МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ «МИСиС»

ИНСТИТУТ *ИТКН*

КАФЕДРА *ИНЖЕНЕРНОЙ КИБЕРНЕТИКИ*

НАПРАВЛЕНИЕ *09.04.03 ПРИКЛАДНАЯ ИНФОРМАТИКА*

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА МАГИСТРА

на тему: Математическое и программное обеспечение для решения каузально-логических игр с использованием технологий самообучения

Студент *Д. А. Новицкий*

Руководитель работы *А. С. Кожаринов*

Нормоконтроль проведен *А. С. Островская*

Проверка на заимствования проведена *Г. В. Кружкова*

Работа рассмотрена кафедрой и допущена к защите в ГЭК

Заведующий кафедрой *А. Р. Ефимов*

Директор института *С. В. Солодов*

Москва, июнь 2022 г.

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ «МИСиС»

УТВЕРЖДАЮ

Институт ИТКН

Кафедра Инженерной Кибернетики **Зав. кафедрой:** Ефимов А. Р.

Направление 09.04.03 Прикладная информатика 24 декабря 2021 г.

ЗАДАНИЕ
НА ВЫПОЛНЕНИЕ ВЫПУСКНОЙ КВАЛИФИКАЦИОННОЙ РАБОТЫ БАКАЛАВРА

Студенту группы МПИ-20-4-2, Новицкому Дмитрию Александровичу

1. Тема работы: «Математическое и программное обеспечение для решения каузально-логических игр с использованием технологий самообучения».
2. Цель работы: Разработка комплекса алгоритмов с элементами самообучения и их программной реализации для независимого от человека поиска эффективного решения каузально-логических игр (на примере игры «Minesweeper»/«Сапёр»).
3. Исходные данные: свободно распространяемые данных о полях игры «Сапёр», имеющих детерминированное решение.
4. Основная литература, в том числе:
   - 4.1. Монография, учебники и т. п.
     - 4.3.1. В. В. Круглов, М. И. Дли, Р. Ю. Голунов. Нечёткая логика и искусственные нейронные сети. Изд. Физматлит, 2001.
     - 4.3.2. Г. Нойнер, Ю. К. Бабанский. Педагогика. М.: Педагогика. 1984.
   - 4.2. Научные работы:
     - 4.2.1. М. Г. Доррер. Психологическая интуиция искусственных нейронных сетей. Сибирский государственный технологический университет. Красноярск, 1998.
     - 4.2.2. А. Д. Комаров. Осторожно, мины! Алгоритм решения игры Сапёр. Компьютерные инструменты в образовании. № 5, 2006.
     - 4.2.3. Е. Ю. Корлякова, М. О. Корлякова. Подход к разработке самообучающегося алгоритма игры в «Сапёр». Наукоёмкие технологии в приборо- и машиностроении и развитие инновационной деятельности в вузе. Материалы Всероссийской научно-технической конференции. Том 2. Изд. КФ МГТУ им. Баумана, Калуга, 2016. С. 23-24.

5. Перечень основных этапов исследования и форма промежуточной отчетности по каждому этапу:

5.1. Литературный обзор предметной области

5.2. Формулировка содержательной постановки задачи

5.3. Формулировка математической постановки задачи

5.4. Разработка методов решения задачи

5.5. Анализ и оценка эффективности разработанных методов

5.6. Разработка программной реализации

5.7. Сбор и анализ полученных результатов

6. Аппаратура и методики, которые должны быть использованы в работе: Методы искусственного интеллекта, машинное обучение, нейронная сеть, экспертная система.

7. Использование ЭВМ: Разработка программы с использованием языка программирования Python.

8. Перечень (примерный) основных вопросов, которые должны быть рассмотрены и проанализированы в литературном обзоре:

8.1. Анализ и выбор подходящего класса логических задач и примера задачи для построения системы с элементами самообучения

8.2. Обзор основных методов искусственного интеллекта

8.3. Сравнение процессов обучения человека и машины

8.4. Обзор методов решения выбранного класса логических задач и примера для реализации

9. Перечень (примерный) графического и иллюстрированного материала:

9.1. Описание предметной области

9.2. Содержательная постановка задачи

9.3. Математическая постановка задачи

9.4. Функциональная схема

9.5. Схемы основных алгоритмов

9.6. Анализ полученных результатов

10. Руководитель работы _____ доцент, к.т.н., Кожаринов А. С.

(подпись) (Должность, звание, ф.и.о.)

Дата выдачи задания 24.12.2021

Задание принял к исполнению студент _____ Новицкий Д. А

(подпись) (ф.и.о.)

РЕФЕРАТ

Выпускная квалификационная работа выполнена на 85 страницах, содержит 47 рисунков, 29 формул, 13 таблиц, список использованных источников из 33 пунктов.

Ключевые слова: СИСТЕМА С ЭЛЕМЕНТАМИ САМООБУЧЕНИЯ, ИСКУССТВЕННЫЙ ИНТЕЛЛЕКТ, КАУЗАЛЬНО-ЛОГИЧЕСКАЯ ИГРА, «САПЁР», СХЕМА

Выпускная квалификационная работа посвящена разработке системы, содержащей элементы самообучения, способной находить решения игры «Сапёр» с помощью нескольких методов и способной оценивать эффективность данных методов.

Целью данной работы является разработка математического, алгоритмического и программного обеспечения, позволяющего находить наиболее эффективное решение игры «Сапёр».

В ходе работы произведён обзор логических задач, обзор основных методов искусственного интеллекта, представлен сравнительный обзор процессов обучения человека и машины, а также рассмотрены методы решения игры «Сапёр».

Результатом работы является программа для решения игры «Сапёр». Данная программа протестирована на нескольких наборах входных данных различной сложности. В ходе работы программы произведено сравнение методов решения и определены наиболее эффективные методы.

Программа разработана на языке программирования Python с использованием библиотек NumPy и Matplotlib. В работе использовались объектно-ориентированная база данных ZODB.

ABSTRACT

Graduation work was completed on 85 pages, contains 47 figures, 29 formulas, 13 tables, a list of sources used from 33 points.

Keywords: SYSTEM WITH ELEMENTS OF SELF-LEARNING, ARTIFICIAL INTELLIGENCE, CAUSAL LOGIC GAME, «MINESWEEPER», SCHEME

The final qualification work is devoted to the development of a system containing elements of self-learning, capable of finding solutions to the game «Minesweeper» using several methods and capable of evaluating the effectiveness of these methods.

The purpose of this work is to develop mathematical, algorithmic and software that allows to find the most effective solution to the game «Minesweeper».

In the course of the work, a review of logical problems, an overview of the main methods of artificial intelligence, a comparative review of human and machine learning processes is presented, as well as methods for solving the game «Minesweeper» are considered.

The result of the work is a program for solving the game "Minesweeper". This program has been tested on several sets of input data of varying complexity. In the course of the program, the comparison of solution methods was made, and the most effective methods were determined.

The program is developed in the Python programming language using the NumPy and Matplotlib libraries. The object-oriented database ZODB was used in the work.

СОДЕРЖАНИЕ

ПЕРЕЧЕНЬ СОКРАЩЕНИЙ И ОБОЗНАЧЕНИЙ

В настоящей магистерской диссертации применяют следующие сокращения и обозначения:

ИНС — искусственные нейронные сети

КЛИ — каузально-логическая игра

СЭС — система с элементами самообучения

ОДЗ — область допустимых значений

ВВЕДЕНИЕ

В последние десятилетия в научно-техническом обществе большое внимание уделяется развитию такой отрасли, как машинное обучение. Оно применяется во многих сферах деятельности людей, например, при поиске информации в интернете, при подборе музыки по предпочтениям, при доступе к банковским данным с применением биометрии, и это далеко не полный список, где используются методы машинного обучения. На данный момент наиболее популярными методами машинного обучения являются нейронные сети и экспертные системы. Нейронные сети позволяют решать большой объём задач, не связанных с логической обработкой данных, а экспертные системы позволяют решать задачи по строго заданному алгоритму. Однако, ни один из данных методов не позволяет найти решение логических задач без предварительной настройки, поскольку у данных методов отсутствуют алгоритмы самообучения. Разработка алгоритмов самообучения позволит решать широкий круг задач, связанных с логической обработкой данных.

Однако, для того чтобы определить основы для построения алгоритмов самообучения, необходимо пройти долгий путь. Для начала необходимо выбрать круг логических задач, разработать алгоритм их решения, затем разделить алгоритм на блоки, для каждого блока найти наиболее подходящий самообучающийся алгоритм (это может быть ИНС или что-то другое) и затем из самообучающихся блоков построить самообучающуюся систему для решения определённого круга задач. И это лишь примерный алгоритм разработки самообучающейся системы.

В данной работе рассматриваются методы построения системы с элементами самообучения для поиска эффективного решения определённого класса задач, связанных с логической обработкой данных. На основе данных методов разработаны алгоритмы и их программная реализация.

Цель работы — разработка комплекса алгоритмов с элементами самообучения и их программной реализации для независимого от человека поиска эффективного решения каузально-логических игр (на примере игры «Minesweeper»/«Сапёр»).

1 АНАЛИТИЧЕСКИЙ ОБЗОР

1.1 Обзор логических задач

Всего существует огромное множество логических задач и важно определить тот класс задач, на основе которых будет разрабатываться СЭС. Для определения класса логических задач необходимо задать критерии, по которым будет происходить их выбор. Это такие критерии, как:

- сложность;
- формализуемость;
- наличие однозначного решения (детерминированность).

Поясним необходимость данных критериев. Сложность логической задачи — это основополагающий критерий выбора. Хотя данный критерий можно считать субъективным, поскольку нет единого стандарта оценки сложности той или иной логической задачи, в данном случае под сложностью будем подразумевать:

- количество входных данных;
- уровень однотипности логических операций;
- «объём» логических операций, необходимых для решения задачи.

Если задача будет сложной, то и алгоритм для СЭС также должен быть сложным, а, поскольку в данной работе предполагается разработка СЭС для поиска решения не трудных логических задач, в приоритете будут задачи с низким и средним уровнями сложности.

От уровня формализуемости зависит сложность представления логической задачи в математических терминах, а, чем выше сложность представления логической задачи в математических терминах, тем выше вероятность запутаться в данных, установить неверные зависимости между ними, не учесть тот или иной элемент задачи. Таким образом, будем также ориентироваться на задачи с высоким и средним уровнями формализуемости.

Наличие однозначного решения также важно при выборе логической задачи для СЭС. Задачи, имеющие однозначные решения, проще решать, поскольку такие задачи могут решаться по «шаблону», которому СЭС необходимо научиться. Задачи, не имеющие однозначного решения, хоть и могут также решаться по шаблону, но такой метод не всегда может быть эффективным, поскольку в некоторых ситуациях необходим творческий подход к решению задачи, которому пока обучить машину не получилось. В данной работе будем отдавать предпочтение логическим задачам, которые имеют однозначное решение.

Головоломки — это распространённые и интересные логические задачи, решение которых зависит не от специальных знаний и творческих умений, а от сообразительности, а,

говоря на языке машины, зависит от алгоритма, определяющие порядок применения логических операций к тем или иным элементам задачи. Рассмотрим, какие бывают виды головоломок.

1.1.1 Виды головоломок

Ещё никто не придумал общей классификации для задач на логику, однако, исходя из их смысла, можно выделить четыре категории [1]:

- головоломки с предметами;
- механические головоломки;
- печатные головоломки;
- устные головоломки;
- компьютерные игры-головоломки.

Рассмотрим подробнее каждую из этих категорий.

1.1.1.1 Головоломки с предметами

Не обязательно покупать головоломку, порой бывает достаточно предметов, которые имеются в каждом доме. Одной из таких, которая обрела невероятную популярность, является головоломка со спичками. Её суть сводится к тому, что нужно переставить определённое количество спичек, чтобы получилась другая фигура. Ещё одна разновидность: переставить спички так, чтобы вышло верное равенство (речь идёт о цифрах). Наподобие этого есть головоломка с монетками.

1.1.1.2 Механические головоломки

Всемирно известный пример — кубик Рубика, автор которого также придумал «змейку». Также здесь можно вспомнить любимую игру детей — пятнашки. Все эти головоломки объединяет одно — это предметы, которые созданы для решения поставленных задач. Представленные игры найдутся далеко не в каждом доме, но есть и такая головоломка, в которую играли, пожалуй, все — это пазлы.

1.1.1.3 Печатные головоломки

К печатным головоломкам относятся такие головоломки, которые напечатаны на бумаге, а для их решения понадобится ручка. Примерами таких задач являются:

- сканворды;
- кроссворды;
- ребусы;
- японские кроссворды;
- судоку и другие.

1.1.1.4 Словесные головоломки

Словесные головоломки — это загадки, построенные на материале слов. В отличие от разного рода словесных игр, головоломки предназначены для индивидуального разгадывания. Словесные головоломки бывают разных типов. Рассмотрим некоторые словесные головоломки [2].

Анаграмма — это перестановка букв в слове, приводящая к новому слову, например: луг — гул, карп — парк, адрес — среда, рост — сорт — торс — трос, клоун — колун — кулон — уклон. Часто анаграммами называют сами слова, составленные из одинаковых букв.

Метаграммы — это слова, различающиеся одной буквой (звуком). Метаграммами часто называют также головоломки, основанные на изменении в слове одной буквы. В метаграммах принято загадывать не любые слова, а существительные в форме именительного падежа (допустимо использовать имена собственные).

Палиндром — такой текст, который читается от конца к началу так же, как от начала к концу (пробелы и знаки препинания не принимаются во внимание). В том же значении иногда используются термин «перевертыш». Один из известнейших примеров — А роза упала на лапу Азора.

Шарада один из наиболее популярных видов словесных головоломок. Шарада заключается в отгадывании слова, части которого могут быть самостоятельными словами. Эти части слов называются слогами. Понятие слога в шарадах не совпадает с понятием слога в фонетике. Слог в шараде лишь в частном случае может представлять собой фонетический слог, но может состоять и из нескольких фонетических слогов, а может вообще не содержать гласных.

Также в качестве словесных головоломок могут рассматриваться сканворды, кроссворды и ребусы, рассмотренные ранее.

1.1.1.5 Компьютерные игры-головоломки

К этой категории головоломок можно отнести наиболее известные игры, выпущенные как встроенные приложения в операционную систему Windows ещё в 90-е годы:

- сапёр;
- пинбол;
- пасьянс;
- червы;
- косынка;
- солитер;
- маджонг и другие.

1.1.2 Выбор класса логических задач

Итак, из большого количества логических игр выберем ту, для которой будем разрабатывать самообучающуюся систему.

Головоломка со спичками имеет достаточно узкий круг возможных задач, притом, данная задача может решаться с помощью полного перебора вариантов решения, поэтому данная задача не подходит.

Кубик Рубика является достаточно сложной задачей, хотя и достаточно популярной с имеющимся алгоритмом решения. Основная проблема при решении данной задачи — возможность визуализации процесса решения, поэтому данная задача также не подходит.

Сканворды, кроссворды и ребусы основаны на поиске информации по заданным данным, а также на игре слов, и при их решении практически не используются логические элементы, а «Судоку» уже подходит по критериям, поскольку данная логическая задача имеет средний уровень сложности, хорошо формализуема, а также при некоторых ограничениях имеет однозначное решение.

Карточные игры, наподобие игр «Пасьянс», «Червы», «Косынка», «Солитер», а также «Маджонг» не всегда могут иметь однозначное решение, «Пинбол» достаточно трудно формализуемая задача, а «Сапёр» подходит по рассматриваемым критериям.

Исходя из того, что весь класс головоломок, как логических задач для построения системы с элементами самообучения, не соответствует представленным ранее требованиям, необходимо по-другому определить класс подходящих для рассмотрения логических задач. Будем опираться на представленные ранее критерии выбора задач, а именно:

- задачи со средним или низким уровнями сложности;
- задачи со средним или высоким уровнями формализуемости;
- задачи, имеющие однозначное (детерминированное) решение.

Главным критерием в данном случае является детерминированность решения. Также, при рассмотрении логических задач, чётко прослеживается причинно-следственная связь

между действиями и поставленной целью, что можно охарактеризовать с помощью термина «каузальный». Таким образом, определим свой класс рассматриваемых задач — детерминированные каузально-логические игры и дадим ему определение. Детерминированные каузально-логические игры — это класс логических задач или игр среднего или низкого уровня сложности, среднего или высокого уровня формализуемости и имеющие однозначное (детерминированное) решение.

К данному классу можно отнести большое количество задач. Исходя из рассмотренных логических задач, это такие задачи как «Судоку» и «Сапёр». Однако, важно понимать, что далеко не все задачи могут иметь однозначное решение. Если рассматривать «Судоку», то, как правило, каждая задача имеет детерминированное решение, поскольку, в противном случае, задача не имеет смысла, а если рассматривать игру «Сапёр», то в ней могут присутствовать «элементы случайностей», что не соответствует критерию детерминированности решения. Однако, если ввести ограничение на наличие однозначного решения задачи (если это ограничение, конечно, возможно ввести), то рассматриваемую логическую задачу возможно отнести к классу каузально-логических.

1.2 Обзор методов искусственного интеллекта

Рассмотрим теперь наиболее распространённые методы искусственного интеллекта, которые используются чаще всего.

1.2.1 Искусственная нейронная сеть

Искусственная нейронная сеть — это математическая модель и её программное или аппаратное воплощение, которое построено по принципу организации и функционирования биологических нейронных сетей — сетей нервных клеток живого организма. Составной частью нейронной сети являются нейроны. Структура нейрона представлена на рисунке 1.

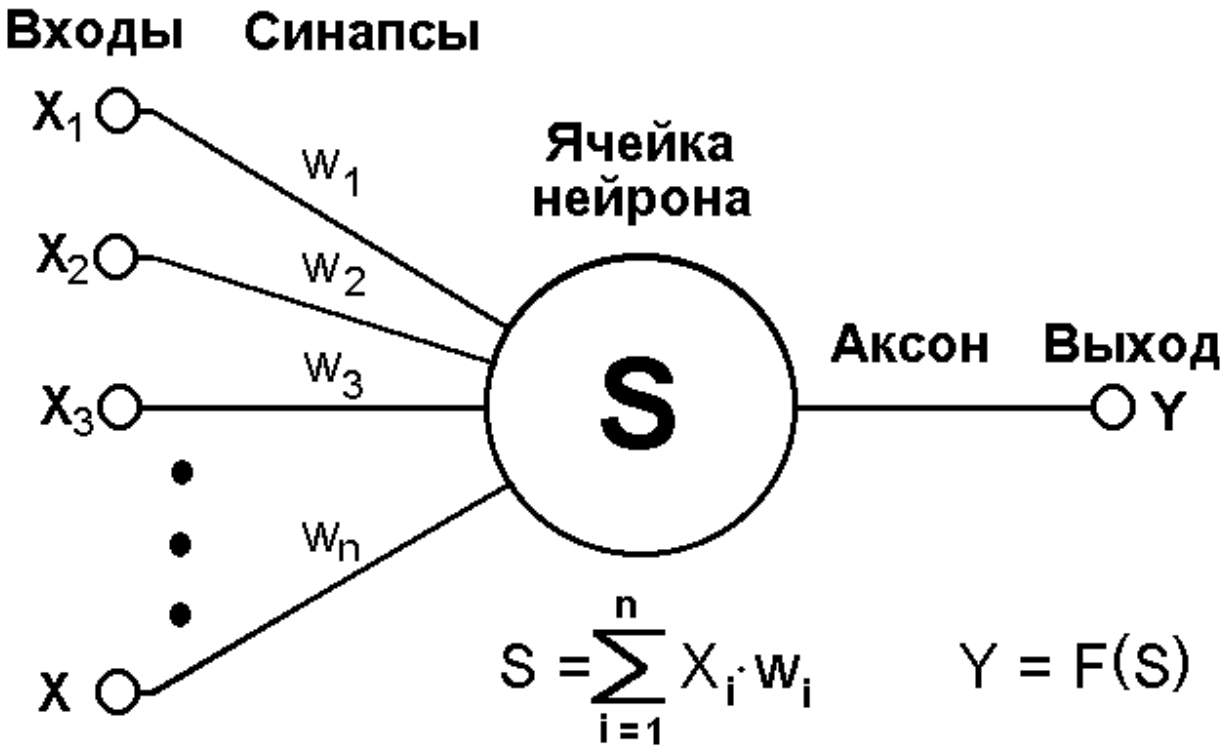

Рисунок 1 — Структура искусственного нейрона

В состав нейрона входят умножители (синапсы), сумматор и нелинейный преобразователь [3]. Синапсы осуществляют связь между нейронами и умножают входной сигнал на число, характеризующее силу связи, — вес синапса. Сумматор выполняет сложение сигналов, поступающих по синаптическим связям от других нейронов и внешних входных сигналов. Нелинейный преобразователь реализует нелинейную функцию одного аргумента — выхода сумматора. Эта функция называется «функция активации» или «передаточная функция» нейрона. Нейрон в целом реализует скалярную функцию векторного аргумента. Математическая модель нейрона описывается соотношениями (формула 1):

$$s = \sum_{i=1}^{n} w_i x_i + b, \tag{1}$$

где w_i — вес синапса ($i = 1, \dots, n$),

b — значение смещения,

s — результат суммирования,

x_i — компонента входного вектора (входной сигнал),

y — выходной сигнал нейрона,

f — нелинейное преобразование (функция активации или передаточная функция).

