Здравствуйте, уважаемые члены комиссии.

В жизни каждого человека присутствуют задачи, решение которых можно разбить на отдельные этапы. Одним из примеров является поступление в университет. На каждом этапе, начиная с выбора университета для подачи заявления, заканчивая тем, в какой университет отнести оригинал документа об образовании, человек принимает решения и, в зависимости от принимаемого решения человек получает обратную связь, то есть новую для себя информацию. Однако, на каждом этапе принятия решений существуют риски нежелательных или негативных последствий. В зависимости от типа принимаемого решения эти риски могут быть как приемлемыми, так и неприемлемыми. Так или иначе, возникает задача принятия решений с минимальными рисками.

С точки зрения искусственного интеллекта подобные решения способны принимать такие системы, как системы, основанные на нейронных сетях или экспертные системы. Однако, у данных систем есть критические недостатки, которые не позволяют им находить широкое применение в повседневной жизни. Так, обучать нейронные сети для решения широкого круга задач, связанных с высокими рисками, крайне проблематично, поскольку всего одно неправильно принятое решение может принести неприемлемые последствия. Это, пожалуй, основная причина, почему беспилотные автомобили пока не используются в массовом применении. Для экспертных же систем крайне проблематично обновлять базу знаний. Таким образом, возникает потребность в разработке систем, способных принимать эффективное решение в любой потенциальной ситуации заданного круга задач.

Исходя из этого, целью данной работы является разработка комплекса алгоритмов с элементами самообучения для решения каузально-логических игр, которые (важно подчеркнуть) представляют из себя модели пошаговых задач, встречающихся в реальной жизни. Каузально-логические игры — это самостоятельно определённый класс логических игр среднего или низкого уровня сложности, среднего или высокого уровня формализуемости и имеющие однозначное (детерминированное) решение. В качестве примера такой задачи была выбрана игра «Сапёр».

Входные данные представлены на слайде. Основной задачей является разработка таких способов решения, содержащих элементы самообучения, которые в дальнейшем можно применять не только для поиска решения выбранной игры, а для всего класса каузально-логических игр.

Исходные данные содержатся в кортеже . Элементами данного кортежа являются 3 кортежа, содержащие данные о полях различного уровня сложности: «Новобранец» (длина кортежа 25), «Новичок» (длина кортежа 50), «Ветеран» (длина кортежа 1000). Элементами данных кортежей является кортеж длины 5. Особое внимание необходимо уделить матрицам S и VOC. На слайде представлено отображение множества значений клеток поля во множество SS, пример матрицы S, отображение множества значений клеток поля во множество SVOC и пример матрицы VOC.

Необходимо найти значения элементов кортежа , который по своей структуре имеет сходство с кортежом . Элементы подкортежей данного кортежа определяют, находится ли в данной клетке поля мина или нет.

Исходя из правил игры определим новые параметры и переменные и установим связь между ними. Для этого определим кортеж TF аналогичный кортежу . Элементами каждого из трёх подкортежей данного кортежа являются кортежи F. С первыми пятью элементами мы уже знакомы, рассмотрим последние 4. На слайде представлено отображение множества значений клеток поля во множество SVCC, пример матрицы VCC, отображение множества значений клеток поля в объединение множество SVOC и SVCC, пример матрицы VC, отображение множества значений клеток поля во множество нулей и единиц, пример матрицы MC.

На данном слайде представлена математическая запись факта, что число в клетке определяет количество мин с соседних с ней клетках. Сначала задаётся система равенств, затем с помощью замены части значений на переменные определяется система уравнений.

На данном слайде задаётся равенство для определения общего количества мин на поле, а затем, также с применением замены части значений записывается уравнение. Получается, что решение игры «Сапёр» сводится к поиску решения системы уравнений.

Свойства системы уравнений представлены на слайде. Особо необходимо выделить свойство детерминированности, то есть наличие единственного решения системы уравнений.

