помощью кнопок «Next», «10», «100» и «1,000» можно вывести на панель дополнительные ответы.

Некоторые операции и предикаты SWI-Prolog

Операция / Предикат	Назначение
true	Истина
fail, false	Ложь
=	Для переменной, стоящей слева от операции: - свободной - присваивание без преобразования (вычисления) выражения справа от операции; - связанной - сравнение без преобразования (вычисления) выражения справа от операции.

<, =<, >=, >	Арифметические (только для чисел) операции сравнения
:=	Арифметическое равенство
=\=	Арифметическое неравенство
is	Для переменной, стоящей слева от операции: - свободной - присваивание с преобразованием (вычислением) выражения справа от операции; - связанной - сравнение с преобразованием (вычислением) выражения справа от операции.
@<, @=<, @>=, @>	Операции сравнения для констант и переменных любого типа (чисел, строк, списков и т.д.)
==	Равенство констант и переменных любого типа
\==	Неравенство констант и переменных любого типа
not(A)	Отрицание логического выражения A
read(A)	Чтение значения с клавиатуры и присваивание его переменной A
write(A)	Печать A на экран с установкой курсора после последнего напечатанного символа
writeln(A)	Печать A на экран с переводом курсора в начало следующей строки
nl	Перевод курсора в начало следующей строки
repeat	Предикат, выдающий новое истинное значение при возврате. Передоказываемый предикат
!	Предикат (cut, сократить), запрещающий возврат далее той точки, где он стоит
assert(A), assertz(A)	Динамическое добавление факта (правила) в конец списка подобных фактов (правил) базы знаний (программы)
asserta(A)	Динамическое добавление факта (правила) в начало списка подобных фактов (правил) базы знаний
retract(A)	Удаление первого факта (правила) базы знаний
retractall(A)	Удаление всех фактов (правил) базы знаний с именем A

Информация по языку Пролог: <https://coollib.com/b/186558#nav>.

Варианты для выполнения лабораторной работы

Варианты для задания 1:

- 17) Родословная Романовых.
- 18) Царства в биологии.
- 19) Географические объекты (страны, столицы, материки, границы государств).
- 20) Состав компьютера.
- 21) Компьютерные сети.
- 22) Организационная структура университета.

- 23) Субъекты РФ.
- 24) Устройство автомобиля.
- 25) Родственные отношения (сваты, зять, свекры и т.д.)
- 26) Свободный вариант (студент предлагает тему и согласует ее с преподавателем).

Варианты для задания 2:

1) Трое, назовем их А, Б и В оказались вместе: один из России, другой из Финляндии, третий из Англии. Один из них увлекается математикой, другой - астрономией, а третий - литературой.

- a. А живет не в России, Б - не в Финляндии
- b. Тот, кто из России равнодушен к математике, а финн любит астрономию.
- c. Для Б не интересна литература.

Чем увлекается В и из какой он страны?

2) По древнему поверью, у каждого месяца есть свой камень-талисман. Так, июню, июлю и сентябрю соответствуют камни рубин, сапфир и жемчуг. Эти камни означают мудрость, здоровье и благополучие. У какого месяца какой камень-талисман и что он означает, если известно, что:

- жемчуг и рубин не соответствуют сентябрю;
- в июне и июле мудрости не наблюдается;
- здоровье не соответствует рубину;
- благополучие не относится к июню.

3) Три друга – Петр, Роман и Сергей учатся на математическом, физическом и химическом факультетах университета.

Если Петр математик, то Сергей не физик.

Если Роман не физик, то Петр – математик.

Если Сергей не математик, то Роман – химик.

Определите специальность Сергея.

4) На новогодний праздник три друга – Евгений, Николай, Алексей, выбрали себе костюмы трех богатырей: Ильи Муромца, Алеши Поповича, Добрыни Никитича.

Известно, что:

- Евгений – самый высокий.
- Выбравший костюм Добрыни Никитича меньше ростом, чем выбравший костюм Ильи Муромца.
- Алексею не подошел костюм Добрыни Никитича.

- Ни у одного из друзей имя не совпадает с именем богатырей, выбранных костюмов.

Какой костюм выбрал каждый из друзей?

5) Три друга заняли первое, второе и третье места в соревнованиях универсиады. Друзья — разной национальности, зовут их по-разному и любят они разные виды спорта.

Майкл предпочитает баскетбол и играет лучше чем американец. Израильтянин Саймон играет лучше теннисиста. Игрок в крикет занял первое место.

Кто является австралийцем? Каким видом спорта занимается Ричард?

6) Один из пяти братьев разбил окно.

Андрей сказал: Это или Витя, или Коля .

Витя сказал: Это сделал не я и не Юра .

Дима сказал: Нет, один из них сказал правду, а другой неправду .

Юра сказал: Нет, Дима ты не прав .