Таким образом, нейрон полностью описывается своими весами w_i и передаточной функцией $f(s)$. Получив набор чисел (вектор) x_i в качестве входов, нейрон выдаёт некоторое число y на выходе.

Описанный вычислительный элемент можно считать упрощённой математической моделью биологических нейронов — клеток, из которых состоит нервная система человека и животных.

Рассмотрим некоторые задачи, которые можно решить с помощью ИНС:

- классификация образов;
- кластеризация/категоризация;
- аппроксимация функций;
- прогноз;
- оптимизация и др.

1.2.2 Экспертная система

Экспертная система — это программа для компьютера, которая оперирует со знаниями в определенной предметной области с целью выработки рекомендаций или решения проблем [4]. Структура экспертной системы представлена на рисунке 2.

Экспертная система может полностью взять на себя функции, выполнение которых обычно требует привлечения опыта человека-специалиста, или играть роль ассистента для человека, принимающего решение. Другими словами, система (техническая или социальная), требующая принятия решения, может получить его непосредственно от программы или через промежуточное звено — человека, который общается с программой. Тот, кто принимает решение, может быть экспертом, и в этом случае программа может повысить эффективность его работы. Альтернативный вариант — человек, работающий в сотрудничестве с такой программой, может добиться с её помощью результатов более высокого качества. Таким образом, правильное распределение функций между человеком и машиной является одним из ключевых условий высокой эффективности внедрения экспертных систем.

Технология экспертных систем является одним из направлений области исследования, которая получила наименование искусственного интеллекта. Исследования в этой области сконцентрированы на разработке и внедрении компьютерных программ, способных эмулировать те области деятельности человека, которые требуют мышления, определенного мастерства и накопленного опыта. К ним относятся задачи принятия решений, распознавания образов и понимания человеческого языка. Эта технология уже успешно применяется в

некоторых областях техники и жизни общества — органической химии, поиске полезных ископаемых, медицинской диагностике.

Рисунок 2 — Структура экспертной системы

Экспертная система отличается от прочих прикладных программ наличием следующих признаков:

- моделирует не столько физическую (или иную) природу определенной проблемной области, сколько механизм мышления человека применительно к решению задач в этой проблемной области. Это существенно отличает экспертные системы от систем математического моделирования или компьютерной анимации. Нельзя, конечно, сказать, что программа полностью воспроизводит психологическую модель специалиста в этой предметной области (эксперта), но важно, что основное внимание уделяется воспроизведению компьютерными средствами методики решения проблем, которая применяется экспертом, — то есть выполнению некоторой части задач так же (или даже лучше), как это делает эксперт;
- система, помимо выполнения вычислительных операций, формирует определенные соображения и выводы, основываясь на тех знаниях, которыми она располагает. Знания в системе представлены, как правило, на некотором специальном языке и хранятся отдельно от собственно программного кода, который и формирует выводы и соображения. Этот компонент программы принято называть базой знаний;

– при решении задач основными являются эвристические и приближенные методы, которые, в отличие от алгоритмических, не всегда гарантируют успех. Эвристика, по существу, является правилом влияния, которое в машинном виде представляет некоторое знание, приобретенное человеком по мере накопления практического опыта решения аналогичных проблем. Такие методы являются приблизительными в том смысле, что, во-первых, они не требуют исчерпывающей исходной информации, и, во-вторых, существует определенная степень уверенности (или неуверенности) в том, что предлагаемое решение является верным.

Экспертные системы отличаются и от других видов программ из области искусственного интеллекта. Так, например, экспертные системы имеют дело с предметами реального мира, операции с которыми обычно требуют наличия значительного опыта, накопленного человеком. Экспертные системы имеют ярко выраженную практическую направленность в научной или коммерческой области.

Одной из основных характеристик экспертной системы является её производительность, то есть скорость получения результата и его достоверность (надежность). Исследовательские программы искусственного интеллекта могут и не быть очень быстрыми, можно примириться и с существованием в них отказов в отдельных ситуациях, поскольку, в конце концов, — это инструмент исследования, а не программный продукт. А вот экспертная система должна за приемлемое время найти решение, которое было бы не хуже, чем то, которое может предложить специалист в этой предметной области.

Экспертная система должна обладать способностью объяснить, почему предложено именно такое решение, и доказать его обоснованность. Пользователь должен получить всю информацию, необходимую ему для того, чтобы быть уверенным, что решение принято «не с потолка». В отличие от этого, исследовательские программы «общаются» только со своим создателем, который и так (скорее всего) знает, на чём основывается её результат. Экспертная система проектируется в расчёте на взаимодействие с разными пользователями, для которых её работа должна быть, по возможности, прозрачной.

Зачастую термин «система, основанная на знаниях», используется в качестве синонима термина экспертная система, хотя, строго говоря, экспертная система — это более широкое понятие. Система, основанная на знаниях, — это любая система, процесс работы которой основан на применении правил отношений к символическому представлению знаний, а не на использовании алгоритмических или статистических методов. Таким образом, программа, способная рассуждать о погоде, будет системой, основанной на знаниях, даже в том случае, если она не способна выполнить метеорологическую экспертизу. А вот чтобы иметь право

называться метеорологической экспертной системой, программа должна быть способна давать прогноз погоды (другой вопрос — насколько он будет достоверен).

Суммируя все сказанное, можно отметить, что экспертная система содержит знания в определенной предметной области, накопленные в результате практической деятельности человека (или человечества), и использует их для решения проблем, специфичных для этой области. Этим экспертные системы отличаются от прочих, «традиционных» систем, в которых предпочтение отдается более общим и менее связанным с предметной областью теоретическим методам, чаще всего математическим. Процесс создания экспертной системы часто называют инженерией знаний, и он рассматривается в качестве «применения методов искусственного интеллекта».

1.3 Сравнение процессов обучения человека и машины

Прежде чем перейти к построению самообучающейся системы или системы с элементами самообучения, необходимо рассмотреть, как происходит процесс обучения у человека, поскольку большая часть алгоритмов машинного обучения основана на тех принципах получения и обработки знаний, которые происходят у человека. Таким образом, появится возможность «позаимствовать» у человека модель обучения, которую получится реализовать в дальнейшем. Рассмотрение процессов обучения у машины позволит определить те достоинства и недостатки, которые также необходимо будет учесть при разработке системы с элементами самообучения.

1.3.1 Процесс обучения у человека

Рассмотрим сначала несколько формулировок понятия «обучение».

Обучение — педагогический процесс, в результате которого учащиеся под руководством учителя овладевают знаниями, умениями и навыками, общими и специальными [5].

Обучение — это сознательная целенаправленная деятельность педагогов и учащихся [6].

Понятие «обучение» характеризует организованный процесс, порождаемый взаимодействием двух деятельностей, — преподавания и учения [7].

Обучение — основной путь получения образования, целенаправленный, планомерно и систематически осуществляемый процесс овладения знаниями, умениями и навыками под руководством опытных лиц — педагогов, мастеров, наставников и т. д [8].

Обобщая разные формулировки, процесс обучения можно наглядно представить в виде следующей схемы (рисунок 3).

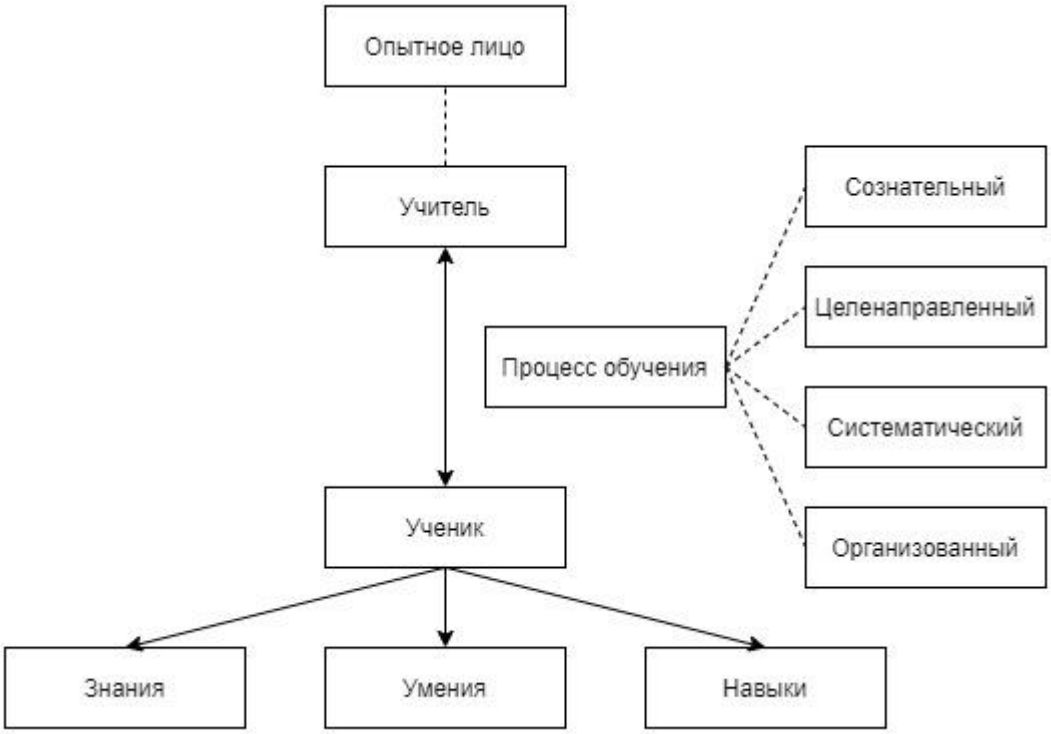

Рисунок 3 — Общая схема процесса обучения

Таким образом, во время обучения при помощи учителя, а именно, лица, обладающего достаточными компетенциями, у обучающегося формируются знания, умения и навыки. Важным будет отметить, что является знаниями, полученными в процессе обучения. Знание — это осведомленность или понимание кого и чего угодно, которое можно логически или фактически обосновать и эмпирически или практически проверить [9]. Главной фразой в данном определении будет «понимание и возможность логически обосновать». Таким образом, понимание позволяет человеку связать уже имеющиеся у него знания с новыми знаниями, позволяет заново вывести данное знание логически в случае утраты данного знания, например, в случае забывания.

Также необходимо обратить внимание на то, что является умением. Умение — это возможность эффективно выполнять действие (деятельность) в соответствии с целями и условиями, в которых приходится действовать [10]. В данном случае ключевой является фраза «эффективно выполнять». Получается, что важно не только наличие знаний в той или сфере, но и умение эффективно применять их для решения поставленных задач. И, наконец, навык — это действие, сформированное путём повторения, характеризующееся высокой степенью

освоения и отсутствием поэлементной сознательной регуляции и контроля [11]. Таким образом, можно сказать, что навык — это некоторое действие, доведённое до автоматизма, которое может происходить без сознательной регуляции.

В качестве примера процесса обучения можно привести обучение чистке зубов детей. Изначально ребёнок, как правило, под руководством родителя получает знания, в том числе для чего необходим данный процесс, как правильно осуществлять данный процесс, как это делать эффективно и т. д. Затем он пробует совершить осознанные действия с зубной щёткой, то есть получает умения. Через некоторое время ребёнок уже может чистить зубы менее осознанно, но быстрее и эффективнее относительно своего первого опыта с зубной щёткой. Таким образом, ребёнок получает навыки.

1.3.2 Процесс обучения у машины

Определим сначала термин «обучение» в контексте теории искусственного интеллекта. Обучение — это процесс, при котором адаптивная система (например, ИНС) совершенствует свои параметры под действием внешнего окружения [12]. Также под обучением может пониматься обучаемость адаптивных систем в целом.

Обучить ИНС — значит, сообщить ей то, что от неё хотят получить. Этот процесс очень похож на обучение ребёнка алфавиту. Показав ребёнку изображение буквы «А», его спрашивают: «Какая это буква?». Если ответ неверен, ребёнку сообщается тот ответ, который от него хотят получить: «Это буква А». Ребёнок запоминает этот пример вместе с верным ответом, то есть в его памяти происходят некоторые изменения в нужном направлении. Процесс предъявления букв будет повторяться снова и снова до тех пор, пока все буквы не будут твёрдо запомнены. Такой процесс называют «обучение с учителем» (рисунок 4).

При обучении ИНС происходят аналогичные процессы. Имеется база данных, содержащая примеры (набор рукописных изображений букв). Предъявляя изображение буквы «А» на вход ИНС, на выходе получается ответ, который не обязательно является верным. Известен и верный (желаемый) ответ — в данном случае требуется, чтобы на выходе с меткой «А» уровень сигнала был максимален. Обычно в качестве желаемого выхода в задаче классификации берут набор (1, 0, 0,…), где 1 стоит на выходе с меткой «А», а 0 — на всех остальных выходах. Вычисляя разность между желаемым ответом и реальным ответом ИНС, получается (для букв русского алфавита) 33 числа — вектор ошибки. Алгоритм обучения — это набор формул, который позволяет по вектору ошибки вычислять требуемые поправки для весов сети. Одну и ту же букву (а также различные изображения одной и той же буквы) можно

предъявлять сети много раз. В этом смысле обучение скорее напоминает повторение упражнений в спорте — тренировку.

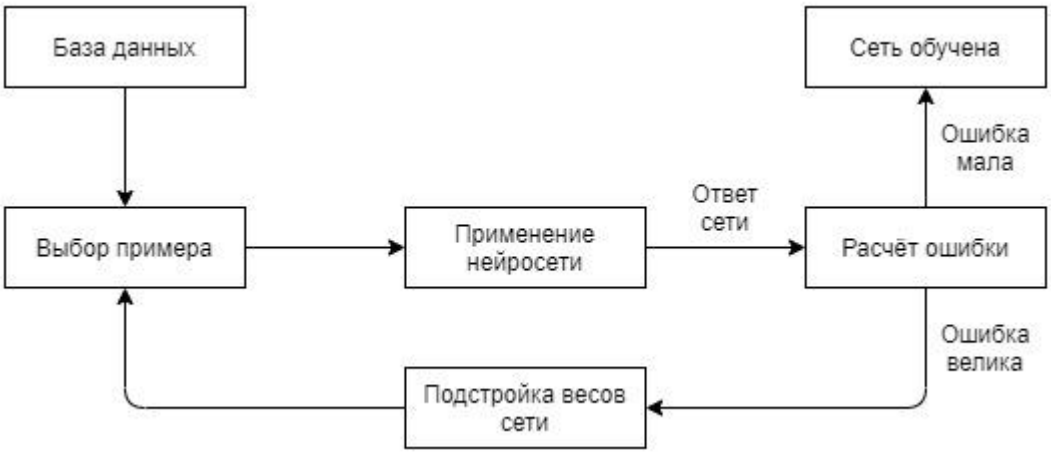

Рисунок 4 — Иллюстрация процесса обучения ИНС

Оказывается, что после многократного предъявления примеров веса сети стабилизируются, причём сеть даёт правильные ответы на все (или почти все) примеры из базы данных. В таком случае говорят, что «сеть обучена» или «сеть натренирована». В программных реализациях можно видеть, что в процессе обучения функция ошибки (например, сумма квадратов ошибок по всем выходам) постепенно уменьшается. Когда функция ошибки достигает нуля или приемлемо малого уровня, тренировку останавливают, а полученную сеть считают натренированной и готовой к применению на новых данных.

Важно отметить, что вся информация, которую сеть имеет о задаче, содержится в наборе примеров. Поэтому качество обучения сети напрямую зависит от количества примеров в обучающей выборке, а также от того, насколько полно эти примеры описывают данную задачу. Так, например, бессмысленно использовать сеть для прогнозирования финансового кризиса, если в обучающей выборке кризисов не представлено. Считается, что для полноценной тренировки требуется хотя бы несколько десятков (а лучше сотен) примеров.

Процесс обучения экспертной системы значительно отличается от процесса обучения ИНС. Главное отличие заключается в том, что экспертная система не обучается сама, а её «обучает» специальный человек — проектировщик экспертной системы. Процесс обучения или процесс приобретения знаний экспертной системой представляет из себя передачу потенциального опыта решения проблемы от некоторого источника знаний и преобразование его в вид, который позволяет использовать эти знания в программе.

Передача знаний выполняется в процессе достаточно длительных и пространных собеседований между специалистом по проектированию экспертной системы (инженером по знаниям) и экспертом в определенной предметной области, способным достаточно чётко сформулировать имеющийся у него опыт (рисунок 5). По существующим оценкам, таким методом можно сформировать от двух до пяти «элементов знания» (например, правил влияния) в день. Конечно, это очень низкая скорость, а потому многие исследователи рассматривают функцию приобретения знаний в качестве одного из главных «узких мест» технологии экспертных систем.

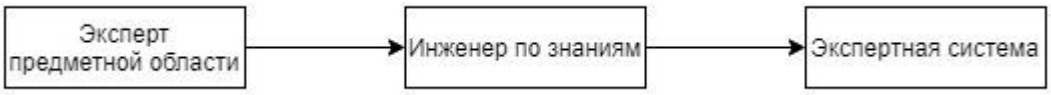

Рисунок 5 — Процесс передачи знаний экспертной системе

Таким образом, передача знаний представляет из себя сначала процесс приобретения знаний инженером по знаниям от эксперта предметной области, а затем процесс «передачи» знаний экспертной системе.

Не маловажным будет также отметить, как представляются знания в экспертной системе. Теория представления знаний — это отдельная область исследований, тесно связанная с философией формализма и когнитивной психологией. Предмет исследования в этой области — методы ассоциативного хранения информации, подобные тем, которые существуют в мозгу человека. При этом основное внимание, естественно, уделяется логической, а не биологической стороне процесса, опуская подробности физических преобразований.

Основная часть представления знаний состоит в том, что представление должно каким-то образом «стандартизировать» семантическое разнообразие человеческого языка. Вот несколько предложений.

«Сэм — отец Билла». «Сэм — Биллов отец». «Биллов отец — Сэм».

«Отцом Билла является Сэм».

Все эти фразы выражают одну и ту же мысль (семантически идентичны). При машинном представлении этой мысли (знания) ищется более простой метод сопоставления формы и содержания, чем в обычном человеческом языке, то есть стараются добиться того, чтобы выражения с одинаковым (или похожим) содержанием были одинаковыми и по форме. Например, все приведенные выше фразы могут быть сведены к выражению в такой форме:

отец (сэм, билл).

В общем, вопрос представления знания был и скорее всего останется вопросом противоречивым. Философы и психологи зачастую бывают шокированы бесцеремонностью специалистов по искусственному интеллекту, которые активно разговариваю о человеческом знании на жаргоне, представляющем жуткую смесь терминологии, взятой из логики, логистики, философии, психологии и информатики. С другой стороны, компьютерный формализм оказался новаторским средством постановки, а иногда и поиска ответов на трудные вопросы, над которыми столетиями бились метафизики.

1.3.3 Сравнительный анализ

Рассмотрим сначала ИНС. Главными характеристиками ИНС являются:

- ИНС построена по принципу организации и функционирования биологических нейронных сетей;
- отсутствие возможности «объяснить», почему ИНС обучилась именно таким образом;
- «постепенное» обучение в том смысле, что необходимо предъявление некоторого количества элементов обучающей выборки, чтобы ИНС научилась различать элементы разных классов.

Исходя из данных характеристик, ИНС можно сравнить с теориями научения Д. Б. Уотсона, И. П. Павлова и Б. Ф. Скиннера. Джон Уотсон — американский психолог и основатель бихевиоризма. Уотсон уделял огромное внимание классическому научению, при котором организм ассоциирует разные стимулы (звук колокольчика — условный раздражитель, а слюноотделение у собаки в ответ на звук этого колокольчика — условный рефлекс). Такой вид научения ориентирован на непроизвольные, автоматические действия.