Перейдём к описанию процедур для поиска решения. Целью первой процедуры является поиск таких открытых клеток поля, для которых можно однозначно вычислить значение в соседних закрытых клетках. Описание процедуры приведено на слайде.

Целью второй процедуры является поиск таких открытых клеток поля, для которых имеются одинаковые соседние закрытые клетки. Описание процедуры приведено на слайде.

Целью третьей процедуры является вычисление таких закрытых клеток, в которых отсутствуют мины на основе информации об общем количестве мин на поле и значений в открытых клетках. Описание процедуры приведено на слайде.

Целью четвёртой процедуры является выдвижение гипотез о наличии или отсутствие мины в некоторой закрытой клетке поля и дальнейшая проверка данных гипотез на основе имеющихся данных с применением процедуры 1.

Дадим теперь определение этапа для каждой процедуры, осуществляющий поиск решения.

Объединим используемые процедуры для поиска решения и определим критерий останова. Это произойдёт при условии нахождения решения или при неуспехе на каждом из четырёх этапов применения процедур.

Будем определять очерёдность применения процедур, исходя из увеличения значения среднего времени одного цикла процедуры, который для каждого процедуры рассчитывается по представленной формуле.

Рассмотрим теперь самообучающийся элемент, основанный на сборе и применении схем. Будем называть схемой найденное решение с помощью одной из процедур, которое можно запомнить, а затем применить полученное знание в аналогичной ситуации.

На основе представленных способов поиска решения разработана программа на языке программирования Python. На данном слайде представлена общая блок-схема алгоритма.

На данном слайде представлена блок-схема алгоритма поиска решения.

Будем искать решения тремя способами, представленными на слайде. Но, для начала, вычислим среднее время цикла для каждой процедуры и определим очерёдность их применения, исходя из увеличения среднего времени одного цикла. Графики зависимости время поиска решения тем или иным способом от порядкового номера полей представлены на слайде. Можно заметить, что применение процедур для поиска решения примерно для первых 900 полей эффективнее, чем применение схем. Далее, поиск решения осуществляется быстрее при применении схем.

В результате работы программы было найдено решение для 95,9% полей. Также, исходя из нормированной диаграммы можно определить, что для решения полей с уровнем сложности «Новобранец» достаточно процедуры 1. Для уровня сложности «Новичок» доминирует процедура 1, но для решения также применяется процедура 2. А для уровня сложности «Ветеран» доминирует процедура 4.

Основным выводом данной работы является разработанная система, которая, хоть и с ограничениями, но может обучаться. Исходя из определения обучения, результатом процесса обучения являются приобретённые знания, умения и навыки. Если рассматривать знания как информацию о некоторой задаче, умения как возможность применять полученные знания для решения задачи, а навыки, как возможность быстро и эффективно применять знания для решения задачи, то, можно сказать, что разработанная система в процессе обучения приобретает знания, умения применять данные знания запрограммировано, но навыки у системы отсутствуют.

В ходе данной работы успешно достигнуты поставленные цели. Имеется публикация в сборнике тезисов 77 Дней науки НИТУ МИСиС.

Рассмотрим теперь самообучающийся элемент — сбор и применение схем. Будем называть схемой найденное решение с помощью одной из процедур, которое можно запомнить, а затем применить полученное знание в аналогичной ситуации. Исходя из этого, необходимо запомнить данные, представленные на слайде. Для метода однозначного применения значений и метода связанных клеток 2 схемы будут выглядеть так, а для метода связанных клеток 1 и метода гипотез так.

Математически схему можно определить как кортеж из семи элементов. Напомню, что фокусные клетки — это открытые клетки, для которых необходимо сохранять значения в соседних клетках. В данном случае это клетки с координатами (2, 1) и (2, 2), а целевые клетки — это закрытые клетки, для которых определяется значение (есть ли в данной клетке мина или нет). В данном случае это клетка с координатами (3, 3).