

Московский Государственный Университет
имени М.В. Ломоносова
Российская Академия Наук
Международная Академия Технологических Наук
Российская Академия Естественных Наук

Интеллектуальные Системы.

Теория и приложения

ТОМ 28 ВЫПУСК 1 * 2024

МОСКВА

УДК 519.95; 007:159.955
ББК 32.81

ISSN 2411-4448
Издаётся с 1996 г.

Главный редактор: д.ф.-м.н., профессор Э.Э.Гасанов

Редакционная коллегия:

к.ф.-м.н., с.н.с. А.В. Галатенко (зам. главного редактора)
д.ф.-м.н., проф. А.А. Часовских (зам. главного редактора)

д.ф.-м.н., проф. В.В. Александров, д.ф.-м.н., проф. С.В. Алешин, д.ф.-м.н., проф. А.Е. Андреев, д.ф.-м.н., проф. Д.Н. Бабин, проф. К. Вашик, проф. Я. Деметрович, академик РАН, д.ф.-м.н., проф. Ю.Л.Ершов, проф. Г. Килибарда, д.ф.-м.н., проф. В.Н. Козлов, д.ф.-м.н., проф. А.В. Михалев, к.ф.-м.н., в.н.с. В.А. Носов, д.ф.-м.н., проф. А.С. Подколзин, д.ф.-м.н., проф. Ю.П. Пытьев, д.т.н., проф. А.П. Рыжов, академик РАН, д.т.н., проф. А.С. Сигов, к.ф.-м.н., доц. А.С. Строгалов, проф. Б. Тальхайм, проф. Ш. Ушчумлич, д.ф.-м.н., проф. А.В. Чечкин, к.ф.-м.н. Р. Шчепанович.

Секретари редакции: И.О. Бергер, Е.В. Кузнецова

В журнале «Интеллектуальные системы. Теория и приложения» публикуются научные достижения в области теории и приложений интеллектуальных систем, новых информационных технологий и компьютерных наук.

Издание журнала осуществляется под эгидой МГУ имени М.В. Ломоносова, Научного Совета по комплексной проблеме «Кибернетика» РАН, Отделения «Математическое моделирование технологических процессов» МАТИ.

Учредитель журнала: ООО «Интеллектуальные системы».

Журнал входит в список изданий, включенных ВАК РФ в реестр публикаций материалов по кандидатским и докторским диссертациям по математике и механике.

Индекс подписки на журнал: 64559 в каталоге НТИ «Роспечать».

Адрес редакции: 119991, Москва, ГСП-1, Ленинские Горы, д. 1, механико-математический факультет, комн. 12-01.

Адрес издателя: 115230, Россия, Москва, Хлебозаводский проезд, д. 7, стр. 9, офис 9. Тел. +7 (495) 939-46-37, e-mail: mail@intsysjournal.org

*) Прежнее название журнала: «Интеллектуальные системы».

© ООО «Интеллектуальные системы», 2024.

ОГЛАВЛЕНИЕ

Часть 1. Общие проблемы теории интеллектуальных систем

Каледина Е.А., И.М.Гуськова Стабилизация управляемой многосвязной системы с использованием генетического алгоритма 5

Козлов В.Н. Электронная цифровая подпись на основе кодов, определяющих изображения с точностью до аффинных преобразований 18

Чечкин А.В. Пять принципов многоагентной автоматизированной системы искусственного интеллекта 31

Часть 2. Специальные вопросы теории интеллектуальных систем

Ковалева Е.С. Алгоритм распознавания эмоций на основе линейной регрессии 49

Фаерштейн И.С. О единственности решения одной системы полиномиальных уравнений над конечным полем 61

Шергин И.А., Рыжов А.П. Проблема оценки позы человека: задачи, методы, решения 69