Организм как человека, так и животного приспосабливается к своему окружению посредством врождённого и приобретённого набора актов, то есть поведения. Всю психическую деятельность Уотсон трактовал как поведение. Он рассматривал его как совокупность реакций организма на стимулы, то есть поведение по принципу «стимул-реакция» (S → R) [13]. Дж. Уотсон считал, что, подобрав верный стимул, можно формировать нужные навыки и качества в человеке или животном.

Иван Павлов — русский и советский учёный, создатель науки о высшей нервной деятельности. Павлов впервые описал теорию классического обусловливания. Классическое обусловливание является одной из форм научения, которая является общей для всех животных, включая человека, и появляется даже у одноклеточных организмов. Павлов показал, что рефлекторные (инстинктивные) реакции можно вызвать в организме действием

нейтральных стимулов, которые сами по себе не вызывают эти реакции. Таким образом, организм научится отвечать на возникшие нейтральные раздражители определенной рефлекторной реакцией. Подача еды — безусловного стимула — собаке приводит к безусловному рефлексу — автоматическому слюноотделению. Если подача пищи ассоциируется со звуком звонка — условный стимул — после нескольких повторений происходит слюноотделение после самого звука — возникает условный рефлекс. Если после звука звонка не последует безусловный раздражитель, после нескольких таких воздействий произойдет постепенное угасание условного рефлекса. Такое научение является типичным бессознательным [14]. Например, до обусловливания математика не вызывает никакой реакции если учитель недружелюбный, с резким голосом — это вызывает у учащихся неприязнь и страх, то есть безусловный стимул не вызывает никаких сильных эмоций, за исключением негативных и научения не происходит. Классическое обусловливание это когда нейтральный стимул (математика) совмещается с безусловным (недружелюбный учитель с резким голосом), порождается страх и неприязнь (к математике), то есть безусловный стимул.

Фредерик Скиннер — американский психолог, изобретатель и писатель. Скиннер ввёл понятие оперантного обусловливания. Краткое объяснение оперантного обусловливания можно представить следующим образом. Чтобы, например, научить собаку новым трюкам, необходимо её вознаградить, а чтобы отучить от чего-либо, то необходимо наказать её. Исследователи обнаружили, что когда после конкретного поведения следует награда, увеличивается вероятность последующего возникновения такого поведения; если оно сопровождается неприятными последствиями, то наказание будет повторяться реже. Скиннер дал этому принципу дал определение «укрепляющий стимул» [15]. Этот тип обучения можно продемонстрировать следующим образом (рисунок 6).

Рисунок 6 — Общая схема оперантного обусловливания

Также важно выделить типы оперантного обусловливания (таблица 1).

Исходя из представленных теорий научения можно сделать следующие выводы:

– Уотсон рассматривал поведение как совокупность реакций организма на стимулы, то есть поведение по принципу «стимул-реакция» (S → R). Аналогичным образом можно

описать «поведение» ИНС, где стимулом является элемент выборки, а реакцией — определённая для данного элемента метка класса;

Таблица 1 — Типы оперантного обусловливания

Вероятность наблюдаемого поведения	Внешнее событие после наблюдаемого поведения	
	Последствие присутствует (оно предоставлено)	Последствие отсутствует (оно устранено)
↑	Положительное подкрепление (вознаграждение)	Отрицательное подкрепление (облегчение, отсутствие наказания)
↓	Положительное ослабление (наказание)	Отрицательное ослабление (отсутствие вознаграждения)

– ранее было сказано, что классическое обусловливание — это типичное бессознательное, то есть это такое научение, которое не входит в сферу сознаний субъекта (человека). Исходя из того, что ИНС не может «объяснить», почему она обучилась именно таким образом, можно провести аналогию между обучением ИНС и бессознательным, а, точнее, говоря более «мягко», неосознаваемым;

– также можно провести аналогию между типами оперантного обусловливания (таблица 1) и матрицей ошибок (confusion matrix), получаемую в ходе процесса обучения нейронной сети (таблица 2) [16]. Можно установить следующее соответствие:

1) True positives — Положительное подкрепление;
2) False positives — Отрицательное ослабление (отсутствие наказания);
3) False negatives — Положительное ослабление (наказание);
4) True negatives — Отрицательное подкрепление (отсутствие наказания).

Таблица 2 — Матрица ошибок (confusion matrix)

True class / Hypothesized class	P	N
Y	True Positives	False Positives
N	False Negatives	True Negatives

Подводя итог, прослеживается сходство между ИНС и теорией научения в том смысле, что ИНС является не только математической моделью биологической нейронной сети, а также в некоторой степени является моделью теории обусловливания.

В отличие от ИНС, экспертная система может и, более того, обязана объяснять, как и почему она пришла к полученному результату. Все выводы, которые делает экспертная система, основываются на законах логики, что позволяет ей давать результаты с высокой точностью и обоснованностью. Таким образом, обучение экспертной системы можно сравнить с обучением человека (рисунок 2 и рисунок 3). В процессе обучения человек обрабатывает получаемую от учителя информацию, преобразуя её в знания. Для экспертной системы аналогичную задачу также выполняет человек, таким образом, экспертная система получает на вход при помощи модуля приобретения знаний уже обработанные данные некоторой предметной области, которые сохраняются в базе данных и базе знаний. Механизм логических выводов экспертной системы позволяет обрабатывать и анализировать информацию, руководствуясь правилами логики. Аналогичным образом может рассуждать и человек с тем лишь замечанием, что человек может допускать когнитивные ошибки при рассуждениях, а экспертная система нет (если, конечно, исходные данные для экспертной системы заданы корректно). Умения и навыки, которые приобретает человек в процессе обучения уже не характерны для экспертной системы, поскольку приобретение умение и навыков связано в основном с практической деятельностью, в то время как большинство экспертных систем рассчитано на теоретическую поддержку человека. Таким образом, хотя можно провести аналогию и выявить сходство между процессом обучения экспертной системы и её функционированием и процессом обучения человека, тем не менее, можно также выделить и серьёзные отличия между данными процессами.

Попробуем дать теперь ответ на главный вопрос: в чём же заключается сходство и различие между процессом обучения человека и машины? Обучение человека по критерию осознаваемости можно разделить на следующие виды:

- неосознаваемое (то есть такое обучение, при котором у человека отсутствует понимание о факте и процессе обучения);
- осознаваемое (то есть такое обучение, при котором у человека присутствует понимание о факте и процессе обучения);
- частично осознаваемое (то есть такое обучение, при котором у человека присутствует частичное понимание о факте и процессе обучения).

При неосознаваемом процесс обучения человека схож с процессом обучения ИНС. При осознаваемом процесс обучения человека схож с процессом обучения человеком экспертной системы. При частично осознаваемом процесс обучения человека схож с совмещённым

процессом обучения ИНС и экспертной системы. Различие заключается в том, что человек способен к самообучению, а машина пока ещё нет, однако, при помощи «совмещения» работы ИНС и экспертной системы с дополнительной её модификацией возможно частично или полностью достичь эффекта самообучения у машины. Таким образом, становится важным разрабатывать методы и алгоритмы для построения и модификации таких систем.

1.3.4 Общая модель системы с элементами самообучения

Прежде чем перейти к описанию системы с элементами самообучения, рассмотрим общую схему поиска решения поставленной задачи (рисунок 7). Как можно увидеть, данная схема состоит из пяти блоков. Входными данными являются правила игры или условие рассматриваемой задачи. При обработке правил формируются содержательная и математическая постановки задачи. На их основе осуществляется поиск методов решения. Затем производится поиск способов эффективного применения разработанных методов. И только после этого разрабатываются программные средства.

Рисунок 7 — Общая схема решения задачи

Для того, чтобы система была полностью самообучающейся, необходимо, чтобы каждый из описанных шагов, связывающих блоки схемы, автоматически обрабатывались (преобразовывались) системой с минимальным участием человека. Поскольку разработка данной системы представляет достаточно трудную и длительную по времени задачу, в данной работе рассмотрим и разработаем не самообучающуюся систему, а систему с элементами самообучения, в которой лишь часть шагов, представленных на схеме, будет автоматически обрабатываться, а именно, это шаг, осуществляющий переход от разработанных методов решения задачи к их эффективному применению.

1.4 Методы решения игры «Сапёр»

Рассмотрим теперь существующие алгоритмы для поиска решения игры «Сапёр».

1.4.1 Гибридные модели анализа ситуаций

Наиболее частым подходом к построению современных интеллектуальных систем является использование гибридных моделей анализа ситуаций [17].

Рассмотрим обучаемую модель игры в составе следующих блоков:

1) Создание поля. Создание игрового поля состоит из двух этапов:
   - с помощью функции генерации случайных чисел размещается заданное число единиц - «мин» по полю заданного размера в матрице A;
   - для каждой клетки матрицы A вычисляется сумма окружающих ее «мин», значение записывается в соответствующую ячейку матрицы B того же размера.

При решении задачи «Сапёра» пользователь или программа обращаются к матрице B во время открытия неизвестных ячеек; с помощью матрицы A проверяется количество оставшихся «мин» и условия проигрыша и победы. В матрице C того же размера ведется учет вероятностей нахождения «мин» в той или иной клетке. Матрица D содержит описание текущей видимой ситуации. В начальный момент все $d_{i,j} = 10$, т. е. не опознаны. Для клеток с «миной» будем использовать маркер $d_{i,j} = 9$, открытые клетки содержат маркеры от $d_{i,j} = 0$ («мин» рядом нет) до $d_{i,j} = 8$, (вокруг только «мины»).

2) Модуль обработки жесткой логики содержит следующие правила:
   - если количество «мин» вокруг данной клетки соответствует ее числу, можно открыть все остальные ячейки вокруг нее, т. е. вероятность нахождения «мины» устанавливается равной 0;
   - если количество неизвестных «мин» вокруг данной клетки равно числу свободных клеток вокруг нее, то можно поставить там «мины», т. е. вероятность нахождения «мины» устанавливается равной 1;
3) Модуль принятия решения содержит следующие правила:
   - при неоднозначной расстановке «мин» следует выбрать наиболее вероятные точки путем составления матрицы вероятных положений с учетом обучения и расположения открытых клеток;
   - при равнозначном выборе в режиме обучения запросить помощь пользователя. Описать решение пользователя в виде модели размера $K(n)_{5x5}$ извлеченной из матрицы D. На основании его решений изменить параметры

вероятности соответствующей клетки и проверить качество принятого решения: «удача\ошибка». Если решение было удачным, то вероятность отсутствия «мины» для клеток, соответствующих области типа $K(n)_{5x5}$ увеличить, иначе следует ее уменьшить. В начальный момент считаем, что $P(K(n)_{5x5} - \text{«мины»}) = 1$, т. е. для всех образцов «мины» есть;

- в режиме игры при равнозначном выборе полагаться на выбор. На основании получившегося результата изменить параметры вероятности;

4) Модуль изменения параметров вероятности. Работа с матрицей вероятностей С включает в себя следующие правила:

- свободные закрытые клетки имеют собственную вероятность нулевого уровня, являющуюся суммой вероятностей нахождения «мины» около открытых ячеек. Соответствующие слагаемые вычисляются как отношение количества недостающих «мин» к количеству свободных закрытых клеток вокруг открытых ячеек, расположенных вокруг текущей клетки;
- изменение вероятности при поощрении решения вычисляется следующим образом: $P_{t+1} = P_t + \Delta$;
- изменение вероятности при наказании решения вычисляется как:

$P_{t+1} = P_t - \Delta$.

Таким образом, картина вероятности для каждой клетки, неоткрытой в текущий момент, обуславливается опытом удачных или неудачных решений в схожей ситуации и, с увеличением объема накопленных результатов, изменяется на соответствующую величину.

Результаты применения данного алгоритма представлены в таблице 3 [18].

Таблица 3 — Результаты применения алгоритма

	Без анализа вероятностей			С анализом вероятностей		
число мин	игр	побед	%	игр	побед	%
$q = 10$	50	38	76,0	49	34	69,4
$q = 15$	50	6	12,0	49	23	46,9
$q = 20$	50	0,5	1,0	72	6	8,3

1.4.2 Сторонний алгоритм решения игры «Сапёр»

Рассмотрим ещё один алгоритм поиска решения игры «Сапёр» [19].

В начале игры психологически проще обезвреживать единицы, а, точнее, те клетки, в которых стоит единица и есть смежная к ней только одна закрытая (рисунок 8). Вокруг клетки C3 может находиться только одна заминированная клетка, а, так как и закрытая смежная клетка только одна (B2), то она и содержит эту мину.

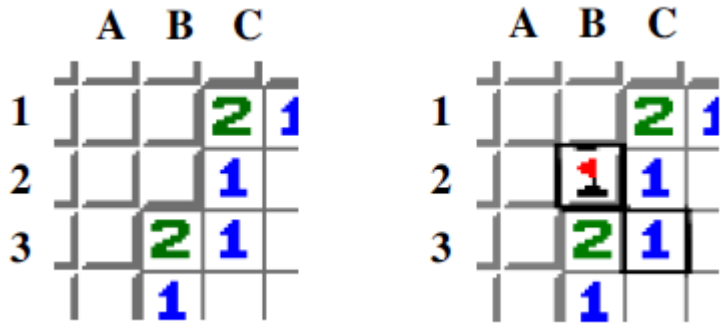

Рисунок 8 — Пример 1

Возможно расширить эти рассуждения на случай большего числа мин: если количество закрытых смежных клеток равно числу, стоящему в данной клетке, то все эти закрытые клетки содержат мины (рисунок 9).

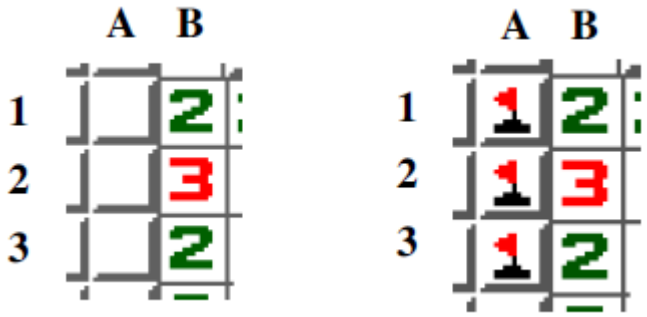

Рисунок 9 — Пример 2

Теперь будем учитывать уже известные заминированные поля. Понятно, что если вокруг клетки обнаружено столько мин, сколько требуется, то все остальные смежные клетки безопасны (рисунок 10). Или если суммарное количество смежных закрытых и заминированных клеток дают в точности необходимое число мин, то все такие закрытые

клетки заминированы (рисунок 11). Также учтём тот факт, что закрытыми являются все те клетки, содержимое которых неизвестно, что даёт возможность не рассматривать клетки, вокруг которых уже всё обезврежено.

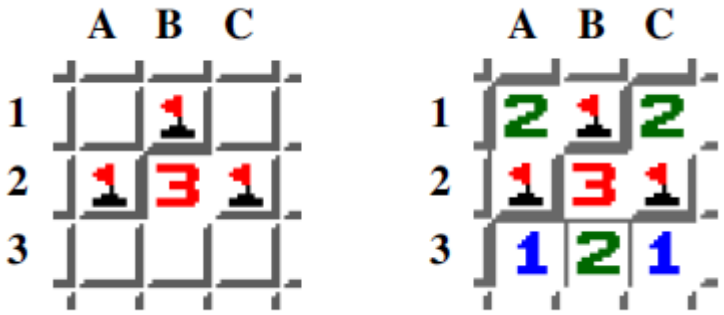

Рисунок 10 — Пример 3

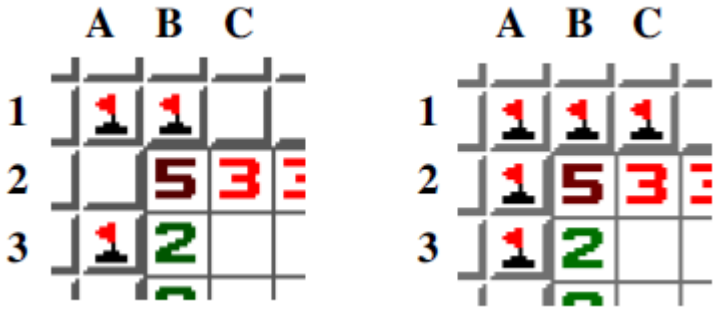

Рисунок 11 — Пример 4

Подобные пересчёты числа расставленных мин и закрытых клеток, несмотря на простоту, довольно продуктивны: алгоритм приведёт к победе на младших уровнях сложности игры. «Профессионал» же из-за высокого относительного количества мин на клетку порождает интересные случаи (рисунок 12).

Понятно, что построенный алгоритм ничего не сможет открыть на таких полях. Тем не менее, видно, что некоторые клетки совершенно точно или заминированы, или безопасны. На рисунке 12 а мина для B1 может находиться в A1 или в A2 и нигде больше, а эти клетки влияют на B2: если в A1-A2 точно находится мина, то в A3 точно безопасно (рисунок 13). Похожим образом открываются клетки на рисунке 12 б; для C2 возможны только три варианта расстановки мин (B1-C1, C1-D1 и B1-D1), и только один вариант правильный, так как в остальных двух превышается число расставленных мин для клеток B2 и C2 (рисунок 14). И,

наконец, на рисунке 12 в клетка B1 заминирована: единица из A3 позволяет ставить в B2-B3 только одну мину, а двойке на A2 нужна ещё одна, место для которой остаётся лишь в B1.

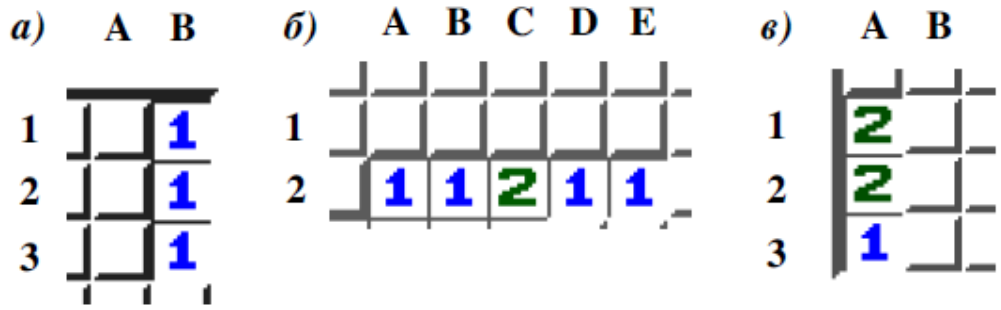

Рисунок 12 — Пример 5

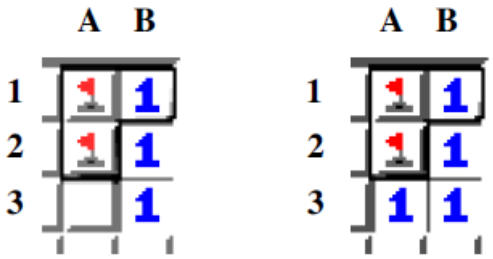

Рисунок 13 — Пример 6

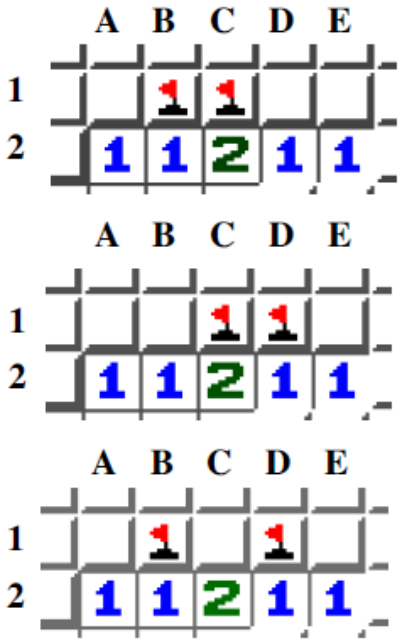

Рисунок 14 — Пример 7

Такие ситуации довольно часты, они формируют шаблоны расстановки мин на поле. Такими шаблонами почти бессознательно мыслят профессиональные игроки в «Сапёра», автоматически, даже не задумываясь, щёлкая по клеткам. Однако, человеческий способ мышления отличается от компьютерных операций: гораздо проще и понятнее действовать методом предположений, дающим тот же результат. Предположим, что в данной клетке стоит мина (или её там точно нет). Тогда если возникает ситуация, приводящая к ошибке, то, в таком случае, можно утверждать о ложности предположения и присвоить соответственный статус клетке: можно открыть или там мина.