Их отец, которому, конечно можно доверять, уверен, что не менее трех братьев сказали правду.

Кто разбил окно?

7) В автомобильных гонках три первых места заняли Алеша, Петя и Коля. Какое место занял каждый из них, если Петя занял не второе и не третье место, а Коля - не третье?

8) Витя, Юра и Миша сидели на скамейке. В каком порядке они сидели, если известно, что Миша сидел слева от Юры, а Витя слева от Миши.

9) Трое ребят вышли гулять с собакой, кошкой и хомячком. Известно, что Петя не любит кошек и живет в одном подъезде с хозяйкой хомячка. Лена дружит с Таней, гуляющей с кошкой. Определить, с каким животным гулял каждый из детей.

10) Свободный вариант (студент предлагает тему и согласует ее с преподавателем).

Темы 1, 4: Методы вывода на основе знаний Экспертные системы

Задание лабораторной работы по рекомендательным системам

Цель работы: получение практических навыков построения рекомендательных систем на языке Python с использованием библиотеки Surprise.

Задание: используя программу Jupiter Notebook, язык программирования Python, библиотеку Surprise и др.:

- 1) загрузить набор данных согласно варианту, преобразовать данные в случае необходимости в соответствующий вид,

- 2) использовать метод согласно варианту для получения рекомендаций (прогнозных рейтингов),
- 3) получить значения оценок модели прогноза и интерпретировать результат,
- 4) вывести запрашиваемый в варианте результат (написать функцию с соответствующими входными параметрами и выводом, привести в отчёте 3 результата вызова функции с разными параметрами).

Отчёт по лабораторной работе должен содержать:

5. Фамилию и номер группы учащегося, задание, вариант.
6. Описание полученного набора данных.
7. Полное описание метода из варианта (алгоритм/формулы, выдаваемые значения, их интерпретация).
8. Пример вычислений (в ручную пошагово) по данным и методу из варианта (обязательно).
9. Скриншоты выполнения программы.
10. Интерпретация результатов (объяснение на конкретных данных)
11. Код с комментариями.

Методические указания по использованию Surprise для построения рекомендательных систем

Библиотека open source Surprise (Simple Python Recommended System Engine) позволяет создавать и тестировать системы рекомендаций на базе различных алгоритмов машинного обучения (прогнозирование рейтинга). Для построения системы необходимы данные об оценках пользователей для конкретных товаров (объектов).

Страница проекта на github: <https://github.com/NicolasHug/ Surprise>

Документация по библиотеке: <http://surpriselib.com/>

Установка библиотеки

Для установки используйте команду (если используете ее в блокноте поставьте ! в начале строки):

```
pip install scikit-surprise
```

Использование библиотеки

Библиотека Surprise содержит (Рисунок 1):

- классы: Trainset, Reader;

- модули: `dataset` (набор данных), `accuracy` (критерии оценки), `similarities` (метрики сходства), `dump` (дамп результатов алгоритма);
- пакеты: `prediction_algorithms` (алгоритмы предсказания), `model_selection` (модель выбора / кросс-валидации).

Для построения рекомендательной системы необходимо (Рисунок 2):

- 1) Загрузить данные (для чтения используется класс `Reader`, для хранения модуль `dataset`);
- 2) Выбрать алгоритм предсказания (прогноза) и определить его параметры (пакет `prediction_algorithms`, алгоритмы, а также модуль `similarities` и метрики сходства.
- 3) Разбить при необходимости набор на тестовое и обучающее множества и применить к ним выбранный алгоритм (`model_selection`),
- 4) Оценить полученную прогнозную модель (модуль `accuracy` и критерии оценки Таблица 3),
- 5) Получить рекомендации по запросу на базе построенной модели (`model_selection`).