Часть 3. Математические модели

Масленников Д.О. О числе состояний автомата, модифицирующего свою диаграмму 86

Родин С.Б. О свойстве линейной реализуемости отображений 98

Часть 1.
Общие проблемы теории
интеллектуальных систем

Стабилизация управляемой многосвязной системы с использованием генетического алгоритма

Е. А. Каледина¹, И. М. Гуськова²

В работе рассматривается многосвязная управляемая динамическая система с локальным управлением. На основе генетического алгоритма разработан метод построения локальных управлений, стабилизирующих систему в целом. Приведены используемые параметры генетического алгоритма, а именно операторы отбора, кроссовера, мутации, вид функции приспособленности и критерии основа. Проведено моделирование и анализ полученных результатов.

Ключевые слова: генетические алгоритмы, многосвязная динамическая система, локальное управление, асимптотическая устойчивость, стабилизация.

1. Постановка задачи

Развитие методов искусственного интеллекта в настоящее время позволяет применять их к все более широкому кругу проблем, включая задачи анализа устойчивости динамических систем, их стабилизации и синтеза оптимального управления. Так, в работах [1] – [5] рассмотрены подходы к синтезу функции Ляпунова для нелинейных динамических систем, основанные на использовании нейронных сетей. Также нейронные сети прямого распространения достаточно активно используются для построения управления различных динамических систем [6]–[9].

Отметим, что для обучения нейронных сетей чаще всего используется метод обратного распространения ошибки [10], что накладывает определенные ограничения на их использование. Нейросетевые методы успешно

¹Каледина Елена Александровна — кандидат физико-математических наук, доцент каф. прикладной математики, дифференциальных уравнений и теоретической механики факультета математики и информационных технологий ФГБОУ ВО Мордовский государственный университет им. Н.П. Огарева e-mail: elena.lizina@gmail.com.

Kaledina Elena Alexandrovna — Ph.D. in Physical and Mathematical Sciences, associate professor of the department of applied mathematics, differential equations and theoretical mechanics, Ogarev Mordovia State University of Faculty of Mathematics and Information Technology

²Гуськова Ирина Михайловна — магистрант факультета математики и информационных технологий ФГБОУ ВО Мордовский государственный университет им. Н.П. Огарева e-mail: guskovai2014@yandex.ru.

Guskova Irina Mikhailovna — master's student of Faculty of Mathematics and Information Technology Ogarev Mordovia State University.

используются, прежде всего, в тех задачах, где получены только приближенные методы решения либо время их решения точными методами достаточно велико. Кроме того, для расчета градиента функции ошибки могут использоваться структуры, имитирующие объекты управления. Иными словами, для построения управления необходим «образец» для обучения нейронной сети.

Указанная сложность создает спрос на исследование методов искусственного интеллекта, обучающихся без учителя. К их числу относятся генетические и эволюционные алгоритмы [11], [12], представляющие собой методы случайного поиска оптимальных решений, имитирующие процесс эволюции. Данные алгоритмы применимы к системам большой размерности, задачам со сложным математическим представлением, поддерживают распараллеливание и пригодны к непрерывному обучению. В настоящее время генетические алгоритмы используются для анализа устойчивости динамических нелинейных систем [13], для решения задач многопараметрической стабилизации [14], [15] и синтеза оптимальных систем управления [16].

В данной работе изучается вопрос стабилизации многосвязной линейной динамической системы посредством локальных управлений, основанный на использовании генетического алгоритма.

Рассмотрим многосвязную управляемую линейную систему

$$\frac{dx}{dt} = Ax + Bu, \quad (1)$$

где $x \in R^n$, $B \in R^n$, A – матрица размерности $n \times n$, допускающая неперекрывающуюся декомпозицию, $u \in R$ – скалярное управление. Учитывая декомпозицию, систему (1) можно записать в виде совокупности q систем

$$\dot{x}_s = A_s x_s + \sum_{j=1, j \neq s}^q A_{sj} x_j + B_s u_s, \quad s = \overline{1, q}, \quad (2)$$

где $x_s \in R^{n_s}$, A_s – квадратные матрицы размерности $n_s \times n_s$; A_{sj} – матрицы размерности $n_s \times n_j$, $\sum_{s=1}^q n_s = n$.