Вообще, и такой алгоритм не решит «Сапёра» во всех случаях, в частности, потому что есть такие неразрешимые ситуации, как на рисунке 15. Как правильно расставить здесь мины, непонятно.

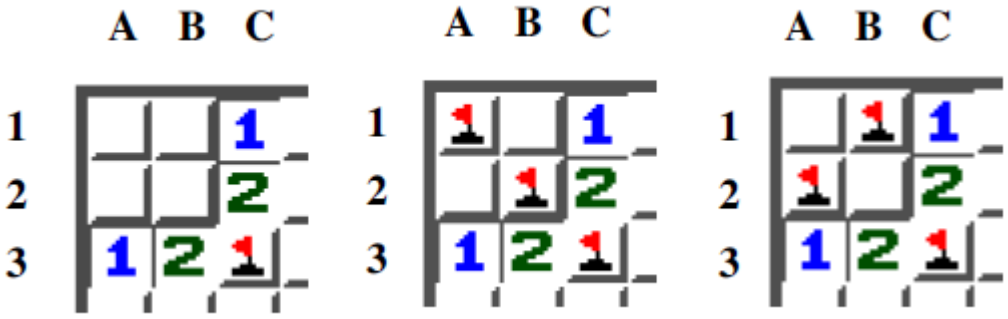

Рисунок 15 — Пример 8

Всё гениальное — просто, и в пользу гениальности этой простой на первый взгляд игры говорит тот факт, что решение «Сапёра» является NP-полной задачей. Из-за этого, а также из-за возможности рассмотренных выше неразрешимых случаев (и других, более сложных, но базирующихся на идее существования нескольких правильных расстановок мин) создание алгоритма, дающего однозначный и правильный ответ, не представляется возможным. К тому же только настоящие сапёры ошибаются один раз, а игроки, даже профессиональные, часто вынуждены действовать наугад, оценивая вероятность ошибки. Как бы то ни было, угадывание выходит за границы компетенции программы.

1.4.3 Разработанный метод с элементами самообучения

Основная идея данного метода заключается в синтезе новых (производных) правил решения задачи на основе базовых правил.

Синтез новых правил будет происходить при помощи обучения на мини-полях размером 3*3, 4*4, 5*5 и 6*6 клеток. В данных полях для каждой клетки генерируются значения, которые могут находиться в клетках (это цифры [0; 8], мина (М), стена (С), закрытая клетка (З)). С помощью полного перебора значений в клетках и с помощью применения базовых правил выявляются:

- недопустимые (запретные) комбинации;
- единственно возможные комбинации (для прогнозирования значений в закрытых клетках).

Все выявленные комбинации записываются в отдельный список правил, которые являются производными правилами от базовых правил логической задачи.

Затем полученные правила анализируются и выявляются зависимости (для определения основных свойств отношений, применяемых в дискретной математике: рефлексивность, симметричность, транзитивность и др.). Это позволит компактнее сформировать уже синтезированные правила, а также провести синтез новых правил, таким образом сформировав набор шаблонов, по которым можно будет выявлять, есть ли в той или клетке мина.

Выявлять новые правила при помощи мини-полей будем итеративно. На первой итерации будут применяться только базовые правила игры. На последующих итерациях будут применяться правила, основанные на принципе «что, если», то есть, сначала, по определённым правилам, будет выдвигаться предположение для закрытой клетки (например: предположим, что в данной клетке мина) и, исходя из данного предположения, будут проверяться уже имеющиеся правила и синтезироваться новые.

Блок-схема данного метода представлена на рисунке 16.

Рассмотрим подробнее описание применения логики *n*-ого порядка.

Логика 1-го порядка основана на правиле поскольку в клетке 1 значение *x*, то в клетке 2 значение *y*. Пример логики 1-го порядка приведён на рисунке 17.

Логика 2-го порядка основана на следующем правиле. Исходя из базовых правил и правил, полученных в ходе применения логики 1-го порядка, установлено, что мина может быть в *клетке 1* или в *клетке 2*. Тогда можно выделить 2 пункта для синтеза новых правил:

- если мина находится в *клетке 1*, то, исходя из данного предположения, в соседних клетках будет определённый набор значений (назовём его *набор 1*). Если мина находится в *клетке 2*, то, исходя из данного предположения, в соседних клетках будет другой набор значений (назовём его *набор 2*). Тогда к *набору 1* и *набору 2*, а, точнее, к соответствующим закрытым значениям клеток *набора 1* и *набора 2* можно применить логическую операцию **И**, чтобы выявить схожие значения;

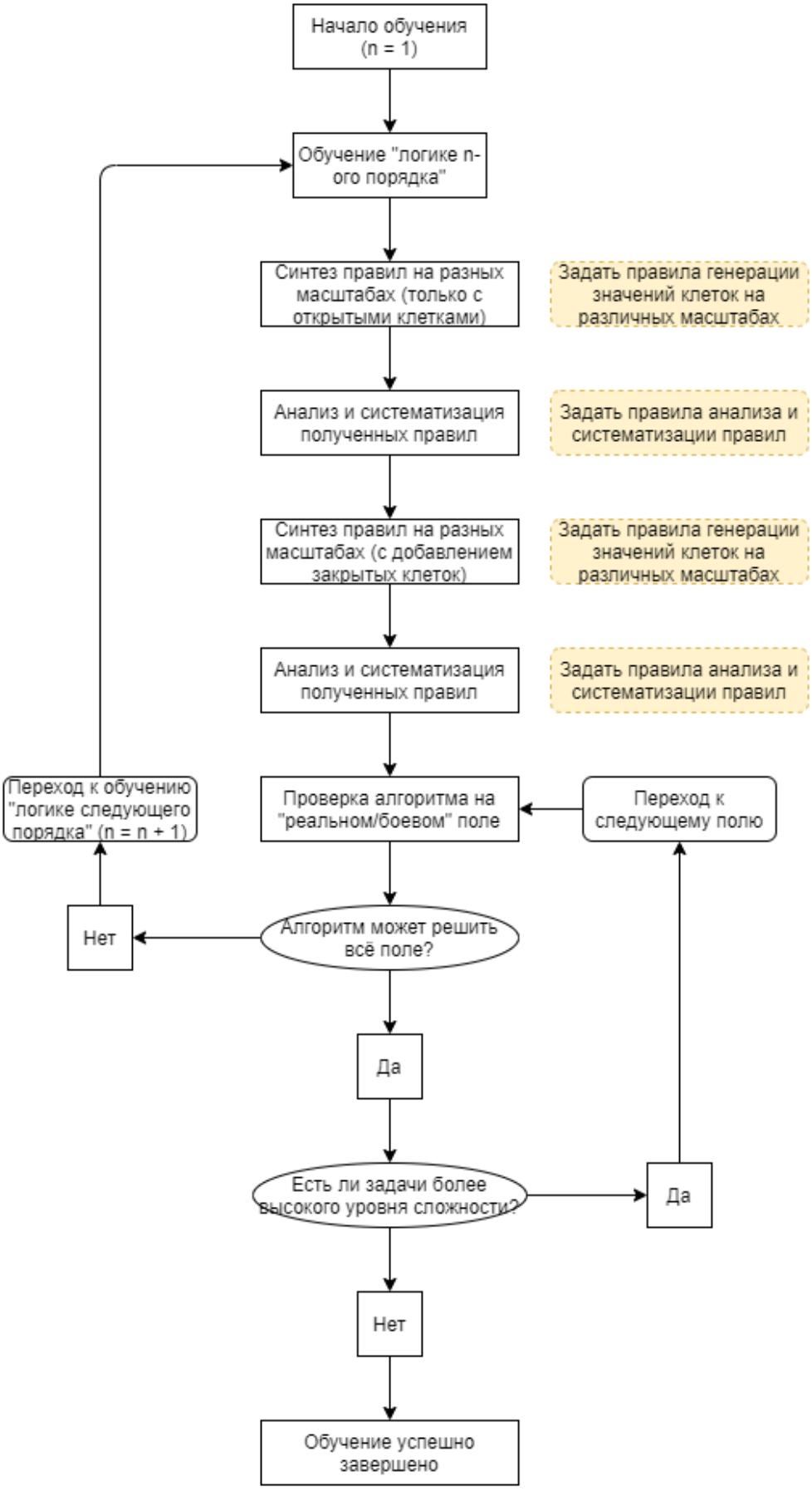

Рисунок 16 — Блок-схема разработанного алгоритма

Рисунок 17 — Пример логики 1-го порядка

– для *набора 1* и *набора 2* производится применение правил низших порядков для выявления ошибочных значений.

Таким образом, если в первом пункте будут найдены схожие значения для закрытых клеток двух полей и/или во втором пункте будут найдены ошибки при применении правил низших порядков, полученные схемы будут занесены в список правил 2-го порядка.

Пример применения первого пункта логики 2-го порядка представлен на рисунке 18.

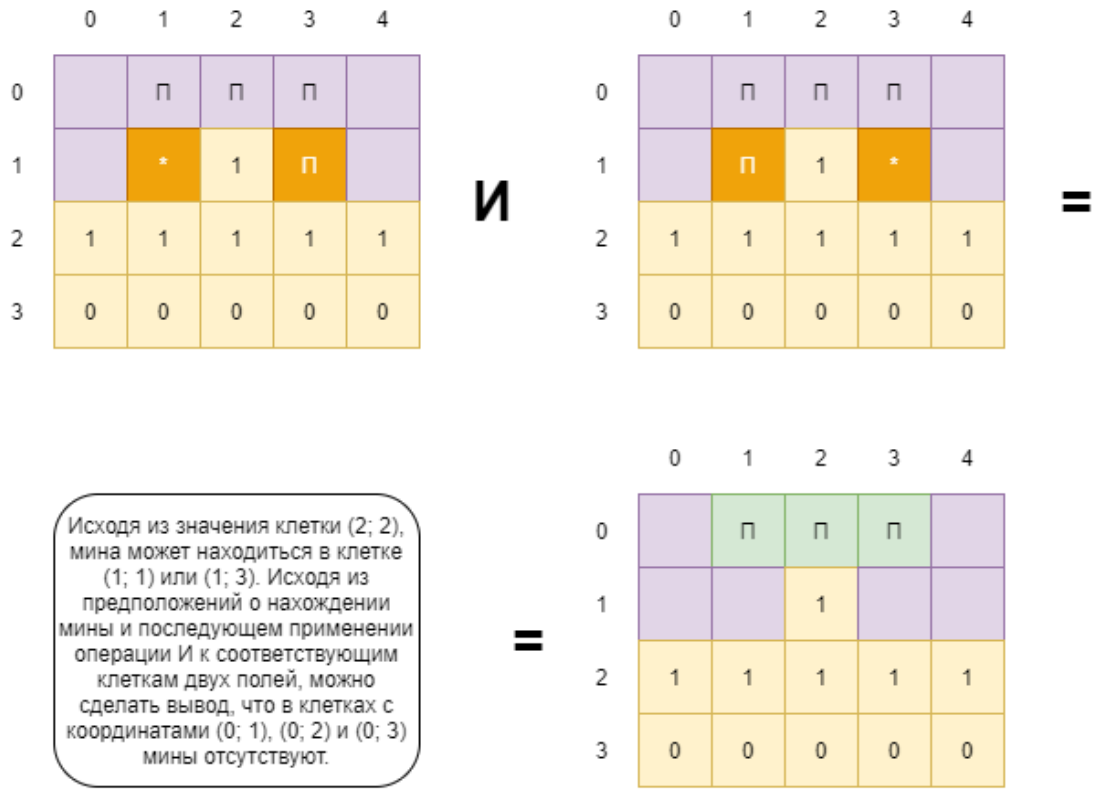

Рисунок 18 — Пример применения первого пункта логики 2-го порядка

1.4.4 Анализ алгоритмов построения самообучающихся систем

Определим теперь ряд методов, которые будут использоваться для построения системы с элементами самообучения.

Первый рассмотренный алгоритм основан на использовании вероятностной обучаемой стратегии. Данный алгоритм хоть и демонстрирует повышение эффективности при решении логической задачи «Сапёр», тем не менее, алгоритм не гарантирует безошибочность решения поля. Более того, данный алгоритм показывает невысокие значения доли побед для полей с числом мин q, равным 10 и 15 (0,694 и 0,469 соответственно), а для полей, с числом мин q, равным 20 алгоритм демонстрирует низкое значение доли побед (0,083). При заданных ограничениях применение данного подхода к разработке самообучающегося алгоритма недопустимо.

Второй и третий рассмотренные подходы основан на «запоминании» схем (шаблонов), которые необходимы для того, чтобы определить, находится ли в той или иной закрытой клетке мина или отсутствует. Важным фактом является то, что в случае, если не удастся найти хотя бы один подходящий шаблон, то система «уйдёт на дообучение», то есть будет искать новые схемы.

Исходя из представленного анализа, можно сделать вывод о том, что второй и третий методы для решения «Сапёра» являются наиболее подходящими, чем гибридная модель анализа ситуаций. Однако, важно в дальнейшем определить, какие из частей метода будут эффективны при решении задачи, а какие нет. Также представленные методы необходимо доработать, чтобы избежать возможных проблем при разработке системы с элементами самообучения.

2 СПЕЦИАЛЬНАЯ ЧАСТЬ

2.1 Постановка задачи исследования

Цель работы — разработка комплекса алгоритмов с элементами самообучения и их программной реализации для независимого от человека поиска эффективного решения каузально-логических игр (на примере игры «Minesweeper»/«Сапёр»).

Разработка специальной части начинается с описания правил игры «Сапёр». Далее, исходя из правил, будут представлены содержательная и математическая постановки задачи исследования. Исходя из них, будет разработана теоретическая часть, которая будет описывать связь входных и выходных данных. На основе теоретической части будет разработан алгоритм поиска эффективных решений, который впоследствии будет разделён на части (блоки), где далее для части блоков будет описан самообучающийся алгоритм, который будет выполнять функции данного блока. В завершении будет описано и представлено программное обеспечение, реализующее разработанный алгоритм с элементами самообучения, а также будут представлены результаты работы программы. Блок-схема специальной части представлена на рисунке 19.

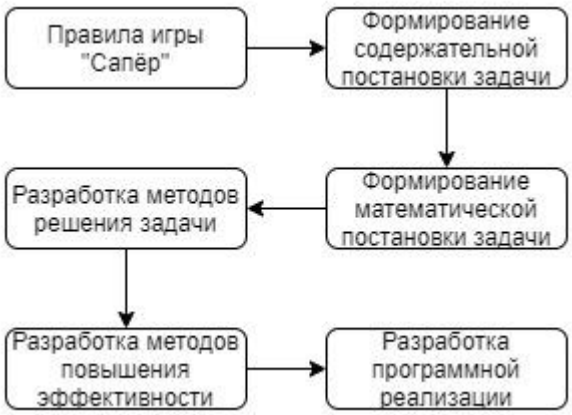

Рисунок 19 — Блок-схема специальной части

Правила игры «Сапёр» представлены в приложении А.

2.2 Содержательная постановка задачи

Содержательная постановка задачи представляет из себя выжимку и краткое содержание правил игры. Таким образом, содержательная постановка задачи — это точная и небольшая по объёму формулировка, понятная читателю.

Основным элементом игры является поле с заданными значениями длины и ширины. Элементами поля являются клетки. Клетка может быть открыта или закрыта. В открытой клетке хранится целое значение [0; 8] или мина. Значения [0; 8] в клетке означает количество мин, находящихся в соседних клетках. В закрытой клетке находится флаг мины, знак вопроса или не находится ничего — пользователь сам решает, что будет находиться в закрытой клетке. Если клетка закрыта, то неизвестно, что в ней находится, в противном случае — известно.

На вход подаётся поле, большая часть клеток которого закрыты, а также число, означающее общее количество мин, находящихся на поле. Пользователь может открыть закрытую клетку, чтобы узнать значение, которое в ней хранится. Если это значение [0; 8], то данное значение отображается в открываемой клетке. Если в открываемой клетке находится мина, то она также отображается в открываемой клетке, но при этом игра заканчивается поражением. Для победы необходимо открыть все клетки поля, в которых отсутствуют мины. Цель: победить.

Также определено, что решение задачи является детерминированным.

Таким образом, исходными данными для решения поставленной задачи являются:

- поле с заданными значениями длины и ширины, состоящее из клеток;
- общее количество мин, которое находится на поле.

Для исходных данных заданы следующие характеристики и логические связи:

1) клетка может быть открыта или закрыта;
2) если клетка закрыта, то неизвестно, что в ней находится, в противном случае — известно;
3) в открытой клетке хранится целое значение [0; 8] или мина;
4) в закрытой клетке находится флаг мины, знак вопроса или не находится ничего — пользователь сам решает, что будет находиться в закрытой клетке;
5) пользователь может открыть клетку, чтобы узнать значение в ней;
6) набор условий, определяющие правила игры, а именно:
    - цифра в клетке определяет количество мин в соседних клетках;
    - условие победы (цель игры): открыть все клетки поля, не содержащие мины;
    - условие поражения: открыть клетку с миной;
7) решение задачи является детерминированным.

Общая блок-схема поиска решения представлена на рисунке 20.

Рисунок 20 — Общая блок-схема поиска решения

2.3 Математическая постановка задачи

Математическая постановка задачи представляет из себя переработанную содержательную постановку задачи, представленную с помощью математических терминов. Большинство терминов, которые представлены в данном разделе можно найти в тезаурусе, но, для лучшего понимания, определение некоторых терминов будет приведено сразу.

2.3.1 Основные данные

Основным элементом задачи является кортеж F (field — поле), исходя из которого можно однозначно определить характеристики поля «Сапёра» (формула 2):

$$F =< h, w, tm, S, VOC, VCC, VC, MC, k > \quad (2)$$

Рассмотрим подробнее каждый из элементов кортежа F.

Определим матрицу S (status — статус) размером $h*w$, элементы которой отвечают за состояние клеток поля: закрыта или открыта. Определим множество SS (set of status — множество статусов) (формула 3):

$$S = \{O, C\}, \quad (3)$$

где O — открыта (open),

C — закрыта (close).

Элементы матрицы S принадлежат множеству SS.

Определим матрицу VOC (values in open cells — значения в открытых клетках) размером $h*w$, элементы которой отвечают за значение, которое находится в открытой клетке поля. Определим множество SVOC (set of values in open cells — множество значений в открытых клетках) (формула 4):

$$SVOC = \{0, 1, 2, 3, 4, 5, 6, 7, 8, M\}, \quad (4)$$

где M — мина (mine).

Элементы матрицы VOC принадлежат множеству SVOC.

Определим матрицу VCC (values in close cells — значения в закрытых клетках), элементы которого отвечают за значение, которое находится в закрытой клетке. Определим множество SVCC (set of values in close cells — множество значений в открытых клетках) (формула 5):

$$SVCC = \{E, MF, Q\}, \quad (5)$$

где E — пустота (emptiness),

MF — флаг мины (mine flag),

Q — вопрос (question).

Элементы матрицы VCC принадлежат множеству SVCC. Изначально каждый элемент матрицы VCC равен значению «E».