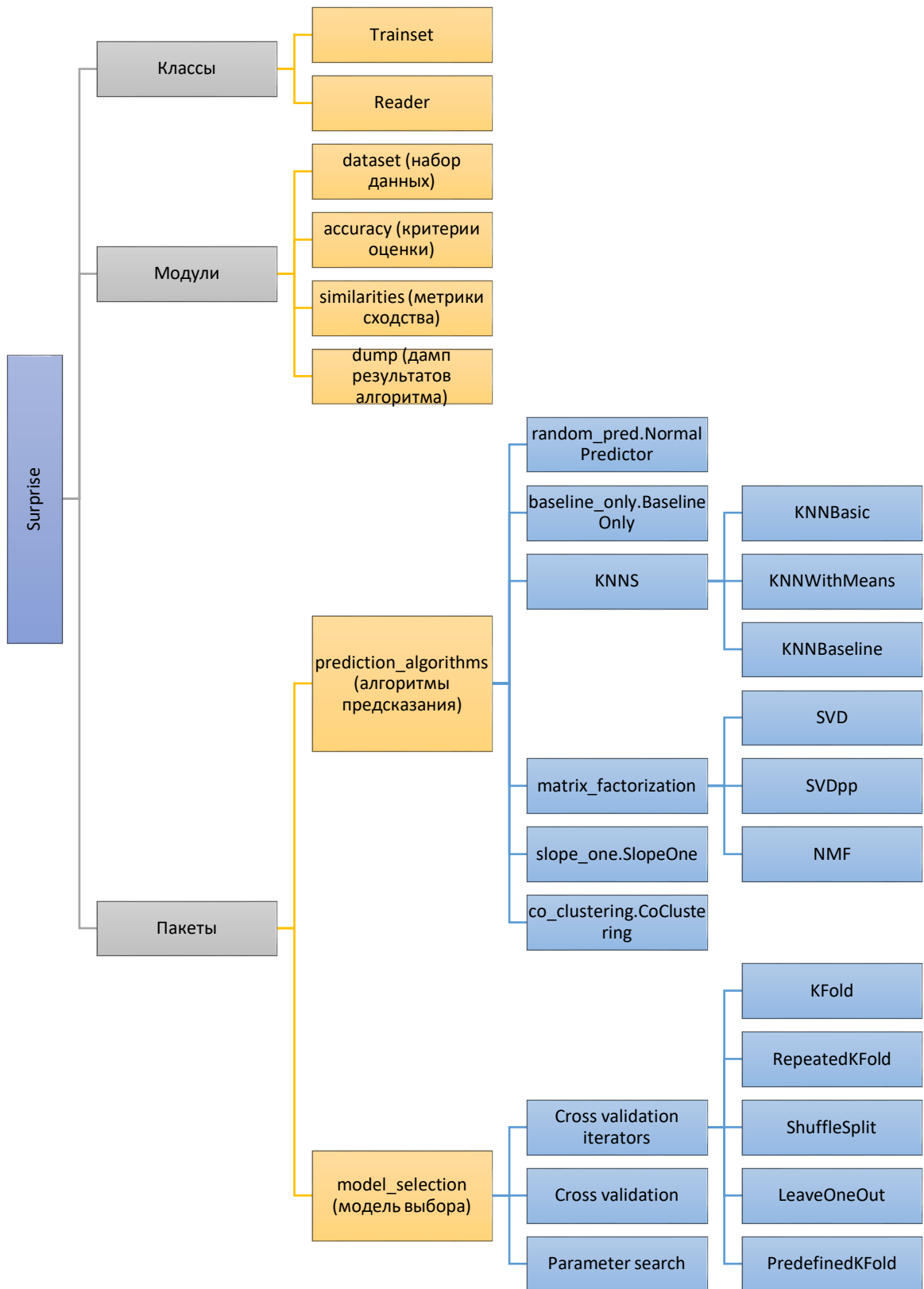


Рисунок 1 – Структура библиотеки

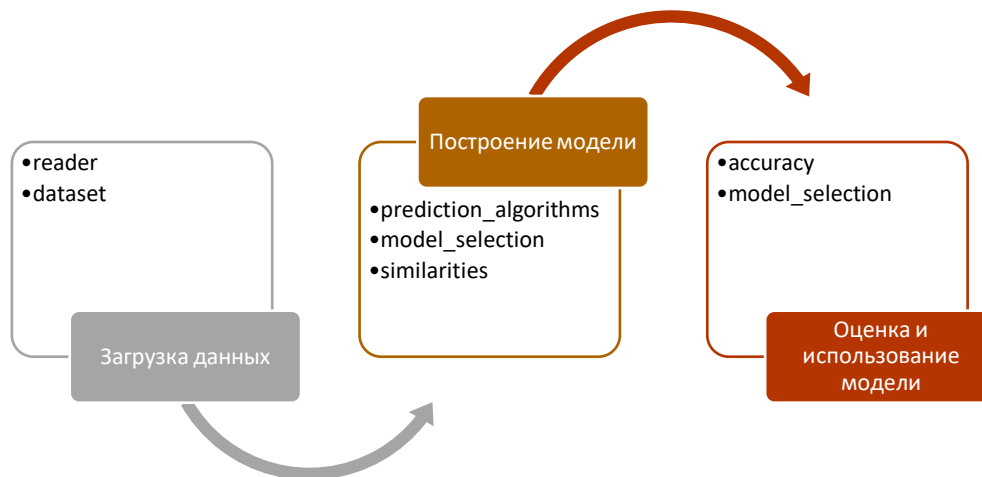


Рисунок 2 – Схема использования элементов библиотеки

Загрузка данных

Данные при использовании библиотека можно:

- воспользоваться встроенными данными,
- загрузить данные из внешнего источника (например файла CSV)
- загрузить данные из DataFrame Pandas (бывает полезно, если необходима подготовка данных).