Управления в системе (2) имеют вид $u_s = K_s^T x_s$, где коэффициенты усиления $K_s \in R^{n_s}$ стабилизируют линейные подсистемы

$$\dot{x}_s = A_s x_s + B_s u_s, \quad s = \overline{1, q},$$

или, с учетом вида управлений,

$$\dot{x}_s = (A_s + B_s K_s^T) x_s, \quad s = \overline{1, q}. \quad (3)$$

Отметим, что построение управлений для систем вида (2) рассматривалось в работах [17], [18], где приведены дополнительные условия, накладываемые на коэффициенты усиления управлений линейных подсистем. Использование генетического алгоритма упрощает поиск локальных коэффициентов усиления, удовлетворяющих данным условиям среди всего множества возможных управлений, стабилизирующих системы (3). Программная реализация рассматриваемого алгоритма не требует никаких дополнительных критериев, кроме исходной системы (1) и размерности ее подсистем.

2. Генетический алгоритм для стабилизации многосвязной системы

Генетические алгоритмы представляют собой рандомизированный эвристический метод поиска экстремума, основанный на аналогии с принципами эволюционной теории Дарвина.

Данные алгоритмы по аналогии с эволюцией в природе формируют популяцию отдельных особей, представляющих собой потенциальные решения поставленной задачи, которые называются индивидуумами. Каждому индивидууму ставится в соответствие хромосома в виде набора генов. Всем индивидуумам в ходе алгоритма итеративно присваивается оценка, от которой зависит, будет ли потенциальное решение участвовать в создании следующего поколения, так как более предпочтительные индивидуумы обладают большей вероятностью преодолеть отбор и передать свои свойства потомкам. Таким образом, постепенно совершенствуются потенциальные решения задачи.

Для формирования новой популяции индивидуумов используется отбор, в процессе которого выбираются те особи, которые будут участвовать в создании следующего поколения, на основании найденного значения функции приспособленности. Наиболее приспособленные индивидуумы выбираются с большей вероятностью.

После операции отбора выбирается пара родителей, части их хромосом скрещиваются с заранее заданной вероятностью и в популяции образуются новые индивидуумы, то есть происходит процесс скрещивания (кроссовера). Если скрещивание не произошло, то родители переходят в новое поколение без изменений.

Мутация – завершающий этап генетического алгоритма, применяемый при создании нового поколения. Она представляет собой операцию, которая может произойти в хромосоме с заданной очень малой вероятностью. Цель оператора мутации – неожиданным образом изменять ген в хромосоме особи, тем самым обновлять популяцию. Мутация обеспечивает

поиск решений в незатронутых участках пространства. Однако применение оператора мутации к индивидууму может также и ухудшить его приспособленность.

3. Реализация генетического алгоритма

Для реализации алгоритма поиска стабилизирующих управлений системы (2) был использован язык программирования Python и пакет DEAP – универсальный и гибкий каркас эволюционных вычислений для решения практических задач с помощью генетических алгоритмов. На рисунке 1 показана блок-схема работы данного алгоритма.

На первом этапе алгоритма происходит инициализация данных: задается размерность исходной системы (1) и размерность линейных подсистем n_s , $s = \overline{1, q}$. Далее осуществляется проверка управляемости введенной системы [19]. Для этого строится матрица управляемости $[B, AB, \dots, A^{(n-1)}B]$ и вычисляется ее ранг. Если он равен размерности введенной системы, то система управляема, и тогда осуществляется поиск стабилизирующих управлений, иначе система не управляема и на этом алгоритм завершается.

Для поиска локальных управлений линейных подсистем используется генетический алгоритм, в котором индивидуумами являются коэффициенты усиления. Каждый индивидуум представляет собой список вещественных чисел размерностью n_s . На первом шаге генетического алгоритма вся популяция заполняется случайными значениями.