Определим матрицу VC (values of cells — значения клеток), элементы которого отвечают за значение клетки, которое отображается пользователю. Поскольку клетка может быть как открытой, так и закрытой, то элементы матрицы VC принадлежат объединению

множеств SVOC и SVCC, то есть $VC_{i,j} \in SVOC \cup SVCC$. Изначально элементы поля VC рассчитываются по формуле 6:

$$VC_{i,j} = \begin{cases} VOC_{i,j}, \text{если } S_{i,j} = O \\ VCC_{i,j}, \text{если } S_{i,j} = C \end{cases}; VC, VCC \in F \quad (6)$$

Теперь необходимо определить тот факт, что цифра в клетке означает количество мин, находящихся в соседних клетках. Для этого определим матрицу MC (mines in cells — мины в клетках), элементы которого отвечают за то, находится ли в заданной клетке поля мина или нет. Элементы матрицы MC принадлежат множеству нулей и единиц, где значение 0 означает, что в заданной клетке поля отсутствует мина, а значение 1 означает, что в заданной клетке поля находится мина. Элементы матрицы MC рассчитываются по формуле 7:

$$MC_{i,j} = \begin{cases} 0, \text{если } VOC_{i,j} \neq M \\ 1, \text{если } VOC_{i,j} = M \end{cases}; MC, VOC \in F \quad (7)$$

Общее количество мин на поле задаётся параметром *tm* (total mines — общее количество мин), значение которого рассчитывается по формуле 8:

$$tm = \sum_{i=1}^{l} \sum_{j=1}^{w} MC_{i,j}, tm, MC \in F \quad (8)$$

Рассмотрим теперь тот факт, что цифра в клетке означает количество мин в соседних клетках. Сначала определим соответствия между множеством координат клетки и множеством координат соседних клеток с заданной клеткой (формула 9):

$$\begin{cases} \{(1,1)\} \to \{(1,2),(2,1),(2,2)\} \\ \{(h,1)\} \to \{(h-1,2),(h-1,2),(h,2)\} \\ \{(h,w)\} \to \{(h-1,w-1),(h-1,w),(h,w-1)\} \\ \{(1,j)\} \to \{(1,j-1),(2,j-1),(2,j),(2,j+1),(1,j+1)\}, \\ \text{если } 1 < j < w \\ \{(i,w)\} \to \begin{Bmatrix} (i-1,w),(i-1,w-1),(i,w-1), \\ (i+1,w-1),(i+1,w) \end{Bmatrix}, \text{если } 1 < i < h \\ \{(h,j)\} \to \begin{Bmatrix} (h,j-1),(h-1,j-1),(h-1,j), \\ (h-1,j+1),(h,j+1) \end{Bmatrix}, \text{если } 1 < j < w \\ \{(i,1)\} \to \{(i-1,1),(i-1,2),(i,2),(i+1,2),(i+1,1)\}, \\ \text{если } 1 < i < h \\ \{(i,j)\} \to \begin{Bmatrix} (i-1,j-1),(i-1,j),(i-1,j+1),(i,j-1), \\ (i,j+1),(i+1,j-1),(i+1,j),(i+1,j+1) \end{Bmatrix}, \\ \text{если } 1 < i < h, 1 < j < w \end{cases} \quad (9)$$

Теперь можно часть содержательной постановки задачи, в которой говорится, что число в клетке означает количество мин в соседних клетках, записать в виде системы равенств (формула 10):

$$\begin{cases} VOC_{1,1} = MC_{1,2} + MC_{2,1} + MC_{2,2}, \text{если } VOC_{1,1} \neq M \\ VOC_{1,w} = MC_{1,w-1} + MC_{2,w-1} + MC_{2,w}, \text{если } VOC_{1,w} \neq M \\ VOC_{h,1} = MC_{h-1,1} + MC_{h-1,2} + MC_{h,2}, \text{если } VOC_{h,1} \neq M \\ VOC_{h,w} = MC_{h-1,w-1} + MC_{h-1,w} + MC_{h,w-1}, \text{если } VOC_{h,w} \neq M \\ VOC_{1,j} = MC_{1,j-1} + MC_{2,j-1} + MC_{2,j} + \\ +MC_{2,j+1} + MC_{1,j+1}, \text{если } 1 < j < w, \text{если } VOC_{1,j} \neq M \\ VOC_{i,w} = MC_{i-1,w} + MC_{i-1,w-1} + MC_{i,w-1} + \\ +MC_{i+1,w-1} + MC_{i+1,w}, \text{если } 1 < i < h, \text{если } VOC_{i,w} \neq M \\ VOC_{h,j} = MC_{h,j-1} + MC_{h-1,j-1} + MC_{h-1,j} + \\ +MC_{h-1,j+1} + MC_{h,j+1}, \text{если } 1 < j < w, \text{если } VOC_{h,j} \neq M \\ VOC_{i,1} = MC_{i-1,1} + MC_{i-1,2} + MC_{i,2} + \\ +MC_{i+1,2} + MC_{i+1,1}, \text{если } 1 < i < h, \text{если } VOC_{i,1} \neq M \\ VOC_{i,j} = MC_{i-1,j-1} + MC_{i-1,j} + MC_{i-1,j+1} + \\ MC_{i,j-1} + MC_{i,j-1} + MC_{i+1,j-1} + MC_{i+1,j} + \\ +MC_{i+1,j+1}, \text{если } 1 < i < h, 1 < j < w, \text{если } VOC_{i,j} \neq M \end{cases} \quad (10)$$

Зададим нумерацию уравнений/равенств системы 10. Поскольку каждое уравнение содержит элемент матрицы VOC с координатами (i, j), то будем определять номер уравнения, исходя из координат (i, j). Так, уравнение $VOC_{1,1} = MC_{1,2} + MC_{2,1} + MC_{2,2}$ имеет номер (1, 1), уравнение $VOC_{1,2} = MC_{1,1} + MC_{2,1} + MC_{2,2} + MC_{2,3} + +MC_{1,3}$ имеет номер (1, 2) и т. д.

Определим теперь матрицу k, элементы которой также являются матрицами. С помощью данной матрицы возможно определить, является ли клетка с координатами (i_2, j_2)

соседней для клетки с координатами (i_1, j_1). Таким образом, элементы матрицы каждой из элементов матрицы k рассчитываются по формуле 11:

$$k_{i_1,j_1,i_2,j_2} = \begin{cases} 1, \text{если } \{(i_1, j_1)\} \to \{(i_2, j_2)\} \\ 0, \text{если } \{(i_1, j_1)\} \nrightarrow \{(i_2, j_2)\} \end{cases} \quad (11)$$

Тогда систему 10 можно записать в следующем виде (формула 12):

$$\{VOC_{i,j} = \sum_{i_2=1}^{l} \sum_{j_2=1}^{w} k_{i,j,i_1,j_1} * MC_{i_1,j_1}, \quad (12)$$
$$1 \le i \le h, 1 \le j \le w, VOC_{i,j} \ne M;\ VOC, k, MC, h, w \in F$$

Или в общем виде (формула 13):

$$VOC = k * MC, VOC_i \ne M, \quad (13)$$

где VOC — вектор-столбец длиной $l * w - tm$,

k — матрица размером $(l * w, l * w)$,

MC — вектор-столбец длиной $l * w$.

Здесь и далее имеется в виду, что $VOC, k, MC \in F$.

Для тех элементов матриц VOC и MC, для координат (i, j) которых выполняется равенство $S_{i,j} = C$ $(S \in F)$ вместо значений $VOC_{i,j}$ определим переменные $y_{i,j}$, а вместо значений $MC_{i,j}$ определим переменные $x_{i,j}$. Тогда систему 13 можно записать в следующем виде (формула 14):

$$k_1 * VOC + (\hat{v} - k_1) * y = k_{21} * MC + k_{22} * x, \quad (14)$$

где k_1 — бинарный вектор-столбец, значения которого определяют, открыта ли данная клетка или нет (1, если открыта; 0, если закрыта),

$\hat{v}$ — единичный вектор,

k_{21} — сильно разряженная бинарная матрица, значения которого определяют, является ли заданная клетка соседней открытой с заданной клеткой клеткой (1, если является; 0, если не является),

k_{22} — сильно разряженная бинарная матрица (двумерный кортеж), значения которого определяют, является ли заданная клетка соседней закрытой с заданной клеткой клеткой (1, если является; 0, если не является) ($k_{21} + k_{22} = k$).

Поскольку в уже открытых клетках не может находиться мина по определению, то будет верно следующее равенство (формула 15).

$$k_{21} * MC = 0 \tag{15}$$

Таким образом, систему 14 можно записать в следующем виде (формула 16):

$$k_1 * VOC + (\hat{v} - k_1) * y = k_{22} * x \tag{16}$$

Теперь определим тот факт из содержательной постановки задачи, что при открытии закрытой клетки поля, в ней отображается имеющееся значение. В системе (16) неизвестными являются значения y и x, однако, значение переменной $y_{i,j}$ однозначно определяется при вычислении переменной $x_{i,j}$ как $y_{i,j} = VOC_{i,j}$.

Также для формулы расчёта общего количества мин на поле учтём, что для части элементов матрицы MC определены переменные x. Тогда формулу 8 можно переписать в следующем виде (формула 17):

$$tm = \sum_{S_{i,j}=C} x_{i,j} + \sum_{S_{i,j}=O} MC_{i,j} \tag{17}$$

Опять же, поскольку в уже открытых клетках не может находиться мина по определению, формулу 17 можно записать в следующем виде (формула 18):

$$tm = \sum_{S_{i,j}=C} x_{i,j} \tag{18}$$

Добавим данное уравнение в систему 16 и получим формулу 19:

$$\begin{cases} k_1 * VOC + (\hat{v} - k_1) * y = k_{22} * x \\ tm = \sum_{S_{i,j}=C} x_{i,j} \end{cases} \tag{19}$$

Определим номер данного уравнения в системе 19 как (0, 0).

Визуально основные элементы математической постановки задаче изображены на рисунке 21.

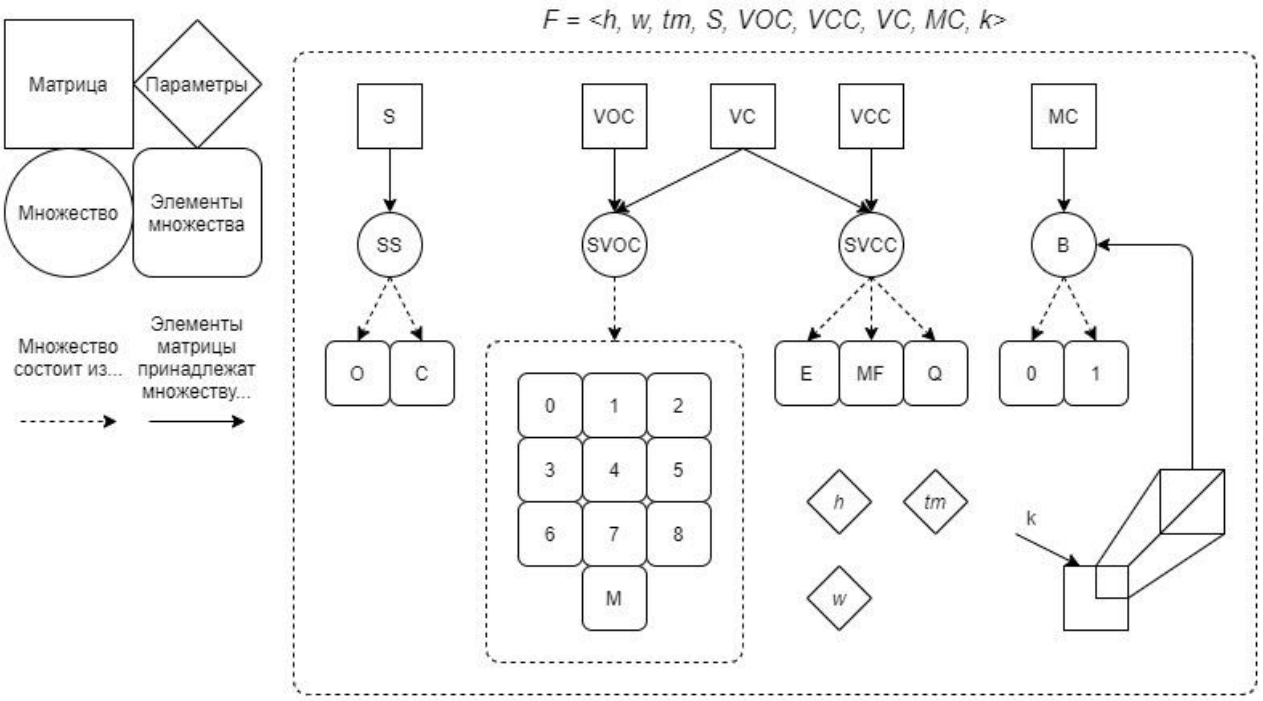

Рисунок 21 — Основные элементы математической постановки задачи

Теперь можно дать определение значению детерминированности решения игры «Сапёр». Решение игры «Сапёр» будет считаться детерминированным, если система 19 имеет единственное решение.

Определим кортеж F_{id} (fields input data — входные данные полей), который состоит из первых пяти элементов поля F (формула 20):

$$F_{id} =< h, w, tm, S, VOC >; h, w, tm, S, VOC \in F \quad (20)$$

Также определим кортеж TF_{id} (tuple of fields input data — кортеж полей исходных данных), который состоит из кортежей F_{id} (формула 21):

$$TF_{id} =< F_{id_1}, F_{id_2}, \dots, F_{id_n} > \quad (21)$$

Таким образом, исходным данным является кортеж TF_{id}.

Успешным выполнением задачи является нахождение решения каждой из систем 19, а именно, вычисление значений $x_{i,j}$ для тех координат (i, j), для которых выполняется условие $S_{i,j} = C, S \in F$, которая определяется исходя из элементов кортежа TF_{id}.

2.3.2 Дополнительные данные

Назовём дополнительными данными такие данные, которые не используются для достижения цели, но которые присутствуют в содержательной постановке задачи и которые должны быть формализованы.

Определим множество пользовательских допустимых изменений состояний (переходов) UT (user transition — пользовательский переход) и множество условных переходов CT (conditional transition — условный переход). Будем называть пользовательским переходом такое изменение переменной или элемента матрицы, которое может осуществить пользователь. Условным переходом будем называть такое изменение переменной и/или элемента или элементов тех или иных двумерных кортежей/матриц, которое происходит в ответ на изменение элементов с помощью пользовательского перехода. Условный переход состоит из двух частей: условной (которое, в свою очередь является пользовательским переходом) и действенной части (определяет переход, который осуществляется при условии выполнения указанного пользовательского перехода). Множество UT состоит из следующих элементов:

- Для $S_{i,j}: C \to O$;
- Для $CCV_{i,j}: E \to MF$ при $S_{i,j} = \text{C}$;
- Для $CCV_{i,j}: E \to Q$ при $S_{i,j} = \text{C}$;
- Для $CCV_{i,j}: MF \to E$ при $S_{i,j} = \text{C}$;
- Для $CCV_{i,j}: MF \to Q$ при $S_{i,j} = \text{C}$;
- Для $CCV_{i,j}: Q \to MF$ при $S_{i,j} = \text{C}$;
- Для $CCV_{i,j}: Q \to E$ при $S_{i,j} = \text{C}$.

Пользовательские переходы представлены на рисунке 22.

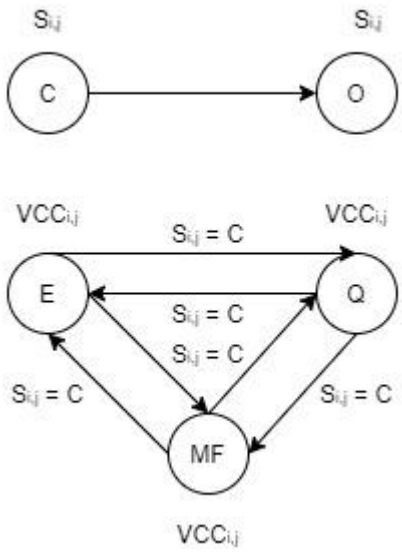

Рисунок 22 — Пользовательские переходы

Множество СТ состоит из следующих элементов:

– ЕСЛИ $S_{i,j}: C \rightarrow O$,

ТО $VC_{i,j}: E \rightarrow X$, где $X \in SVOC$;

– ЕСЛИ $S_{i,j}: C \rightarrow O$,

ТО $VC_{i,j}: Q \rightarrow X$, где $X \in SVOC$;

– ЕСЛИ $VCC_{i,j}: E \rightarrow MF$,

ТО $VC_{i,j}: E \rightarrow MF$;

– ЕСЛИ $VCC_{i,j}: E \rightarrow Q$,

ТО $VC_{i,j}: E \rightarrow Q$;

– ЕСЛИ $VCC_{i,j}: MF \rightarrow E$,

ТО $VC_{i,j}: MF \rightarrow E$;

– ЕСЛИ $VCC_{i,j}: MF \rightarrow Q$,

ТО $VC_{i,j}: MF \rightarrow Q$;

– ЕСЛИ $VCC_{i,j}: Q \rightarrow MF$,

ТО $VC_{i,j}: Q \rightarrow MF$;

– ЕСЛИ $VCC_{i,j}: Q \rightarrow E$,

ТО $VC_{i,j}: Q \rightarrow E$.

Условные переходы представлены на рисунке 23.

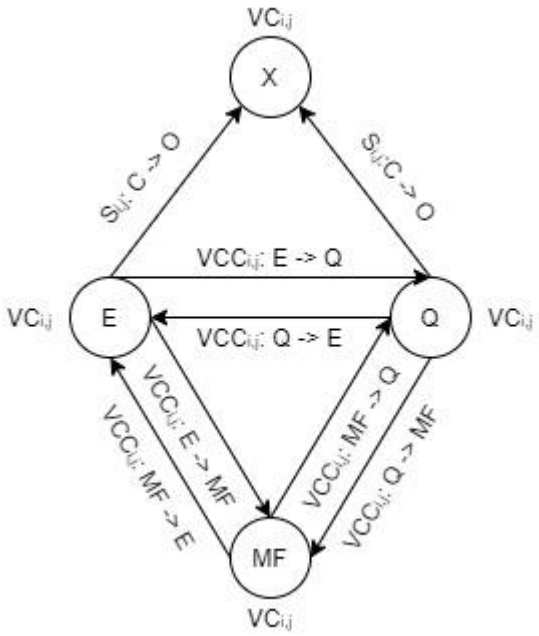

Рисунок 23 — Условные переходы

2.4 Методы поиска решения

Рассмотрим далее несколько методов поиска решения поставленной задачи. Для начала рассмотрим систему 19 и определим её свойства:

- система 19 является дискретной. Это означает, что переменные *x* и *y* системы могут принимать только дискретные значения, а также все параметры системы могут принимать только дискретные значения;
- система 19 является бинарной, поскольку переменные *x* системы могут принимать только значения 0 или 1. Также элементы параметра-матрицы системы k_{22} также могут принимать только значения 0 или 1;
- система 19 является сильно разряженной. Это означает, что параметр-матрица системы k_{22} содержит преимущественно нулевые элементы;
- система 19 является зависимой. Это означает, что переменные *y* системы нельзя вычислить «напрямую», то есть, значение переменной *y* с заданными координатами становится известно, когда становится известно значение переменной *x* с соответствующими координатами;
- система 19 является детерминированной. Это означает, что система имеет единственное решение;
- исходя из того, что параметр-матрица системы 19 k_{22} содержит преимущественно нулевые элементы, некоторые уравнения системы могут быть преобразованы в равенства.

Исходя из того, что система 19 является зависимой, найти решение тех уравнений системы 19, для которых значение параметра $k_1 \neq 0$ невозможно. Таким образом, необходимо сосредоточиться на поиске решения тех уравнений из системы 19, которые можно записать в следующем виде (формула 22):

$$VOC = k_{22} * x, \tag{22}$$

а также на последнем уравнении системы 19 (формула 23):

$$tm = \sum_{S_{i,j}=C} x_{i,j} \tag{23}$$

2.4.1 Метод однозначного вычисления значений в соседних клетках

Рассмотрим первый способ поиска решения системы 19.

2.4.1.1 Содержательное описание метода

Поскольку переменные $x_{i,j}$ системы уравнений 19 имеют дискретные значения, то одним из простых способов поиска решения системы является перебор допустимых значений для каждого уравнения системы. Тогда если любое уравнение системы 19 будет верным только при единственной комбинации значений переменных $x_{i,j}$, принадлежащих ОДЗ, то данные переменные $x_{i,j}$ равны именно этим определённым значениям.

Данный метод можно представить следующим образом. Рассмотрим рисунок 24. На нём представлено небольшое поле «Сапёра». Здесь и далее будем обозначать цветом клетки:

- белая клетка — это открытая клетка поля, в которой находится значение;
- синяя клетка — закрытая клетка поля;
- зелёная клетка — клетка поля, которая рассматривается в данной ситуации;
- красная клетка — открытая клетка поля, для которой не выполняется правило, что число в клетке равно количеству мин в соседних клетках.

Проверим возможные значения в закрытой клетке и посмотрим, как данные значения будут влиять на правило о количестве мин в соседних клетках для зелёной клетки. Предположим, что в синей клетке отсутствует мина, тогда количество соседних для зелёной клетки клеток без мин будет равно нулю, что не соответствует значению в зелёной клетке. Если в синей клетке находится мина, то количество соседних для зелёной клетки клеток без

мин будет равно единице, что соответствует значению в зелёной клетке. Поскольку количество соседних закрытых с зелёной клеткой клеток всего одна и для неё верно только одно значение, то данное значение и будет в данной клетке.