Для использования данных нужно воспользоваться элементами библиотеки Dataset и Reader.

Пример загрузки встроенного набора данных:

```

from surprise import Dataset
# Загружать набор данных movielens по умолчанию
data = Dataset.load_builtin('ml-100k')
  
```

Пример загрузки данных из файла:

```

from surprise import Dataset
from surprise import Reader
import os
# Укажите путь к файлу
file_path = os.path.expanduser('data.csv')
# Скажите читателю, какой формат текста
reader = Reader(line_format='user item rating', sep=',')
# Загрузить данные
data = Dataset.load_from_file(file_path, reader=reader)
  
```

Пример загрузки данных из DataFrame Pandas:

```

import pandas as pd
from surprise import Dataset
from surprise import Reader
# считываем данные из файла
fix_df1 = pd.read_csv('d://ratings.csv')
# загружаем данные из набора Pandas в набор Surprise
reader = Reader(rating_scale=(0, 5))
data = Dataset.load_from_df(fix_df[['userId', 'movieId', 'rating']], reader)
  
```

Построение модели

При построении модели необходимо определить, какой алгоритм будет строить предсказательную модель и как он будет работать с данными. Surprise предоставляет различные инструменты для запуска процедур перекрестной проверки и поиска лучших параметров для алгоритма прогнозирования. Библиотека позволяет осуществлять 3 способа, реализованных в пакете `model_selection`:

- автоматизированная кросс-валидация (класс `surprise.model_selection.validation.cross_validate`);
- итерационная кросс-валидация (модуль `model_selection.split`);
- параметрический поиск (класс `surprise.model_selection.search.GridSearchCV`).

Подробнее см. https://surprise.readthedocs.io/en/stable/model_selection.html

Наиболее простой способ – это использование метода `cross_validate` с одним из алгоритмов построения модели из пакета `prediction_algorithms` (Таблица 1).

Таблица 1 – Алгоритмы предсказания (рейтинга)

Имя класса алгоритма	Объяснение
random_pred.NormalPredictor	Случайно дать прогнозируемое значение в соответствии с характеристиками распределения обучающего набора
baseline_only.BaselineOnly	Учитывая пользователя и предмет, дать оценку на основе базовой линии
knns.KNNBasic	Самая базовая совместная фильтрация
knns.KNNWithMeans	Внедрение совместной фильтрации с учетом среднего значения рейтинга каждого пользователя
knns.KNNBaseline	Совместная фильтрация с учетом базовых рейтингов
matrix_factorization.SVD	Реализация SVD
matrix_factorization.SVDpp	SVD ++, то есть LFM + SVD
matrix_factorization.NMF	Совместная фильтрация на основе матричной декомпозиции
slope_one.SlopeOne	Простой, но точный алгоритм совместной фильтрации
co_clustering.CoClustering	Алгоритм совместной фильтрации, основанный на совместной кластеризации

Пример:

```
### Использовать NormalPredictor
from surprise import NormalPredictor
algo = NormalPredictor()
perf = cross_validate(algo, data, measures=['RMSE', 'MAE'], cv=3)

### Использовать BaselineOnly
```

```

from surprise import BaselineOnly
algo = BaselineOnly()
perf = cross_validate(algo, data, measures=['RMSE', 'MAE'], cv=3)

### Использовать базовую версию совместной фильтрации
from surprise import KNNBasic, evaluate
algo = KNNBasic()
perf = cross_validate(algo, data, measures=['RMSE', 'MAE'], cv=3)

### Использовать среднюю совместную фильтрацию
from surprise import KNNWithMeans, evaluate
algo = KNNWithMeans()
perf = cross_validate(algo, data, measures=['RMSE', 'MAE'], cv=3)

### Использовать базовую линию совместной фильтрации
from surprise import KNNBaseline, evaluate
algo = KNNBaseline()
perf = cross_validate(algo, data, measures=['RMSE', 'MAE'], cv=3)

### Используйте SVD
from surprise import SVD, evaluate
algo = SVD()
perf = cross_validate(algo, data, measures=['RMSE', 'MAE'], cv=3)

### Используйте SVD ++
from surprise import SVDpp, evaluate
algo = SVDpp()
perf = cross_validate(algo, data, measures=['RMSE', 'MAE'], cv=3)

### Используйте NMF
from surprise import NMF
algo = NMF()
perf = cross_validate(algo, data, measures=['RMSE', 'MAE'], cv=3)

```

Для сравнения строк или столбцов матрицы (расчета схожести объектов или пользователей) могут быть использованы метрики модуля `similarities` (Таблица 2).