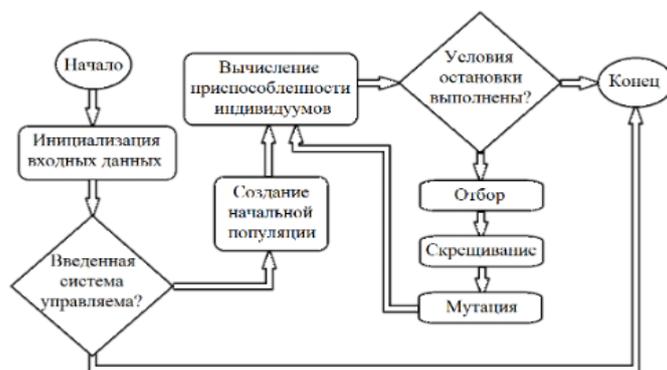


Рис. 1. Блок-схема работы генетического алгоритма

Оценка популяции происходит с помощью функции приспособленности. Задача генетического алгоритма оптимизировать эту функцию. Вид

функции приспособленности для поставленной задачи [20]:

$$\sum_{j=1}^n e^{-\lambda_j} \rightarrow \max, \quad (4)$$

где λ_j , $s = \overline{1, q}$ – собственные числа матриц линейных подсистем (3).

Отметим, что заменив знак перед λ_j на противоположный, задачу поиска максимума функции (4) можно заменить на поиск минимума.

Для того, чтобы генетический алгоритм выполнял свою цель необходимо зарегистрировать генетические операторы отбора, скрещивания и мутации. Оператор отбора используется для формирования нового поколения индивидуумов на основании найденного значения функции приспособленности. В работе был использован метод турнирного отбора с размером турнира равным 3. Данный метод заключается в том, что на каждом шаге отбора из популяции отбираются три индивидуума. Проходит отбор та особь, чье значение функции приспособленности больше остальных. Отметим, что при большем размере турнира вероятность отбора менее приспособленных индивидуумов уменьшается, что может привести к вырождению популяции. В качестве оператора кроссовера использован метод имитации двоичного скрещивания. Основное свойство данного метода – среднее значение генов родителей и потомков соответственно равны. Значения генов потомков вычисляются по формулам

$$offspring_1 = \frac{1}{2}[(1 + \beta)parent_1 + (1 - \beta)parent_2]$$

$$offspring_2 = \frac{1}{2}[(1 - \beta)parent_1 + (1 + \beta)parent_2],$$

где β – случайное число, называемое коэффициентом распределения, $parent_1$ и $parent_2$ – значения генов родителей.

При скрещивании важно сохранять сходство родителей и потомков, для этого β нужно выбирать из случайного распределения. Поэтому индекс распределения вычисляется с помощью случайной равномерно распределенной на отрезке $[0, 1]$ величины u . В таком случае β находится по формулам:

$$\begin{aligned} \text{если } u \leq 0.5 : \beta &= (2u)^{\frac{1}{\eta+1}}; \\ \text{если } u > 0.5 : \beta &= \left(\frac{1}{2}(1 - u)\right)^{\frac{1}{\eta+1}}, \end{aligned}$$

где η – постоянная, называемая индексом распределения. В работе был использован индекс распределения $\eta = 20$.

Также в работе был использован метод гауссовой мутации, которая состоит в замене значения гена на случайное число близкое к исходному,

взятое из нормально распределенной мутации с параметрами $\mu = 0$, $\sigma = 0,2$.

Константы используемого генетического алгоритма:

- вероятность скрещивания 0,9;
- вероятность мутации индивидуума 0,1.

Условиями останова генетического алгоритма является асимптотическая устойчивость линейных подсистем (3) и системы (2) в целом. Первое условие выполняется при выполнении неравенств $Re\lambda_j(A_s + B_s K_s^T) < 0$, $j = \overline{1, n_s}$, $s = \overline{1, q}$. Данное условие проверяется с помощью функции приспособленности (4).