Рисунок 24 — Пример 1 для метода однозначного определения значений в соседних клетках

Рассмотрим теперь пример 2 (рисунок 25). Проверим возможные значения в закрытых клетках и посмотрим, как данные значения будут влиять на правило о количестве мин в соседних клетках для зелёной клетки. Предположим, что в обоих синих клетках отсутствуют мины, тогда количество соседних для зелёной клетки клеток без мин будет равно нулю, что не соответствует значению в зелёной клетке. Пусть тогда в одной из синих клеток находится мина, а в другой отсутствует. В таком случае количество соседних для зелёной клетки клеток без мин будет равно единице, что также не соответствует значению в зелёной клетке. Предположим теперь, что в обоих синих клетках находятся мины, тогда количество соседних для зелёной клетки клеток без мин будет равно двум, что соответствует значению в зелёной клетке. Поскольку количество соседних закрытых с зелёной клеткой клеток две и для каждой из них верно только одно значение, то данное значение и будет в данной клетке.

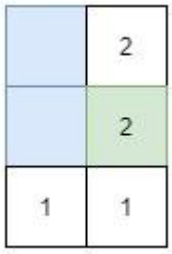

Рисунок 25 — Пример 2 для метода однозначного определения значений в соседних клетках

2.4.1.2 Математическое описание метода

Рассмотрим одно из уравнений системы 19 при $k_1 = 1$ (формула 24):

$$VOC_{i,j} = \sum_{i_1} \sum_{j_1} k_{22\ i,j,i_1,j_1} * x_{i_1,j_1} \quad (24)$$

Тогда для данного уравнения можно записать первое правило вычисления значений $x_{i,j}$:

– если $\sum_{i_1} \sum_{j_1} k_{22\ i,j,i_1,j_1} = VOC_{i,j}$, то для тех значений i_1 и j_1, для которых выполняется равенство $k_{22\ i,j,i_1,j_1} = 1$, будет верным равенство $x_{i_1,j_1} = 1$;

– если $VOC = 0$, то для тех значений i_1 и j_1, для которых выполняется равенство $k_{22\ i,j,i_1,j_1} = 1$, будет верным равенство $x_{i_1,j_1} = 0$.

2.4.2 Метод гипотез

Рассмотрим второй способ поиска решения системы 19.

2.4.2.1 Содержательное описание метода

Рассмотрим ситуацию для системы 19, когда для переменной $x_{i,j}$ задаём некоторое значение $x_{i,j} = 0$ или $x_{i,j} = 1$. Поскольку система 19 имеет единственное решение, существует 2 варианта:

– если заданное значение $x_{i,j}$ не равно истинному значению данной переменной, то при подстановке данного значения в другие уравнения системы и при вычислении других значений $x_{i,j}$ с применением первого правила может быть найдено такое уравнение или равенство системы, которое не имеет решений на ОДЗ (если это уравнение) или являются неверным (если это равенство);

– если заданное значение $x_{i,j}$ равно истинному значению данной переменной, то при подстановке данного значения в другие уравнения системы и при вычислении других значений $x_{i,j}$ с применением первого правила не может быть найдено такое уравнение или равенство системы, которое не имеет решений на ОДЗ (если это уравнение) или являются неверным (если это равенство).

Стоит заметить, что при данном предположении нельзя вычислять значения $y_{i,j}$, поскольку в данном методе делается лишь предположение о возможном значении $x_{i,j}$, а $y_{i,j}$ становится известным только когда однозначно вычислено значение $x_{i,j}$.

Данный метод можно визуализировать следующим образом. Рассмотрим рисунок 26. На нём представлено небольшое поле «Сапёра». Определим для каждой клетки нумерацию. Попробуем применить данный метод для клетки с координатами (2, 2).

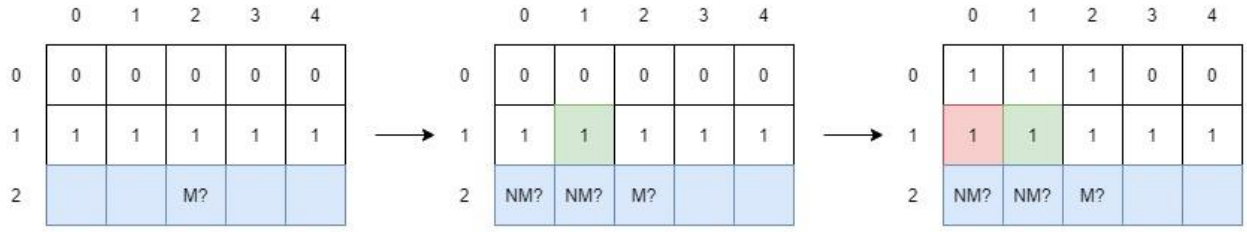

Рисунок 26 — Пример для метода гипотез

Предположим, что в клетке с координатами (2, 2) находится мина (рисунок 27, левое поле). Тогда попытаемся вычислить значения в других закрытых клетках, исходя из значений в открытых клетках. Рассмотрим клетку с координатами (1, 1). В соседних с ней клетках находится всего одна мина и, поскольку было выдвинуто предположение о том, что в клетке (2, 2) находится мина (а данная клетка является соседней для клетки с координатами (1, 1)), то в клетках с координатами (2, 0) и (2, 1) отсутствуют мины (рисунок 27, среднее поле). Однако, можно заметить, что в соседних клетках с клеткой с координатами (1, 0) отсутствуют мины, хотя в соседних клетках должна находиться хотя бы одна мина. Таким образом, можно сделать вывод о том, что предположение о том, что в клетке с координатами (2, 2) находится мина — неверное.

2.4.2.2 Математическое описание метода

Рассмотрим систему 19 для рисунка 26 (формула 25):

$$\begin{cases} x_{2,0} + x_{2,1} = 1 \\ x_{2,0} + x_{2,1} + x_{2,2} = 1 \\ x_{2,1} + x_{2,2} + x_{2,3} = 1 \\ x_{2,2} + x_{2,3} + x_{2,4} = 1 \\ x_{2,3} + x_{2,4} = 1 \end{cases} \quad (25)$$

Первое уравнение системы соответствует клетке с координатами (1, 0), второе — (1, 1), третье — (1, 2), четвёртое — (1, 3) и пятое — (1, 4). По аналогии с представленным примером для рисунка 23 предположим, что переменная $x_{2,2} = 1$. Тогда уравнение 2 системы 25 будет следующим (формула 26):

$$x_{2,0} + x_{2,1} = 0 \quad (26)$$

Поскольку ОДЗ для переменных $x_{i,j} \in \{0, 1\}$, то уравнение 26 будет верным только при $x_{2.0} = 0$ и $x_{2.1} = 0$, но, в таком случае, при подстановке данных значений в уравнение 1 системы 25 получается неверное равенство 0 = 1. Исходя из этого, можно сделать вывод о том, что предположение $x_{2,2} = 1$ — неверное.

2.4.3 Метод связанных клеток 1

Рассмотрим третий способ поиска решения системы 19.

2.4.3.1 Содержательное описание метода

Данный метод основан на методе исключения переменных из системы 19 без применения последнего уравнения системы, в котором отражается подсчёт общего количества мин на поле.

Данный метод можно представить следующим образом. Рассмотрим рисунок 27. На нём представлено небольшое поле «Сапёра». Определим для каждой клетки нумерацию. Рассмотрим клетку с координатами (1, 0). В соседних с данной клеткой клетках должна находиться одна мина. Поскольку клетки с координатами (0, 0), (0, 1) и (1, 1) открыты и в них отсутствуют мины, то мина находится в одной из двух клеток с координатами (2, 0) или (2, 1). Рассмотрим теперь клетку с координатами (1, 1). В соседних с ней клетках также должна находиться одна мина. Поскольку клетки с координатами (1, 0), (0, 0), (0, 1), (0, 2) и (1, 2) открыты и в них отсутствуют мины, то мина должна находиться в одной из трёх соседних клеток с координатами (2, 0), (2, 1) или (2, 2). Однако, при рассмотрении клетки с координатами (1, 0) было определено, что одна мина однозначно находится в одной из клеток с координатами (2, 0) или (2, 1). Исходя из этого, получается, что в клетке с координатами (2, 2) отсутствует мина, поскольку единственная мина, которая находится в соседних с клеткой с координатами (1, 1) клетках находится в одной из клеток с координатами (2, 0) или (2, 1), а, в таком случае, в остальных соседних клетках мины отсутствуют.

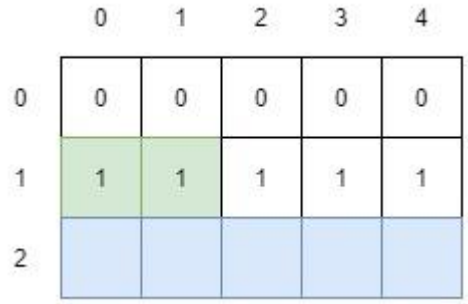

Рисунок 27 — Пример 1 для метода связанных клеток 1

Рассмотрим ещё один пример работы данного метода (рисунок 28). Рассмотрим клетку с координатами (1, 2). В соседних с данной клеткой клетках находится одна мина. Мина может находиться в одной из двух клеток с координатами (2, 1) или (2, 3). Рассмотрим теперь клетку с координатами (2, 2). В соседних с данной клеткой клетках также находится одна мина. Мина может находиться в одной из пяти клеток с координатами (2, 1), (3, 1), (3, 2), (3, 3), (2, 3), но, поскольку при рассмотрении клетки с координатами (1, 2) было определено, что мина находится в одной из двух клеток с координатами (2, 1) или (2, 3), следовательно, в клетках с координатами (3, 1), (3, 2) и (3, 3) мины отсутствуют.

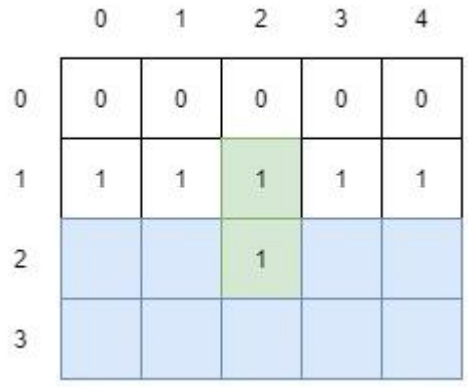

Рисунок 28 — Пример 2 для метода связанных клеток 1

2.4.3.2 Математическое описание метода

Математическое описание метода следующее. Рассмотрим систему 19. Необходимо из данной системы уравнений найти два таких уравнений, чтобы для уравнения, полученного при вычитании одного уравнения из другого, оказался применим первый метод. При этом нельзя

использовать последнее уравнение системы, в котором отображается подсчёт общего количества мин на поле.

Рассмотрим несколько примеров.

Рассмотрим систему 19 для рисунка 27 (формула 27):

$$\begin{cases} x_{2,0} + x_{2,1} = 1 \\ x_{2,0} + x_{2,1} + x_{2,2} = 1 \\ x_{2,1} + x_{2,2} + x_{2,3} = 1 \\ x_{2,2} + x_{2,3} + x_{2,4} = 1 \\ x_{2,3} + x_{2,4} = 1 \end{cases} \quad (27)$$

Рассмотрим первое и второе уравнения данной системы. Вычтем первое уравнение системы из второго, таким образом, исключив переменные $x_{2,0}$ и $x_{2,1}$ (формула 28):

$$\begin{gathered} \left(x_{2,0} + x_{2,1} + x_{2,2}\right) - \left(x_{2,0} + x_{2,1}\right) = 1 - 1 \\ x_{2,2} = 0 \end{gathered} \quad (28)$$

Таким образом, получилось найти однозначное значение переменной $x_{2,2}$.

Рассмотрим систему 19 для рисунка 28 (формула 29):

$$\begin{cases} x_{2,0} + x_{2,1} = 1 \\ x_{2,0} + x_{2,1} = 1 \\ x_{2,1} + x_{2,3} = 1 \\ x_{2,3} + x_{2,4} = 1 \\ x_{2,3} + x_{2,4} = 1 \\ x_{2,1} + x_{3,1} + x_{3,2} + x_{3,3} + x_{2,3} = 1 \end{cases} \quad (29)$$

Поясним сначала, почему в системе присутствуют повторяющиеся уравнения 1, 2 и 4, 5. Уравнение 1 соответствует клетке с координатами (1, 0), уравнение 2 соответствует клетке с координатами (1, 1). Исходя из того, что в клетках с данными координатами находится цифра 1 и того, что в соседних с данными клетками находятся две одинаковые закрытые клетки, получается, что уравнения одинаковые для двух данных клеток. Аналогичная ситуация для уравнений 4 и 5.

Теперь рассмотрим третье и шестое уравнения данной системы. Вычтем третье уравнение системы из шестого, таким образом, исключив переменные $x_{2,1}$ и $x_{2,3}$ (формула 30):

$$\begin{gathered}\left(x_{2,1}+x_{3,1}+x_{3,2}+x_{3,3}+x_{2,3}\right)-\left(x_{2,1}+x_{2,3}\right)=1-1\\ x_{3,1}+x_{3,2}+x_{3,3}=0\end{gathered} \quad (30)$$

Уравнение 30 имеет единственное решение на ОДЗ, а именно (формула 31):

$$x_{3,1}=0, x_{3,2}=0, x_{3,3}=0 \quad (31)$$

2.4.4 Метод связанных клеток 2

Рассмотрим четвёртый способ поиска решения системы 19.

2.4.4.1 Содержательное описание метода

Данный метод основан на методе исключения переменных из системы 19, но теперь уже только с применением последнего уравнения системы, в котором отражается подсчёт общего количества мин на поле.

Данный метод можно представить следующим образом. Рассмотрим рисунок 29. На нём представлены три небольших поля «Сапёра». Определим для каждой клетки нумерацию. Рассмотрим сначала левое поле (рисунок 30). Известно, что общее количество мин на данном поле — 2. Используем данную информацию, чтобы вычислить значения в закрытых клетках поля. Рассмотрим клетку с координатами (2, 2). В данной клетке находится цифра 2, что означает, что в соседних с данной клеткой клетках находится 2 мины. Поскольку известно, что на поле всего находится 2 мины и эти 2 мины находятся в соседних с клеткой с координатами (2, 2) клетках (данные клетки поля отмечены розовым цветом), то в других клетках поля мины отсутствуют (а именно в клетках с координатами (1, 0), (0, 0), (0, 1), (0, 3), (0, 4), (1, 4), (3, 4), (4, 4), (4, 3), (4, 1), (4, 0), (3, 0)). Таким образом, получилось вычислить значение в двенадцати закрытых клеток поля (рисунок 30, среднее поле). Исходя из полученных значений несложно вычислить значения в оставшихся четырёх закрытых клетках поля (рисунок 30, правое поле).

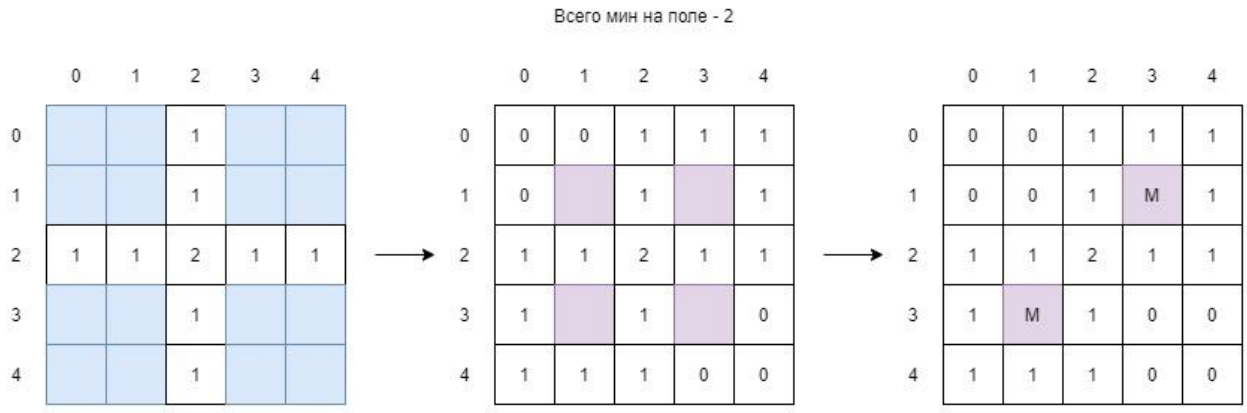

Рисунок 29 — Пример для метода связанных клеток 2

2.4.4.2 Математическое описание метода

Математическое описание метода следующее. Рассмотрим систему 19. Необходимо из данной системы уравнений найти два таких уравнений, чтобы для уравнения, полученного при вычитании одного уравнения из другого, оказался применим первый метод. При этом необходимо использовать последнее уравнение системы, в котором отображается подсчёт общего количества мин на поле.

Рассмотрим систему 19 для рисунка 29 (формула 32):

$$\begin{cases} x_{0,1} + x_{0,3} + x_{1,1} + x_{1,3} = 1 \\ x_{0,1} + x_{0,3} + x_{1,1} + x_{1,3} = 1 \\ x_{1,1} + x_{1,1} + x_{3,0} + x_{3,1} = 1 \\ x_{1,1} + x_{1,1} + x_{3,0} + x_{3,1} = 1 \\ x_{1,1} + x_{1,3} + x_{3,1} + x_{3,3} = 2 \\ x_{1,3} + x_{1,4} + x_{3,3} + x_{3,4} = 1 \\ x_{1,3} + x_{1,4} + x_{3,3} + x_{3,4} = 1 \\ x_{3,1} + x_{4,1} + x_{3,3} + x_{4,3} = 1 \\ x_{3,1} + x_{4,1} + x_{3,3} + x_{4,3} = 1 \\ x_{0,0} + x_{0,1} + x_{0,3} + x_{0,4} + x_{1,0} + x_{1,1} + x_{1,3} + x_{1,4} + \\ + x_{3,0} + x_{3,1} + x_{3,3} + x_{3,4} + x_{4,0} + x_{4,1} + x_{4,3} + x_{4,4} = 2 \end{cases} \quad (32)$$

Рассмотрим пятое и последнее уравнения данной системы. Вычтем пятое уравнение системы из последнего, таким образом, исключив переменные $x_{1,1}, x_{1,3}, x_{3,1}$ и $x_{3,3}$ (формула 33):

$$
\begin{gathered}
(x_{0,0} + x_{0,1} + x_{0,3} + x_{0,4} + x_{1,0} + x_{1,1} + x_{1,3} + x_{1,4} + \\
+x_{3,0} + x_{3,1} + x_{3,3} + x_{3,4} + x_{4,0} + x_{4,1} + x_{4,3} + x_{4,4}) - \\
-(x_{1,1} + x_{1,3} + x_{3,1} + x_{3,3}) = 2 - 2 \\
x_{0,0} + x_{0,1} + x_{0,3} + x_{0,4} + x_{1,0} + x_{1,4} + \\
+x_{3,0} + x_{3,4} + x_{4,0} + x_{4,1} + x_{4,3} + x_{4,4} = 0
\end{gathered}
\tag{33}
$$

Уравнение 33 имеет единственное решение на ОДЗ, а именно (формула 34):

$$
\begin{gathered}
x_{0,1} = x_{1,0} = x_{0,3} = x_{0,4} = x_{1,0} = x_{1,4} = \\
= x_{3,0} = x_{3,4} = x_{4,0} = x_{4,1} = x_{4,3} = x_{4,4} = 0
\end{gathered}
\tag{34}
$$

2.5 Связь между методами поиска решения

Рассмотрим ещё раз методы решения системы 19. Для каждого метода определим результат работы метода. Результат работы метода будет положительным, если для заданных входных данных метода удалось однозначно определить одно из значений переменных *x* системы 19. В противном случае результат работы метода будет считаться отрицательным. Тогда для каждого метода определим:

- количество подаваемых на вход уравнений;
- зависит ли результат работы метода только от подаваемых на вход уравнений.

Для метода однозначного определения значений в соседних клетках на вход подаётся одно уравнение системы 19 и результат работы метода зависит только от подаваемого на вход уравнения. Для метода связанных клеток 1 на вход подаются 2 уравнения системы 19 (за исключением уравнения для расчёта общего количества мин) и результат работы метода зависит только от подаваемых на вход уравнений. Для метода связанных клеток 2 на вход подаётся *n* уравнений системы 19 (одно из входных уравнений обязательно является уравнением для расчёта общего количества мин) и результат работы метода зависит только от подаваемых на вход уравнений. Для метода гипотез на вход подаётся одно уравнение системы 19 и результат работы метода зависит не только от подаваемого на вход уравнения.

Рассмотрим подробнее метод гипотез. Рассмотрим ситуацию, когда результат работы метода гипотез является успешным и зависит только от входного уравнения системы. Рассмотрим в качестве примера следующее уравнение (формула 35):

$$
x_{1,2} + x_{2,2} + x_{2,1} = VOC_{1,1} \tag{35}
$$

Предположим, что $x_{1,2} = 1$, тогда уравнение 35 можно записать в следующем виде (формула 36):

$$x_{2,2} + x_{2,1} = VOC_{1,1} - 1 \quad (36)$$

Для того, чтобы результат работы метода был успешным и зависел только от входного уравнения, уравнение 36 не должно иметь решений на ОДЗ, но при этом уравнение 35 должно иметь решение на ОДЗ. Это возможно только при $VOC_{1,1} = 0$.