Таблица 2 - Метрика сходства

Метрика	Метрическое описание
cosine	Вычислите косинусное сходство между всеми парами пользователей (или предметов).
msd	Рассчитайте среднеквадратичную разницу между всеми пользователями (или предметами).
pearson	Рассчитайте коэффициент корреляции Пирсона между всеми парами пользователей (или предметов).
pearson baseline	Рассчитайте (уменьшенный) коэффициент корреляции Пирсона между всеми парами пользователей (или предметов), используя базовую линию для центрирования, а не для среднего значения.

Пример:

```

# Загружать набор данных movielens по умолчанию
data = Dataset.load_builtin('ml-100k')
trainset = data.build_full_trainset()
#Используйте метод pearson_baseline для вычисления сходства.
#Ложное вычисление сходства на основе элемента.

```

```
#Этот пример - сходство между фильмами
sim_options = {'name': 'pearson_baseline', 'user_based': False}
#Использование алгоритма KNNBaseline
algo = KNNBaseline(sim_options=sim_options)
#Тренировочная модель
algo.fit(trainset)
```

Оценка модели

Для определения насколько модель обучена необходимо ее оценить и проверить на тестовом множестве (если выполнялось разделение набора данных). В библиотеке используются критерии оценки (Таблица 3), реализованные в модуле accuracy.

Таблица 3 - Критерии оценки

Критерии	Название
rmse	Рассчитать RMSE (среднеквадратическая ошибка)
mae	Рассчитать MAE (средняя абсолютная ошибка)
fcp	Рассчитать FCP (оценка координационных пар)

Пример

```
from surprise import SVD
from surprise import Dataset
from surprise.model_selection import KFold
from surprise import accuracy

data = Dataset.load_builtin('ml-100k')
kf = KFold(n_splits=3)
algo = SVD()
for trainset, testset in kf.split(data):
    algo.fit(trainset)
    predictions = algo.test(testset)
    accuracy.rmse(predictions, verbose=True)
```

Пример кода

1) Пример использования разных алгоритмов прогноза

```
# загружаем библиотеки
import pandas as pd
from surprise import Dataset
from surprise import Reader
from surprise.model_selection import cross_validate
from surprise import NormalPredictor
from surprise import SVD
from surprise import accuracy

# считываем данные из файла
fix_df1 = pd.read_csv('d://ratings.csv')
#ограничиваем набор для учебных целей
fix_df=fix_df1[:10000]
# загружаем данные из набора Pandas в набор Surprise
reader = Reader(rating_scale=(0, 5))
data = Dataset.load_from_df(fix_df[['userId', 'movieId', 'rating']], reader)
# применить алгоритм SVD для загруженного набора данных и рассчитать оценки
RMSE, MAE
```

```

cross_validate(SVD(), data, measures=['RMSE', 'MAE'], cv=5, verbose=True)
# применить алгоритм NormalPredictor к набору данных
cross_validate(NormalPredictor(), data, cv=2)

# делим на обучающую и тестовую выборки
trainset, testset = train_test_split(data, test_size=.25)
# применить алгоритм SVD
al=SVD().fit(trainset)
out2 = al.test(testset)
accuracy.rmse(out2)

# применить алгоритм SVD с указанием параметров
al2=SVD(n_factors=80, n_epochs=20, lr_all=0.005, reg_all=0.2)
al2.fit(trainset)
out4 = al2.test(testset)
df_pred = pd.DataFrame(out4, columns=['user_id', 'isbn', 'actual_rating',
'pred_rating', 'details'])
df_pred['impossible'] = df_pred['details'].apply(lambda x:
x['was_impossible'])
df_pred['pred_rating_round'] = df_pred['pred_rating'].round()
df_pred['abs_err'] = abs(df_pred['pred_rating'] - df_pred['actual_rating'])
df_pred.drop(['details'], axis=1, inplace=True)

```

2) Пример с выводом рекомендованных значений для всех пользователей

```

from collections import defaultdict

from surprise import SVD
from surprise import Dataset
from surprise import Reader

import pandas as pd

def get_top_n(predictions, n=10):
    #top-N рекомендация для каждого пользователя из набора.
    top_n = defaultdict(list)
    for uid, iid, true_r, est, _ in predictions:
        top_n[uid].append((iid, est))
    for uid, user_ratings in top_n.items():
        user_ratings.sort(key=lambda x: x[1], reverse=True)
        top_n[uid] = user_ratings[:n]
    return top_n

# считываем данные из файла
fix_df = pd.read_csv('d://ratings.csv')

# загружаем данные из набора Pandas в набор Surprise
reader = Reader(rating_scale=(0, 5))
data = Dataset.load_from_df(fix_df[:10000][['userId', 'movieId', 'rating']],
reader)

trainset = data.build_full_trainset()
algo = SVD()
algo.fit(trainset)

# Вычисляем прогнозные значения для тренировочной выборки
testset = trainset.build_anti_testset()
predictions = algo.test(testset)

top_n = get_top_n(predictions, n=10)

# Выводим рекомендации