Для проверки того, что исходная система стабилизируется локальными управлениями, использована вектор-функция Ляпунова $V = (V_1, \dots, V_n)$. Так как асимптотическая устойчивость линейных подсистем (3) обеспечена работой генетического алгоритма, элементы вектор-функции Ляпунова представляются в виде квадратичных форм $V_s = x_s^T C_s^T x_s$, где C_s – симметричные положительно определенные матрицы. В работе C_s находятся из уравнений Ляпунова

$$(A_s + B_s K_s^T)^T C_s + C_s (A_s + B_s K_s^T) = -E,$$

где E – единичная матрица соответствующей размерности. Полученные функции Ляпунова будут удовлетворять неравенствам Красовского [21],

$$\lambda_{1s} \|x_s\|^2 \leq V_s(x_s) \leq \lambda_{2s} \|x_s\|^2,$$

$$\left| \frac{\partial V_s(x_s)}{\partial x_s} \right| \leq c_s \|x_s\|,$$

$$\left. \frac{dV_s(x_s)}{dt} \right|_{(3)} = -d_s \|x_s\|^2,$$

где $\lambda_{1s} = \lambda_m(C_s)$, $\lambda_{2s} = \lambda_M(C_s)$, $c_s = \lambda_m(E)$, $\lambda_m(C_s)$. Здесь через λ_m и λ_M обозначены соответственно наименьшее и наибольшее собственные числа матриц C_s . Для проверки асимптотической устойчивости исходной

системы воспользуемся теоремой из работы [18] и составим матрицу

$$\tilde{A} = \begin{pmatrix} -\frac{d_1}{2\lambda_{21}} & \frac{c_1\|A_{12}\|}{\lambda_{11}^{\frac{1}{2}}\lambda_{12}^{\frac{1}{2}}} & \dots & \frac{c_1\|A_{1q}\|}{\lambda_{11}^{\frac{1}{2}}\lambda_{1q}^{\frac{1}{2}}} \\ \frac{c_2\|A_{21}\|}{\lambda_{12}^{\frac{1}{2}}\lambda_{11}^{\frac{1}{2}}} & -\frac{d_2}{2\lambda_{22}} & \dots & \frac{c_2\|A_{2q}\|}{\lambda_{12}^{\frac{1}{2}}\lambda_{1q}^{\frac{1}{2}}} \\ \dots & \dots & \dots & \dots \\ \frac{c_q\|A_{q1}\|}{\lambda_{1q}^{\frac{1}{2}}\lambda_{11}^{\frac{1}{2}}} & \frac{c_q\|A_{q2}\|}{\lambda_{1q}^{\frac{1}{2}}\lambda_{12}^{\frac{1}{2}}} & \dots & -\frac{d_q}{2\lambda_{2q}} \end{pmatrix}$$

Тогда для того, чтобы система (2) была асимптотически устойчивой, необходимо, чтобы действительные части собственных чисел матрицы \tilde{A} были отрицательны.

Таким образом условия останова алгоритма будут иметь вид

$$y = \begin{cases} Re \lambda_j(A_s + B_s K_s^T), & j = \overline{1, n_s}, s = \overline{1, s}, \\ Re \lambda_j(\tilde{A}) < 0, & j = \overline{1, q}. \end{cases}$$

4. Пример

В качестве иллюстративного примера рассмотрим управляемую систему вида

$$\begin{aligned} \dot{x}_1 &= -3,35x_1 + 6,25x_2 - 6,83x_3 + 4,05x_4 + 2,82u_1, \\ \dot{x}_2 &= 3,44x_1 - 6,21x_2 - 7,39x_3 + 1,98x_4 - 1,01u_2, \\ \dot{x}_3 &= 9,79x_1 + 3,89x_2 - 9,44x_3 + 1,56x_4 - 0,39u_3, \\ \dot{x}_4 &= -6,82x_1 + 8,89x_2 - 9,12x_3 + 4,13x_4 - 2,37u_4. \end{aligned} \quad (5)$$