Предположим теперь, что $x_{1,2} = 0$, тогда уравнение 36 можно записать в следующем виде (формула 37):

$$x_{2,2} + x_{2,1} = VOC_{1,1} \quad (37)$$

Для того, чтобы результат работы метода был успешным и зависел только от входного уравнения, уравнение 37 не должно иметь решений на ОДЗ, но при этом уравнение 36 должно иметь решение на ОДЗ. Это возможно только при $VOC_{1,1} = 3$.

Исходя из представленных рассуждений можно заметить, что при условии применения только входного уравнения для метода гипотез, работа данного метода аналогична методу однозначного определения значений в соседних клетках. Таким образом, метод однозначного определения значений в соседних клетках является частным случаем метода гипотез.

Аналогично, рассматривая метод гипотез при условии успешного результата и зависимости результата только от входного и ещё одного уравнения системы, можно определить, что метод связанных клеток 1 также является частным случаем метода гипотез. А, если рассматривать метод гипотез при условии успешного результата и зависимости результата только от входного уравнения и ещё $n - 1$ уравнений системы, то можно определить, что метод связанных клеток 2 также является частным случаем метода гипотез.

Рассмотрим также метод связанных клеток 1. Результат данного метода зависит только от подаваемых на вход двух уравнений. Рассмотрим ситуации, когда каждое из входных уравнений метода связанных клеток сначала подаётся на вход метода однозначного вычисления значений в соседних клетках, определяется результат работы метода, и только потом подаётся на вход метода связанных клеток 1. Тогда возможны следующие комбинации результатов работы метода однозначного вычисления значений в соседних клетках для подаваемых на вход уравнений (У — это успешный результат работы метода, Н — неуспешный результат работы метода):

- (У, У);

- (У, Н);
- (Н, У);
- (Н, Н).

Для каждой из представленных комбинаций определим возможные варианты результата работы метода связанных клеток 1 (формулы 38, 39, 40):

$$(\text{У},\text{У}).\ \text{Н}:\begin{cases} x_{11}+x_{12}=0 \\ x_{21}+x_{22}=0 \end{cases},\ \text{У}:\begin{cases} x_{11}+x_{12}=2 \\ x_{21}+x_{22}=0 \end{cases} \tag{38}$$

$$(\text{У},\text{Н})/(\text{Н},\text{У}).\ \text{Н}:\begin{cases} x_{11}+x_{12}=2 \\ x_{21}+x_{22}=1 \end{cases},\ \text{У}:\begin{cases} x_{11}+x_{12}+x_{13}=2 \\ x_{11}+x_{12}=2 \end{cases} \tag{39}$$

$$(\text{Н},\text{Н}).\ \text{Н}:\begin{cases} x_{11}+x_{12}=1 \\ x_{21}+x_{22}=1 \end{cases},\ \text{У}:\begin{cases} x_{12}+x_{22}+x_{32}=2 \\ x_{22}+x_{32}+x_{42}=1 \end{cases} \tag{40}$$

Таким образом, вне зависимости от результата работы метода однозначного определения значений в соседних клетках для каждого из входных уравнений метода связанных клеток 1, результат работы данного метода может быть различным. Исходя из этого, можно сделать вывод, что метод связанных клеток 1 не может являться заменой методу однозначного определения значений в соседних клетках, однако, метод однозначного вычисления значений в соседних клетках является частным случаем метода связанных клеток 1 при условии, что одно из входных уравнений системы является равенством (формула 41):

$$\begin{cases} x_{11}+x_{12}=2 \\ 0=0 \end{cases} \tag{41}$$

Аналогично рассматривая метод связанных клеток 2, можно определить, что метод однозначного определения значений в соседних клетках является частным случаем метода связанных клеток 2.

Таким образом, описанные методы можно представить в виде схемы (рисунок 30). Стрелочки в данном случае обозначают, что метод, в который стрелочка входит, является частным случаем метода, из которого стрелочка выходит.

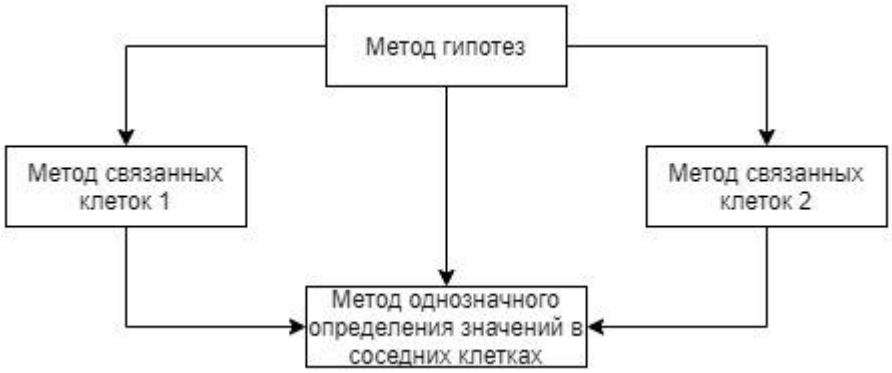

Рисунок 30 — Схема методов решения задачи

2.6 Методы повышения эффективности решения

Нам удалось определить ряд методов, с помощью которых можно найти решения игры «Сапёр». Однако, не для каждого поля можно найти решение с использованием одного конкретного метода. Также поиск решения лишь одним из представленных методов может быть менее эффективным, чем при грамотной комбинации данных методов. Исходя из этого, возникает задача комбинирования описанных методов для решения поля таким образом, чтобы повысить эффективность решения задачи.

2.6.1 Очерёдность применения методов

Одним из способов определения эффективности того или иного метода является среднее время одного цикла t_c, а именно, среднее временя поиска уравнения (или нескольких уравнений) системы 19, с помощью которых возможно будет однозначно определить одно из значений переменных *x*. Среднее время одного цикла можно вычислить по следующей формуле (формула 42):

$$t_c = \frac{n_u}{n_s} * t_u + t_s, \tag{42}$$

где n_u — среднее количество неудачных результатов работы метода,

n_s — среднее количество удачных результатов работы метода,

t_u — среднее время неудачной проверки метода,

t_s — среднее время удачной проверки метода.

Получается, что, чем меньше значение t_c определённого метода, тем в среднем данный метод работает быстрее. Таким образом, можно определить очерёдность применения методов для поиска решения системы 19, исходя из увеличения среднего времени одного цикла выбранного метода.

2.6.2 Сбор и применение схем

Рассмотрим возможность «запоминания» свойств тех уравнений системы 19, исходя из которых, результат работы того или иного метода оказался успешным. Рассмотрим рисунок 31 (левое поле). Для клетки с координатами (2, 2) будет верным следующее уравнение (формула 43):

$$x_{1,1} + x_{1,2} + x_{1,3} = 3 \tag{43}$$

Исходя из уравнения 43 с помощью метода однозначного определения значений в соседних клетках можно однозначно вычислить значения переменных $x_{1,1} = x_{1,2} = x_{1,3} = 1$ (рисунок 31, правое поле).

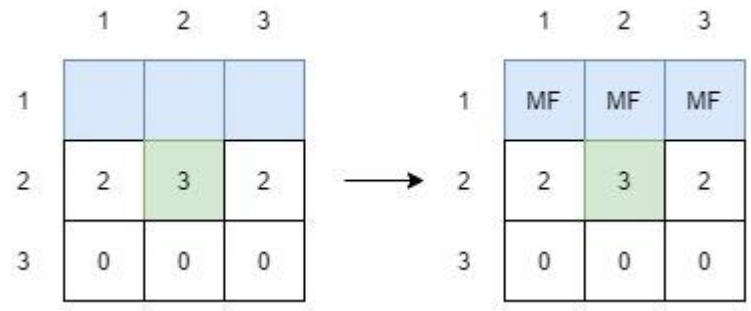

Рисунок 31 — Пример 1

Для того, чтобы в следующий раз можно было бы определить значения в клетках с координатами (1, 1), (1, 2) и (1, 3) без применения метода однозначного определения значений в соседних клетках, используем метод запоминания. Для этого будем сохранять в памяти следующие значения:

- координаты зелёной (фокусной) клетки;
- значение в зелёной (фокусной) клетке;
- для каждой соседней с «зелёной» клеткой клетки определить, закрыта ли она или нет;

- координаты клеток, для которых определяется значение (целевые клетки);
- значения в целевых клетках.

Таким образом, исходя из данных значений, можно восстановить часть значений поля и определить, в каких закрытых клетках восстановленного поля какие значения находятся (рисунок 32).

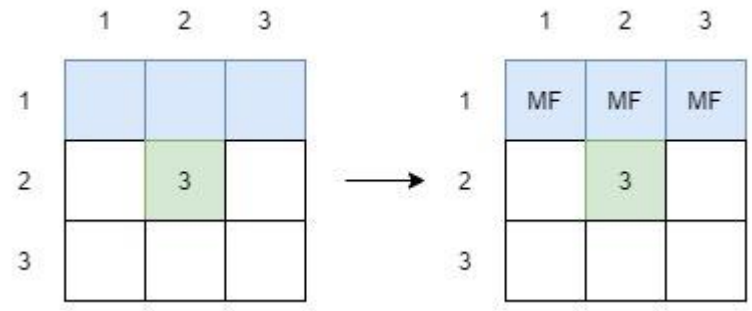

Рисунок 32 — Пример 2

Рассмотрим теперь рисунок 33 (левое поле). Для клеток с координатами (2, 2) и (3, 4) можно определить следующие уравнения (формула 44):

$$\begin{aligned} x_{1,3} + x_{2,3} + x_{3,3} &= 2 \\ x_{2,3} + x_{3,3} + x_{4,3} &= 1 \end{aligned} \tag{44}$$

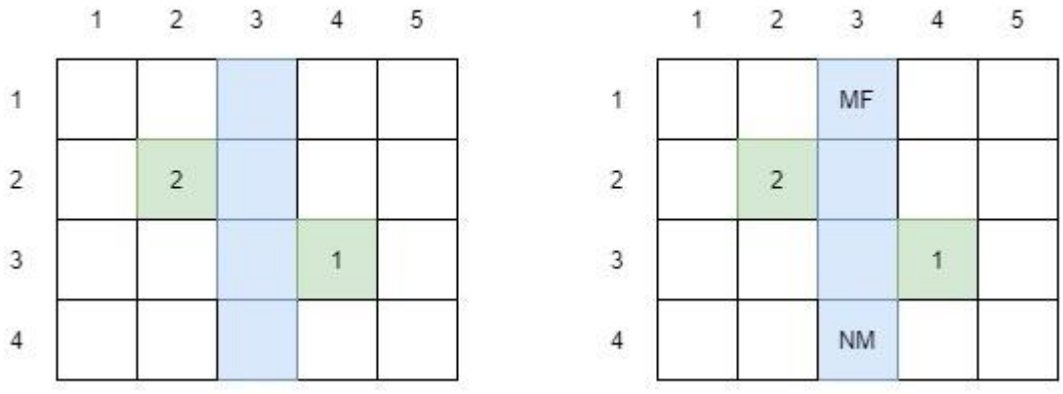

Рисунок 33 — Пример 3

Применим для уравнений формулы 44 метод связанных клеток 1 (формула 45):

$$\begin{gathered}\left(x_{1,3}+x_{2,3}+x_{3,3}\right)-\left(x_{2,3}+x_{3,3}+x_{4,3}\right)=2-1\\ x_{1,3}-x_{4,3}=1 => \ x_{1,3}=1, x_{4,3}=0\end{gathered} \quad (45)$$

Таким образом, удалось определить значения в клетках с координатами (1, 3) и (4, 3) (рисунок 33, правое поле). Теперь необходимо запомнить данную ситуацию для того, чтобы в следующий раз можно было бы восстановить значения в закрытых клетках, не применяя метод связанных клеток 1. Для этого необходимо «исключить» клетки с координатами (1, 4), (1, 5), (4, 1), (4, 2) (рисунок 34, левое поле) и сохранить в памяти перечисленные ранее характеристики, чтобы возможно было бы однозначно определить значения в клетках с координатами (1, 3) и (4, 3) (рисунок 34, правое поле).

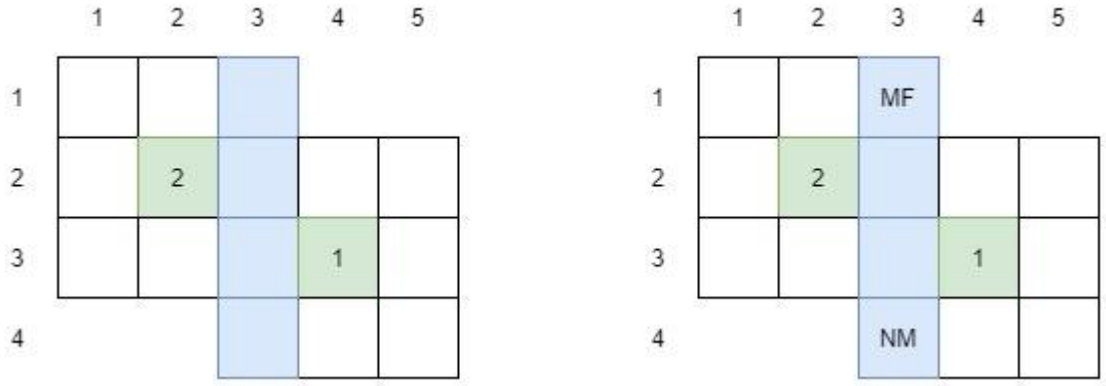

Рисунок 34 — Пример 4

Будем называть сохранение в памяти представленных выше характеристик уравнения сбором схем. Назовём схемой следующий кортеж значений (формула 46):

$$Sc =< h, w, V, CF, VF, CG, VG > \quad (46)$$

Рассмотрим подробнее каждый из элементов кортежа Sc.

Определим матрицу V (value — значение) размером $h*w$, элементы которой хранят состояния сохраняемых в памяти клеток. Определим множество SV (set of values — множество значений) (формула 47):

$$SV = \{O, C, N/A\}, \quad (47)$$

где O — открыта (open);

C — закрыта (close);

N/A — не пригодна (not applicable) (клетка, состояние которой не учитывается при проверке).

Определим кортеж *CF* (coordinates of focus — координаты фокуса), который содержит координаты фокусных клеток.

Определим кортеж *VF* (value of focus — значение фокуса), который содержит значения в фокусных (зелёных) клетках. Значения элементов кортежа VF принадлежат подмножеству множества VOC (формула 48):

$$VF_i \in \{0, 1, 2, 3, 4, 5, 6, 7, 8\} \subset VOC \quad (48)$$

Определим кортеж *CG* (coordinates of goal — координаты цели (целевых клеток)), который содержит координаты целевых клеток.

Определим кортеж *VG* (values of goal — значения цели (целевых клеток)), который содержит значения в целевых клетках. Определим множество SVG (set values of goal — множество значений цели (целевых клеток)) (формула 49):

$$SVG = \{MF, NM\}, \quad (49)$$

где MF (mine flag) — флаг мины,

NM (not mine) — отсутствие мины.

Таким образом, при «наложении» схем на поле «Сапёра» можно определять значения в закрытых клетках поля.

2.7 Описание программной реализации

В качестве реализации представленных методов разработана программа на языке программирования Python с использованием дистрибутива Anaconda [20], а также библиотек NumPy и Matplotlib. В качестве базы данных для хранения информации была выбрана объектно-ориентированная база данных ZODB [21]. Также при разработке использовались следующие модули:

- csv;
- os;
- copy;
- collections;

- math;
- time.

Код программы расположен на репозитории GitHub [22].

Общая блок-схема алгоритма работы программы представлена на рисунке 35.

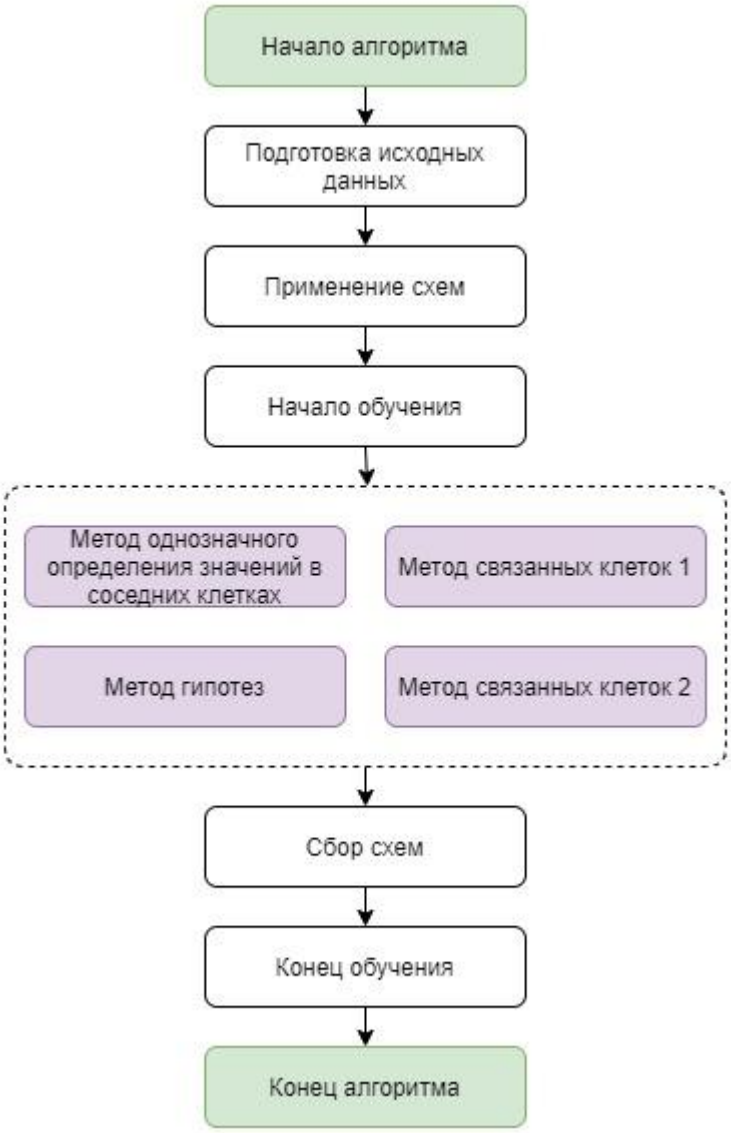

Рисунок 35 — Общая блок-схема алгоритма работы программы

2.7.1 Исходные данные

В качестве исходных данных были выбраны данные «Сапёра», представленные в приложении «Сапёр Go» [23]. В данном приложении присутствует «Кампания», содержащая поля трёх уровней сложности, имеющие детерминированные решения:

- новобранец (25 полей);

- любитель (50 полей);
- ветеран (1000 полей).

Данные о полях с небольшими изменениями в структуре использованы в качестве входных данных в программе. Таким образом, каталог с исходными данными можно представить в виде следующего дерева (рисунок 36):

Рисунок 36 — Дерево каталогов с исходными данными

В файлах с именем fields_info.csv хранится основная информация о поле, а именно:

- высота поля, l;
- ширина поля, w;
- количество мин на поле, tm.

В файлах с именем status.csv хранятся статусы клеток поля. Таким образом, исходя из данных в файле можно однозначно определить значение двумерного кортежа S.

В файлах с именем values.csv хранятся значения открытых клеток. Таким образом, исходя из данных в файле можно однозначно определить значение двумерного кортежа VOC.

При запуске программы происходит создание списка экземпляров класса Minesweeper (которые хранят данные о поле) и запись данных в переменные данного класса. Определим основные переменные класса Minesweeper:

- *l* — высота поля;
- *w* — ширина поля;
- *tm* — общее количество мин на поле;
- *VOC* — список списков значений открытых клеток поля;
- *S* — список списков статусов клеток поля.

Поскольку считывание данных из файлов при запуске программы занимает время и оперативную память, то будем хранить список экземпляров класса Minesweeper в файлах с помощью базы данных ZODB.

2.7.2 Описание классов и методов программы

В ходе разработки программы создано 3 класса:

- Menu (файл *menu.py*);
- Minesweeper (файл *minesweeper.py*);
- Scheme (файл *scheme.py*).

Основным классом является класс **Menu**. С его помощью происходит загрузка и инициализация входных данных, а также схем, если они были ранее сохранены. Также в данном классе реализован запуск работы программы (поиск решения для всех входных данных).