```

```
for uid, user_ratings in top_n.items():
    print(uid, [iid for (iid, _) in user_ratings])
```

3) Пример использования метод `pearson_baseline` для вычисления сходства¹

```
from __future__ import (absolute_import, division, print_function,
                        unicode_literals)

import os
import io
from surprise import KNNBaseline
from surprise import Dataset

import logging

logging.basicConfig(level=logging.INFO,
                    format='%(asctime)s %(filename)s[line:%(lineno)d]
%(levelname)s %(message)s',
                    datefmt='%a, %d %b %Y %H:%M:%S')

# Обучение рекомендуемой модели Шаги: 1
def getSimModle():
    # Загружать набор данных movielens по умолчанию
    data = Dataset.load_builtin('ml-100k')
    trainset = data.build_full_trainset()
    #Используйте метод pearson_baseline для вычисления сходства. Ложное
    #вычисление сходства на основе элемента. Этот пример - сходство между фильмами
    sim_options = {'name': 'pearson_baseline', 'user_based': False}
    ## Использование алгоритма KNNBaseline
    algo = KNNBaseline(sim_options=sim_options)
    #Тренировочная модель
    algo.fit(trainset)
    return algo

# Получить взаимное сопоставление идентификатора с именем Шаг: 2
def read_item_names():
    """
        Получить сопоставление названия фильма с идентификатором фильма и
        идентификатора фильма с названием фильма
    """
    file_name = (os.path.expanduser('~') +
                 '/.surprise_data/ml-100k/ml-100k/u.item')
    rid_to_name = {}
    name_to_rid = {}
    with io.open(file_name, 'r', encoding='ISO-8859-1') as f:
        for line in f:
            line = line.split('|')
            rid_to_name[line[0]] = line[1]
            name_to_rid[line[1]] = line[0]
    return rid_to_name, name_to_rid

# Порекомендуйте похожие фильмы по мотивам ранее обученной модели. Шаги: 3
def showSimilarMovies(algo, rid_to_name, name_to_rid):
    # Получить raw_id фильма История игрушек (1995)
    toy_story_raw_id = name_to_rid['Toy Story (1995)']
    logging.debug('raw_id=' + toy_story_raw_id)
    #Преобразуйте raw_id фильма во внутренний идентификатор модели.
    toy_story_inner_id = algo.trainset.to_inner_iid(toy_story_raw_id)
    logging.debug('inner_id=' + str(toy_story_inner_id))
    #Получить рекомендуемые фильмы через модель здесь 10
```

¹ <https://russianblogs.com/article/8373229539/>


```

toy_story_neighbors = algo.get_neighbors(toy_story_inner_id, 10)
logging.debug('neighbors_ids=' + str(toy_story_neighbors))
#Преобразовать внутренний идентификатор модели в фактический
идентификатор фильма
neighbors_raw_ids = [algo.trainset.to_raw_iid(inner_id) for inner_id in
toy_story_neighbors]
#По списку идентификаторов фильмов или списку рекомендаций по фильмам
neighbors_movies = [rid_to_name[row_id] for row_id in neighbors_raw_ids]
print('The 10 nearest neighbors of Toy Story are:')
for movie in neighbors_movies:
    print(movie)

if __name__ == '__main__':
    # Получить взаимное сопоставление от идентификатора к имени
    rid_to_name, name_to_rid = read_item_names()

    # Модель рекомендаций по обучению
    algo = getSimModle()

    ## Показать похожие фильмы
    showSimilarMovies(algo, rid_to_name, name_to_rid)

```

Варианты для выполнения лабораторной работы

Вариант	Набор данных (закачать любой подходящий набор данных с ресурса)	Метод прогноза	Вывод
1	MovieLens 25M Dataset – набор рейтинговых данных с веб-сайта MovieLens, который описывает 5-звездочные рейтинги и действия с произвольным тегированием по более 60 тысячам фильмов от 1,5 миллионов пользователей с 1995 по 2019 годы. https://grouplens.org/datasets/movielens/	random_pred.Normal Predictor	Топ-10 рекомендованных объектов (товаров) по пользователю
2	Netflix Prize - многовариантный датасет временных рядов, который использовался в конкурсе Netflix Prize с рейтингами примерно 100 миллионов фильмов. В наборе данных более 480000 пользователей, каждый из которых промаркирован уникальным целочисленным идентификатором. http://academictorrents.com/details/9b13183dc4d60676b773c9e2cd6de5e5542cee9a	baseline_only.BaselineOnly	5 наиболее похожих пользователей для заданного пользователя
3	Book-Crossing – датасет с рейтингами около 300 тысяч миллионов книг и обезличенными демографическими данными о более 250 тысячах их читателей. http://www2.informatik.uni-freiburg.de/~cziegler/BX/	knns.KNNBasic	Топ-10 товаров (объектов), конкретного пользователя (по реальному рейтингу)
4	Amazon Review Data – многомиллионный набор	knns.KNNWithMean	Топ-10