Размерность системы $n = 4$, размерность линейных подсистем $n_1 = 2$, $n_2 = 2$. Данная система управляема, так как ранг матрицы управляемости равен 4. Локальные управления, действующие на линейные подсистемы, имеют вид $u_1 = K_1x_1$, $u_2 = K_2x_2$, $u_3 = K_3x_3$, $u_4 = K_4x_4$ и выбираются таким образом, чтобы стабилизировать соответствующие линейные системы

$$\begin{aligned} \dot{x}_1 &= (-3,35x_1 + 2,82K_1)x_1, \\ \dot{x}_2 &= (-6,21x_2 - 1,01K_2)x_2, \\ \dot{x}_3 &= (-9,44x_3 - 0,39K_3)x_3, \\ \dot{x}_4 &= (4,13x_4 - 2,37K_4)x_4. \end{aligned} \quad (6)$$

Разработанный генетический алгоритм произвел поиск коэффициентов усиления за 501 поколение. Результатом его применения стало

найденной стабилизирующее управление вид $u_1 = -20,0015x_1$, $u_2 = 19,9999x_2$, $u_3 = 17,9253x_3$, $u_4 = 8,5736x_4$.

На рисунках 2–3 показано движение линейных подсистем (6). Очевидно, что найденные управления действительно обеспечивают асимптотическую устойчивость обеих линейных подсистем, так как графики их движения с течением времени сходятся.

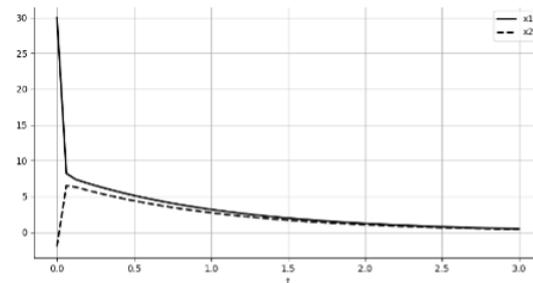


Рис. 2. Стабилизация линейных подсистем (6) относительно переменных x_1, x_2

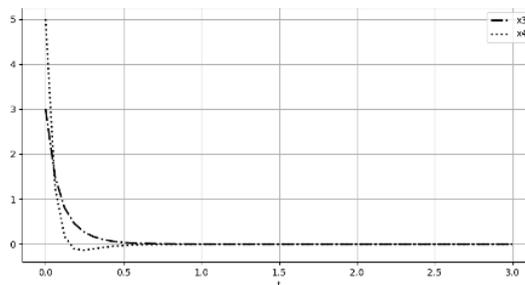


Рис. 3. Стабилизация линейных подсистем (6) относительно переменных x_3, x_4

Для того чтобы удостовериться, что локальные управления гарантируют устойчивость системы в целом также построен график движения системы (5). По рисунку 4 можно окончательно убедиться, что построенные управления стабилизируют систему в целом.

5. Выводы

В работе разработан и программно реализован алгоритм построения локальных управлений, стабилизирующих как подсистемы, так и всю системы в целом. Управляющие воздействия строятся в виде линейной обратной связи, а поиск коэффициентов усиления осуществляется с ис-

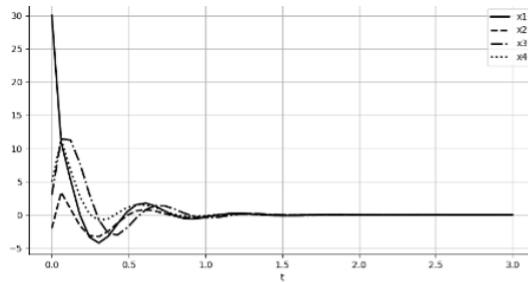


Рис. 4. Процесс стабилизации системы (5) относительно всех переменных

пользованием генетического алгоритма. Предложенный алгоритм отличается общим подходом к решению поставленной задачи, так как, в отличие от других методов машинного обучения, не требует приближенного решения или других примеров для обучения.