Рассмотрим методы данного класса.

- **draw_graphics**() — отрисовка графиков по окончании работы программы;
- **get_filenames**() — поиск файлов с исходными данными о полях «Сапёра» (fields_info.csv, status.csv, values.csv) в заданной директории и сохранение полных путей к данным файлам в соответствующие списки (таблица 4);

Таблица 4 — Характеристики метода get_filenames

Возвращаемые значения:	**fields_info_count**: числовое значение количество обработанных «fields_info.csv» **status_count**: числовое значение количество обработанных «status.csv» **values_count**: числовое значение количество обработанных «values.csv»

- **learning**(*learning_type=WITH_SAVING_AND_APPLYING_SCHEMAS*) — поиск решения на основе данных о загруженных полях (таблица 5);
- **load_fields**(*check_data=False*) — загрузка данных о полях на основе списков fields_info, fields_status, fields_values (таблица 6);
- **load_schemes**() — загрузка схем из файлов базы данных ZODB.

Класс **Minesweeper** предназначен для поиска решения на основе данных о поле. В данном классе реализованы методы решения задачи и методы повышения эффективности.

Рассмотрим методы данного класса.

- **add_scheme**(*scheme*) — сохранение схемы в список схем (таблица 7);

Таблица 5 — Характеристики метода learning

Параметры:	**learning_type**: строковое значение определяет тип поиска решения *WITH_SAVING_AND_APPLYING_SCHEMAS*: поиск решения с сохранением и применением схем *WITHOUT_SAVING_AND_APPLYING_SCHEMAS*: поиск решения без сохранения и применения схем *ONLY_WITH_APPLYING_SCHEMAS*: поиск решения только с применением уже имеющихся схем

Таблица 6 — Характеристики метода load_fields

Параметры:	**check_data=False**: значение типа bool проверка исходных данных на корректность
Возвращаемые значения:	**result**: значение типа bool результат загрузки данных о полях

Таблица 7 — Характеристики метода add_scheme

Параметры:	**scheme**: экземпляр класса Scheme добавляемая схема

- **check_result**() — проверка корректности найденного решения для заданного поля (таблица 8);

Таблица 8 — Характеристики метода check_result

Параметры:	**result**: значение типа bool корректность найденного решения

- **create_system**() — создание системы уравнений/равенств и базовая обработка системы;
- **full_search_method1**() — полный перебор уравнений/равенств системы для метода однозначного определения значений в соседних клетках;
- **full_search_method2**() — полный перебор уравнений/равенств системы для метода связанных клеток 1;

– **full_search_method3**() — полный перебор уравнений/равенств системы для метода связанных клеток 2;

– **full_search_method4**() — полный перебор уравнений/равенств системы для метода гипотез;

– **full_search_schemes**() — полный перебор сохранённых схем для дальнейшего их применения;

– **get_copy**(*my_minesweeper*) — получение копии экземпляра класса Minesweeper (таблица 9);

Таблица 9 — Характеристики метода get_copy

Параметры:	**my_minesweeper**: экземпляр класса Minesweeper
Возвращаемое значение:	**copy_minesweeper**: экземпляр класса Minesweeper копия экземпляра my_mineswepeer класса Minesweeper

– **get_near_close**(*coords*) — получение списка координат соседних с заданной клеткой закрытых клеток (таблица 10);

Таблица 10 — Характеристики метода get_near_close

Параметры:	**coords**: список числовых значений координаты исходной клетки
Возвращаемое значение:	**cords_list**: список списков числовых значений список координат соседних клеток

– **get_near_coords**(*coords*) — получение списка координат соседних с заданной клеткой клеток (таблица 11);

Таблица 11 — Характеристики метода get_near_coords

Параметры:	**coords**: список числовых значений координаты исходной клетки
Возвращаемое значение:	**cords_list**: список списков числовых значений список координат соседних клеток

– **learning**(*collect_schemes=True*, *use_schemes=True*, *use_methods=True*) — поиск решения для выбранного поля (таблица 12);

Таблица 12 — Характеристики метода learning

Параметры:	**collect_schemes=True**: значение типа bool сохранение схем при поиске решения **use_schemes=True**: значение типа bool использование схем при поиске решения **use_methods=True**: значение типа bool использование методов поиска решения

– **method1**(*equal*, *caution=False*, *correct_check=False*, *collect_schemes=True*) **—** реализация метода однозначного определения значений в соседних клетках решения системы уравнений/равенств (таблица 13);

Таблица 13 — Характеристики метода method1

Параметры:	**equal**: числовое значение номер входного уравнения **caution=False**: значение типа bool возможность «осторожного» применения метода **correct_check=False**: значение типа bool проверка уравнения на корректность **collect_schemes=True**: значение типа bool сохранение схем при успешной проверке

– **method2**(*equal1*, *equal2*, *collect_schemes=True*) **—** реализация метода связанных клеток 1 решения системы уравнений/равенств (таблица 14);

– **method3**(*equals*, *collect_schemes=False*) **—** реализация метода связанных клеток 2 решения системы уравнений/равенств (таблица 15);

– **method4**(*equal*, *collect_schemes=True*) — реализация метода гипотез решения системы уравнений/равенств (таблица 16);

– **prepare_data**() — подготовка и обработка исходных данных о поле;

– **set_mine**(*coords*, *caution=False*) — установка значения — в клетке отсутствует мина (таблица 17);

– **set_no_mine**(*coords*) — установка значения — в клетке находится мина (таблица 18);

Таблица 14 — Характеристики метода method2

Параметры:	**equal1**: числовое значение номер первого входного уравнения **equal2**: числовое значение номер второго входного уравнения **collect_schemes=True**: значение типа bool сохранение схем при успешной проверке

Таблица 15 — Характеристики метода method3

Параметры:	**equals**: список числовых значений список номеров входных уравнений **collect_schemes=True**: значение типа bool сохранение схем при успешной проверке

Таблица 16 — Характеристики метода method4

Параметры:	**equal**: числовое значение номер входного уравнения **collect_schemes=True**: значение типа bool сохранение схем при успешной проверке

Таблица 17 — Характеристики метода set_mine

Параметры:	**coords**: список числовых значений координаты исходной клетки **caution=False**: значение типа bool возможность «осторожной» установки значения

Таблица 18 — Характеристики метода set_no_mine

Параметры:	**coords**: список числовых значений координаты исходной клетки

- **set_open_cell**(*coords*) — открытие клетки с заданными координатами (таблица 19);

Таблица 19 — Характеристики метода set_open_cell

Параметры:	**coords**: список числовых значений координаты исходной клетки

- **use_scheme**(*scheme*) — использование схемы для вычисления значений в закрытых клетках (таблица 20);

Таблица 20 — Характеристики метода use_scheme

Параметры:	**scheme**: экземпляр класса Scheme схема для применения

Класс **Scheme** необходим для хранения данных о схемах с возможностью их дальнейшего применения. В данном классе отсутствуют методы.

Запуск работы программы осуществляется с помощью запуска скрипта с именем *main.py*.

2.8 Результаты работы программы

2.8.1 Полученные значения

2.8.2 Выводы по результатам работы программы

ВЫВОДЫ

В данной работе произведён поиск и анализ как различных классов логических задач, так и частных логических задач, которые можно было бы использовать в качестве примера для разработки системы с элементами самообучения. Поскольку ни один из рассмотренных классов задач не удовлетворял представленным критериям, был определён свой класс задач: каузально-логические игры. В качестве примера игры для разработки системы выбрана игра «Сапёр»/«Minesweeper».

Для выбранной игры рассмотрены методы поиска решения, в том числе и собственный метод, который был разработан на первом этапе данной работы. Каждый из представленных методов имеет свои достоинства и недостатки, однако, ни один из них не подходил для разработки алгоритма с элементами самообучения. В связи с этим было принято решение о разработке новых методов решения.

Кроме того, были рассмотрены процессы обучения человека и машины, определены сходства и отличия, достоинства и недостатки, которые в дальнейшем использовались для разработки системы с элементами самообучения.

Для выбранной каузально-логической игры «Сапёр» представлены правила, описана содержательная и математическая постановки задачи исследования. Получилось, что решение задачи сводится к поиску решения системы уравнений. Для решения системы разработаны и продемонстрированы четыре метода:

- метод однозначного вычисления в соседних клетках;
- метод связанных клеток 1;
- метод связанных клеток 2;
- метод гипотез.

В качестве метода повышения эффективности системы определена формула для определения очерёдности применения методов, а в качестве самообучающегося элемента использовались схемы, основанные на сохранении (запоминании) данных, необходимых для решения задачи с дальнейшим их применением.

Имеется публикация в сборнике тезисов «77-е Дни науки НИТУ МИСиС» [24].

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1 Виды головоломок. Саморазвитие 2.0. URL: http://pruslin.ru/vidy-golovolomok/ (дата обращения: 26.12.21).

2 Минскин Е.М. Занимательные задачи и головоломки для больших и маленьких. — В кн.: Всегда всем весело. М., 1969.

3 В. В. Круглов, М. И. Дли, Р. Ю. Голунов. Нечёткая логика и искусственные нейронные сети. Изд. Физматлит, 2001.

4 П. Джексон. Введение в экспертные системы. 3-е изд. М.: Вильямс, 2001.

5 Большая Советская Энциклопедия. — 1954. — Т. 30., 406.

6 М. А. Данилова, М. Н. Скаткина. Дидактика средней школы. М.: Просвещение, 1975, с. 5.

7 Г. Нойнер, Ю. К. Бабанский. Педагогика. М.: Педагогика, 1984, с. 109.

8 Советский Энциклопедический Словарь. М.: Сов. энциклопедия, 1984, с. 908.

9 А. А. Ивин. Философия: Энциклопедический словарь. М.: Гардарики, 2004.

10 Педагогическая энциклопедия. М., 1968, с. 362.

11 Бим-Бад Б. М. Педагогический энциклопедический словарь. М.: Большая Российская энциклопедия, 2002, с. 156–157.

12 К. В. Махотило. Разработка методик эволюционного синтеза нейросетевых компонентов систем управления. Диссертация на соискание ученой степени кандидата технических наук. Харьков, ХГПУ, 1998.

13 Бихевиоризм: Дж. Уотсон, Э. Торндайк, Б. Скиннер, Э. Толмен. Психология — Лекции, советы, материалы для студентов. URL: https://impsi.ru/general-psychology/biheviorizm-dzh-uotson-e-torndajk-b-skinner-e-tolmen/ (дата обращения: 25.04.22).

14 А. Гадаев. И. Павлов и Д. Уотсон создатели классического обусловливания. «Избачитальня» — литературный портал, 2013. URL: https://www.chitalnya.ru/work/883280/ (дата обращения: 25.04.22).

15 Ян Прашко, Петр Можны, Милош Шлепецки и коллектив. Когнитивно-бихевиоральная терапия психических расстройств. Институт общегуманитарных исследований, М., 2015. С. 32.

16 T. Fawcett. An introduction to ROC analysis. Institute for the Study of Learning and Expertise. USA, 2005.

17 Е. Ю. Корлякова, М. О. Корлякова. Подход к разработке самообучающегося алгоритма игры в «Сапёр». Наукоёмкие технологии в приборо- и машиностроении и развитие

инновационной деятельности в вузе. Материалы Всероссийской научно-технической конференции. Том 2. Изд. КФ МГТУ им. Баумана, Калуга, 2016. С. 23-24.

18 Е. Ю. Корлякова. Подход к разработке самообучающегося алгоритма игры в «Сапёр». Презентация к докладу. Калужский филиал МГТУ им. Баумана, Калуга, 2016.

19 Комаров А. Д. Осторожно, мины! Алгоритм решения игры Сапёр. Компьютерные инструменты в образовании. №5, 2006.

20 Anaconda. Дистрибутив программирования Python. URL: https://www.anaconda.com (дата обращения: 17.04.22).

21 ZODB. Объектно-ориентированная база данных для Python-объектов URL: https://zodb.org/en/latest/ (дата обращения: 17.04.22).

22 GitHub. Репозиторий с исходным кодом программы и сопроводительной документацией. URL: https://github.com/xex238/Diploma_2022 (дата обращения: 24.05.22).

23 Google Play. Сапер GO — классическая игра. URL: https://play.google.com/store/apps/details?id=com.EvolveGames.MinesweeperGo (дата обращения: 17.04.22).

24 77-е Дни науки НИТУ МИСиС. Сборник тезисов.: Издательский дом МИСиС, М., 2022.

ПРИЛОЖЕНИЕ А. Тезаурус

– Поле — это основной элемент игры «Сапёра». Является кортежом F и состоит из следующих элементов:

$$F =< h, w, tm, S, VOC, VCC, VC, MC, k >.$$

– Клетка — это кортеж значений элементов матриц S, VOC, VCC, VC, MC с координатами (i, j), принадлежащих кортежу F:

$$C =< s, voc, vcc, vc, mc, i, j >,$$

$$s_{i,j} = S_{i,j}, voc_{i,j} = VOC_{i,j}, vcc_{i,j} = VCC_{i,j}, vc_{i,j} = VC_{i,j}, mc_{i,j} = MC_{i,j}, S, VOC, VCC, VC, MC \in F$$

– Матрица S (status — статус) — это матрица размером $h*w$ $(h, w, S \in F)$, элементы которой хранят состояния клеток поля. Элементы матрицы S принадлежат множеству SS.

– Множество SS (set of status — множество статусов) — множество допустимых статусов клетки:

$$SS = \{O, C\},$$

где *O* (open) — клетка открыта,

C (close) — клетка закрыта.

– Матрица VOC (values in open cells — значения в открытых клетках) — это матрица размером $h*w$ $(h, w, VOC \in F)$, элементы которой хранят значения открытых клеток поля. Элементы матрицы VOC принадлежат множеству SVOC.

– Множество SVOC (set of values in open cells — множество значений в открытых клетках) — множество допустимых значений открытых клеток поля.

$$SVOC = \{0, 1, 2, 3, 4, 5, 6, 7, 8, M\},$$

где *M* (mine) — мина.

– Матрица VCC (values in close cells — значения в закрытых клетках) — это матрица размером $h*w$ $(h, w, VOC \in F)$, элементы которой хранят значения закрытых клеток поля. Элементы матрицы VCC принадлежат множеству SVCC.

– Множество SVCC (set of values in close cells — множество значений в открытых клетках) — множество допустимых значений в закрытых клетках поля.

$$SVCC = \{E, MF, Q\},$$

где *E* (emptiness) — клетка без значения,

MF (mine flag) — флаг мины,

Q (question) — вопрос.

– Матрица VC (values of cells — значения клеток) — это матрица размером $h*w$ $(h, w, VC \in F)$, элементы которой хранят значения клеток поля, которые отображаются пользователю. Элементы матрицы VC принадлежат множеству $SVOC \cup SVCC$.

– Матрица MC (mines in cells — мины в клетках) — это матрица размером $h*w$ $(h, w, MC \in F)$, элементы которой хранят значения, которые определяют, находится в заданной клетке поля мина или нет. Элементы матрицы MC принадлежат множеству нулей и единиц $\{0, 1\}$.

– Открытая клетка (поля) — это такая клетка C кортежа F, для элемента s которой верно:

$$s = O$$

– Закрытая клетка (поля) — это такая клетка C кортежа F, для элемента s которой верно:

$$s = C$$

– Соседняя клетка (поля) — для заданной клетки C_1 кортежа F с координатами (i_1, j_1) это такая клетка C_2 кортежа F с координатами (i_2, j_2), для которых будет верно:

$$i_1 - 1 \leq i_2 \leq i_1 + 1, j_1 - 1 \leq j_2 \leq j_1 + 1, (i_1, j_1) \neq (i_2, j_2)$$

– Матрица k — это матрица размером $h*w$ $(h, w, k \in F)$, элементами которой также являются матрицы размером $h*w$. Элементы k_{i_1, j_1, i_2, j_2} матрицы каждой из элементов матрицы k определяют, является ли клетка C_1 кортежа F с координатами (i_1, j_1) соседней с клеткой C_2 кортежа F с координатами (i_2, j_2).

– Схема — это кортеж *Sc*, предназначенный для сохранения состояний части элементов клеток поля F для того, чтобы в дальнейшем однозначно определять значения в закрытых клетках поля. Кортеж *Sc* состоит из следующих элементов:

$$Sc = < h, w, V, CF, VF, CG, VG >$$

ПРИЛОЖЕНИЕ Б. Правила игры «Minesweeper»/«Сапёр»

Игра имеет три уровня сложности: Новичок, Любитель и Профессионал. Уровни различаются размером игрового поля, а также общим количеством мин на поле.

На уровне Новичок размер поля равен 8x8, однако в операционной системе Windows XP он равен 9x9.

На уровне Любитель размер поля равен 16x16; на уровне Профессионал размер поля равен 30x16.

Суть игры заключается в том, что некоторые клетки поля свободны, а некоторые заминированы. Каждая клетка поля либо свободна, либо заминирована; третьего не дано.

Перед началом игры точно известно, сколько всего мин будет установлено на поле. По умолчанию, на уровне Новичок устанавливается 10 мин; на уровне Любитель — 40 мин; на уровне Профессионал — 99 мин. Это число можно изменить. Количество необнаруженных мин указывается в левом верхнем углу.

Однако при этом неизвестно, какие именно клетки заминированы. До того, как игрок не откроет клетку или не закончится игра, неизвестно, какие именно клетки свободны, а какие заминированы.

Если игрок считает, что некая клетка свободна, он может нажать её левой кнопкой мыши. При этом клетка откроется. Возможны три варианта:

1) Если открываемая клетка была заминирована, то игра показывает это. В той клетке, которая была открыта, появляется мина, залитая красным. Также при этом открываются все остальные клетки с минами: появляется рисунок мины, но уже без красной заливки. Игра немедленно заканчивается, игрок умер (проиграл).

2) Если открываемая клетка была свободна, но хотя бы одна из соседних клеток содержит мину, то в открываемой клетке появляется цифра, указывающая общее количество мин в соседних клетках. 1 — одна соседняя клетка заминирована; 2 — две соседние клетки заминированы, и так далее. Соседними в Сапёре считаются клетки, которые имеют либо общую сторону (соприкасаются по горизонтали или по вертикали), либо имеют общий угол (соприкасаются в углу по диагонали). Очевидно, что в клетке может появляться одна из цифр от 1 до 8. Каждая цифра имеет свой цвет: 1 — голубой, 2 — зелёный, 3 — красный, 4 — синий, 5 — коричневый, и так далее. Игра продолжается.

3) Если открываемая клетка была свободна и все соседние клетки тоже свободны, то все соседние клетки, как и открываемая клетка, автоматически разминируются и пометятся как свободные (окрасятся в серый цвет). Если какая-либо из разминированных клеток также будет иметь только свободных соседей, то процесс автоматического разминирования на этом

же ходу продолжится далее. И так далее, пока не обнаружатся клетки, имеющие заминированных соседей, и не появятся цифры. Игра продолжается.

Если игрок считает, что в клетке находится мина, он может нажать её правой кнопкой мыши. При этом клетка пометится как потенциально заминированная: в этой клетке появится своеобразный флажок. Кроме того, количество необнаруженных мин при этом уменьшится на единицу. Подчёркиваю, что флажок означает мнение игрока о наличии мины, но вовсе не означает фактическую мину.

Нельзя открывать (нажимать левой кнопкой) клетки, помеченные флажком. Нельзя помечать флажками разминированные клетки — то есть серые либо помеченные цветными цифрами.

Но если игрок сомневается, то он может нажать на помеченную флажком клетку правой кнопкой мыши вторично. При этом флажок поменяется на вопросительный знак. Вопросительный знак в клетке означает, что клетка имеет неопределённый статус и заслуживает внимания. Эту клетку можно оставить на время и вернуться к ней позже.

Можно нажать на клетку со знаком вопроса правой кнопкой мыши в третий раз. При этом знак вопроса исчезнет, и клетка вернётся к своему изначальному состоянию — неопределённое состояние, отсутствие изображения.

Игра продолжается до тех пор, пока каждая клетка поля не будет иметь один и только один из следующих двух статусов:

1) разминирована (либо серая, либо содержит цветную цифру);
2) правильно помечена флажком как содержащая мину.

Если игрок пройдёт всё поле — он выиграл.

Очевидно, что часто игрок в любой момент может наткнуться на мину, это означает, что он проиграл. При этом часто бывает, что одна или несколько клеток помечены игроком неверно. В этом случае игра помечает такие клетки перечёркнутой миной, что означает, что мины там не было, несмотря на то что игрок поставил там флажок.