	обзоров, рейтингов и метаданных продуктов (описание, категория, цена, бренд, характеристики, фото), а также данные о просмотре ссылок. https://nijianmo.github.io/amazon/index.html	s	рекомендованных объектов (товаров) по пользователю
5	REKKO CHALLENGE – набор данных от онлайн-кинотеатра ОККО для конкурса по разработке рекомендательных систем 2019 года. https://boosters.pro/championship/rekko_challenge/data	knns.KNNBaseline	5 наиболее похожих пользователей для заданного пользователя
6	LastFM – датасет содержит информацию о социальных сетях, тегах и прослушивании музыкальных исполнителей от 2 тысяч пользователей онлайн-музыки Last.fm. https://files.grouplens.org/datasets/hetrec2011/	matrix_factorization.SVD	Топ-10 товаров (объектов), конкретного пользователя (по реальному рейтингу)
7	Social Network Influencer – датасет Peerindex, который включает стандартную задачу изучения парных предпочтений. Здесь каждая точка данных описывает двух человек и предварительно рассчитанные стандартизованные функции на основе активности в Twitter: объем взаимодействий, количество подписчиков и пр. для каждого человека. https://www.kaggle.com/c/predict-who-is-more-influential-in-a-social-network/data	matrix_factorization.SVDpp	Топ-10 рекомендованных объектов (товаров) по пользователю
8	Million Song Dataset – набор звуковых фич и метаданных для миллиона современных музыкальных треков от Echo Nest. http://millionsongdataset.com/	matrix_factorization.NMF	5 наиболее похожих пользователей для заданного пользователя
9	Free Music Archive (FMA) – набор легальных аудиозаписей для задач анализа музыки - просмотр, поиск и организация коллекций. https://github.com/mdeff/fma	slope_one.SlopeOne	Топ-10 товаров (объектов), конкретного пользователя (по реальному рейтингу)
10	Steam Video Games - набор данных о действиях пользователей и их характеристиках от самого популярного хаба видеоигр, PC Gaming Steam https://www.kaggle.com/tamber/steam-video-games/data	co_clustering.CoClustering	Топ-10 рекомендованных объектов (товаров) по пользователю
11	Ta-Feng – набор данных о покупках от ACM RecSys	random_pred.Normal	5 наиболее

	по 23+ тысяч товаров, от продуктов питания и канцелярских товаров до мебели. http://www.bigdatalab.ac.cn/benchmark/bm/dd?data=Ta-Feng	Predictor	похожих пользователей для заданного пользователя
12	Beiren – данные о реальных покупках более миллиона человек в супермаркетах Китая за период с 2012 по 2013 год. http://www.bigdatalab.ac.cn/benchmark/bm/dd?data=Beiren	baseline_only.BaselineOnly	Топ-10 товаров (объектов), конкретного пользователя (по реальному рейтингу)
13	MovieLens 25M Dataset – набор рейтинговых данных с веб-сайта MovieLens, который описывает 5-звездочные рейтинги и действия с произвольным тегированием по более 60 тысячам фильмов от 1,5 миллионов пользователей с 1995 по 2019 годы. https://grouplens.org/datasets/movielens/	knnns.KNNBasic	Топ-10 рекомендованных объектов (товаров) по пользователю
14	Jester - Анонимные данные о рейтингах шуток (анекдотов) из системы Jester. https://goldberg.berkeley.edu/jester-data/	knnns.KNNWithMeans	5 наиболее похожих пользователей для заданного пользователя
15	REKKO CHALLENGE – набор данных от онлайн-кинотеатра ОККО для конкурса по разработке рекомендательных систем 2019 года. https://boosters.pro/championship/rekko_challenge/data	knnns.KNNBaseline	Топ-10 товаров (объектов), конкретного пользователя (по реальному рейтингу)
16	MovieLens 25M Dataset – набор рейтинговых данных с веб-сайта MovieLens, который описывает 5-звездочные рейтинги и действия с произвольным тегированием по более 60 тысячам фильмов от 1,5 миллионов пользователей с 1995 по 2019 годы. https://grouplens.org/datasets/movielens/	matrix_factorization.SVD	Топ-10 рекомендованных объектов (товаров) по пользователю
17	Netflix Prize - многовариантный датасет временных рядов, который использовался в конкурсе Netflix Prize с рейтингами примерно 100 миллионов фильмов. В наборе данных более 480000 пользователей, каждый из которых промаркирован уникальным целочисленным идентификатором. http://academictorrents.com/details/9b13183dc4d60676b773c9e2cd6de5e5542cee9a	matrix_factorization.SVDpp	5 наиболее похожих пользователей для заданного пользователя