Список литературы

- [1] Dai H., Landry B., Yang L., Pavone M., Tedrake R., “Counter-example guided synthesis of neural network Lyapunov functions for piecewise linear systems”, *2020 59th IEEE Conference on Decision and Control (CDC)*, 2020, 1274–1281.
- [2] Dai H., Landry B., Yang L., Pavone M., Tedrake R., *Lyapunov-stable neural-network*, 2021, arXiv: nucl-ex/2109.14152.
- [3] Chen S., Fazlyab M., Morari M., Pappas G.J., Preciado V.M., “Learning Lyapunov Functions for Hybrid Systems”, *2021 55th Annual Conference on Information Sciences and Systems (CISS)*, Baltimore, MD, USA, 2021.
- [4] Verdier C.F., Mazo M., “Formal Synthesis of Analytic Controllers for Sampled-Data Systems via Genetic Programming”, *2018 IEEE Conference on Decision and Control (CDC)*, Miami, FL, USA, 2018, 4896–4901.
- [5] Ben Sassi M. A., Sankaranarayanan S., Chen X., Ábrahám E., “Linear relaxations of polynomial positivity for polynomial Lyapunov function synthesis”, *MA Journal of Mathematical Control and Information*, **33**:3 (2016), 723–756.
- [6] Khapkin D.L., Feofilov S.V., Kozyr A.V., “Study of neural network control stability based on mixed linear integer programming”, *4th In-*

ternational Conference on Control Systems, Mathematical Modeling, Automation and Energy Efficiency (SUMMA), Lipetsk, Russia, 2022, 326–329.

- [7] Хапки́н Д.Л. , Феофи́лов С.В. , Козырь А.В., “Синтез устойчивой нейросетевой замкнутой системы управления существенно нелинейным объектом с оптимизацией времени регулирования”, *Информатика: проблемы, методы, технологии*, XXIII Международная научно-практическая конференции им. Э.К. Алгазинова (Воронеж, 15–17 февраля 2023 г.), Материалы конференции, Воронежский государственный университет, Воронеж, 2023, 405–406.
- [8] Седова Н.О., Токмаков С.В., “Об использовании нейроуправления с запаздывающей обратной связью в задаче стабилизации по выходу”, *Нечеткие системы и мягкие вычисления*, **15:1** (2020), 26–42.
- [9] Zhou R., Quartz T., De Sterck H., Liu J., *Neural Lyapunov Control of Unknown Nonlinear Systems with Stability Guarantees*, arXiv: nucl-ex/2206.01913.
- [10] Хайкин С., *Нейронные сети: полный курс*, «Вильямс», Москва, 2008, 1103 с.
- [11] Вирсански Э., *Генетические алгоритмы на Python*, ДМК Пресс, Москва, 2020, 286 с.
- [12] Емельянов В.В., Курейчик В.В. , Курейчик В.М., *Теория и практика эволюционного моделирования*, «Физматлит», Москва, 2003, 432 с.
- [13] Zenkin A., Peregudin A., Bobtsov, A., *Lyapunov function search method for analysis of nonlinear systems stability using genetic algorithm.*, 2023, arXiv: nucl-ex/2307.03030.
- [14] Беляев А.С., Суменков О.Ю., “Применение генетического алгоритма для синтеза параметров линейно-квадратичного регулятора опорной системы обезвешивания”, *Молодежь и современные информационные технологии*, XVIII Международная научно-практическая конференция студентов, аспирантов и молодых учёных (Томск, 22-26 марта 2021 г.), Сборник трудов, Изд-во ТПУ, Томск, 2021, 405–406.
- [15] Рогачев Г.Н., “Генетические алгоритмы в задачах параметрического синтеза оптимальных систем управления”, *Вестник Самарского государственного технического университета. Серия: Технические науки*, 2005, № 33, 67–72.