18	Book-Crossing – датасет с рейтингами около 300 тысяч миллионов книг и обезличенными демографическими данными о более 250 тысячах их читателей. http://www2.informatik.uni-freiburg.de/~ctiegl/BX/	matrix_factorization.NMF	Топ-10 товаров (объектов), конкретного пользователя (по реальному рейтингу)
19	MovieLens 25M Dataset – набор рейтинговых данных с веб-сайта MovieLens, который описывает 5-звездочные рейтинги и действия с произвольным тегированием по более 60 тысячам фильмов от 1,5 миллионов пользователей с 1995 по 2019 годы. https://grouplens.org/datasets/movielens/	slope_one.SlopeOne	Топ-10 рекомендованных объектов (товаров) по пользователю
20	Netflix Prize - многовариантный датасет временных рядов, который использовался в конкурсе Netflix Prize с рейтингами примерно 100 миллионов фильмов. В наборе данных более 480000 пользователей, каждый из которых промаркирован уникальным целочисленным идентификатором. http://academictorrents.com/details/9b13183dc4d60676b773c9e2cd6de5e5542cee9a	co_clustering.CoClustering	5 наиболее похожих пользователей для заданного пользователя

* При несогласованности указаний в варианте, сделать соответствующие пояснения и изменения условий.

РЕКОМЕНДУЕМАЯ ЛИТЕРАТУРА И ИНФОРМАЦИОННОЕ ОБЕСПЕЧЕНИЕ

Список рекомендуемой литературы

основная

- 1) Малышева, Е. Н. Экспертные системы : учебное пособие по специальности 080801 «Прикладная информатика (в информационной сфере)» / Е. Н. Малышева. — Кемерово : Кемеровский государственный институт культуры, 2010. — 86 с. — ISBN 2227-8397. — Текст : электронный // Электронно-библиотечная система IPR BOOKS : [сайт]. — URL: <http://www.iprbookshop.ru/22126.html>
- 2) Богданова, Е. А. Инженерия знаний : учебное пособие / Е. А. Богданова. — Самара : Поволжский государственный университет телекоммуникаций и информатики, 2016. — 103 с. — ISBN 2227-8397. — Текст : электронный // Электронно-библиотечная система IPR BOOKS : [сайт]. — URL: <http://www.iprbookshop.ru/71833.html>
- 3) Смагин, А. А. Интеллектуальные информационные системы : учеб. пособие для вузов / А. А. Смагин, С. В. Липатова, А. С. Мельниченко ; УлГУ, Фак. математики и информ. технологий, Каф. телекоммуникац. технологий и сетей. - Ульяновск : УлГУ, 2010.- URL: <ftp://10.2.96.134/Text/smagin2.pdf>

дополнительная

- 4) Болотова Л.С., Системы искусственного интеллекта: модели и технологии, основанные на знаниях : учебник / Л.С. Болотова. - М. : Финансы и статистика, 2012. - 664 с. - ISBN 978-5-279-03530-4 - Текст : электронный // ЭБС "Консультант студента": [сайт]. - URL : <http://www.studentlibrary.ru/book/ISBN9785279035304.html>
- 5) Яковина И.Н., Системы искусственного интеллекта. Модуль "Модели и методы извлечения знаний" / Яковина И.Н. - Новосибирск : Изд-во НГТУ, 2014. - 54 с. - ISBN 978-5-7782-2587-9 - Текст : электронный // ЭБС "Консультант студента" : [сайт]. - URL : <http://www.studentlibrary.ru/book/ISBN9785778225879.html>
- 6) Исаев, С. В. Интеллектуальные системы : учебное пособие / С. В. Исаев, О. С. Исаева. — Красноярск : Сибирский федеральный университет, 2017. — 120 с. — ISBN 978-5-7638-3781-0. — Текст : электронный // Электронно-библиотечная система IPR BOOKS : [сайт]. — URL: <http://www.iprbookshop.ru/84365.html>

учебно-методическая

- 7) Интерактивные системы Scilab, Matlab, Mathcad : учебное пособие / И. Е. Плещинская, А. Н. Гитов, Е. Р. Бадертдинова, С. И. Дуев. — Казань : Казанский национальный исследовательский технологический университет, 2014. — 195 с. — ISBN 978-5-7882-1715-4. — Текст : электронный // Электронно-библиотечная система IPR BOOKS : [сайт]. — URL: <http://www.iprbookshop.ru/62173.html>

Программное обеспечение

1. Оболочка экспертных систем CLIPS (open source),
2. Online-редактор SWI-Prolog (open source),
3. Online-сервис Coggle (<https://coggle.it/>) (open source).
4. Anaconda (open source).