- [16] Метлицкая Д.В., “Генетические алгоритмы поиска оптимального управления непрерывными детерминированными системами”, *Труды МАИ*, **45**:45 (2011), 3–18.
- [17] O. V. Druzhinina, E. A. Kaledina, O. N. Masina, “Modeling of the Multiply Connected Controlled Dynamical Systems”, *2018 10th International Congress on Ultra Modern Telecommunications and Control Systems (ICUMT)*, Moscow, Russia, 2018, 383–387.
- [18] Лизина Е.А., Щенников В.Н., “Стабилизация многосвязной управляемой гибридной динамической системы с неперекрывающимися декомпозициями”, *Известия высших учебных заведений. Поволжский регион. Физико-математические науки.*, **4**:20 (2011), 14–23.
- [19] Зубов В.И., *Лекции по теории управления*, «Наука», Москва, 1975, 496 с.
- [20] Petridis V., Petridis S., “Construction of Neural Network Based Lyapunov Functions”, *2006 IEEE International Joint Conference on Neural Network Proceedings*, Vancouver, BC, Canada, 2006, 5059–5065.
- [21] Красовский Н.Н., *Некоторые задачи теории устойчивости движения*, «Физматлит», Москва, 1959, 222 с.

Stabilization of a controlled multiply connected system using a genetic algorithm

Kaledina E.A., Guskova I.M.

In this work, a multiply connected controlled dynamic system with local control is reviewed. Based on a genetic algorithm, a method for constructing local controls that stabilize the system as a whole has been developed. In this case, the parameters of the genetic algorithm are used, namely the selection, crossover, mutation operators, the type of fitness function and the critical basis. Modeling and analysis of the results obtained were carried out.

Keywords: genetic algorithms, multiply connected dynamic system, local control, asymptotic stability, stabilization.

References

- [1] Dai H., Landry B., Yang L., Pavone M., Tedrake R., “Counter-example guided synthesis of neural network Lyapunov functions for piecewise linear systems”, *2020 59th IEEE Conference on Decision and Control (CDC)*, 2020, 1274–1281.

- [2] Dai H., Landry B., Yang L., Pavone M., Tedrake R., *Lyapunov-stable neural-network*, 2021, arXiv: nucl-ex/2109.14152.
- [3] Chen S., Fazlyab M., Morari M., Pappas G.J., Preciado V.M., “Learning Lyapunov Functions for Hybrid Systems”, *2021 55th Annual Conference on Information Sciences and Systems (CISS)*, Baltimore, MD, USA, 2021.
- [4] Verdier C.F., Mazo M., “Formal Synthesis of Analytic Controllers for Sampled-Data Systems via Genetic Programming”, *2018 IEEE Conference on Decision and Control (CDC)*, Miami, FL, USA, 2018, 4896–4901.
- [5] Ben Sassi M. A., Sankaranarayanan S., Chen X., Ábrahám E., “Linear relaxations of polynomial positivity for polynomial Lyapunov function synthesis”, *MA Journal of Mathematical Control and Information*, **33**:3 (2016), 723–756.
- [6] Khapkin D.L., Feofilov S.V., Kozyr A.V., “Study of neural network control stability based on mixed linear integer programming”, *4th International Conference on Control Systems, Mathematical Modeling, Automation and Energy Efficiency (SUMMA)*, Lipetsk, Russia, 2022, 326–329.
- [7] Khapkin D.L., Feofilov S.V., Kozyr A.V., “Synthesis of a stable neural network closed-loop control system for a substantially nonlinear object with optimization of control time”, *Computer science: problems, methods, technologies*, XXIII International Scientific and Practical Conference named after. E.K. Algazinova (Voronezh, February 15–17, 2023), Conference materials, Voronezh State University, Voronezh, 2023, 405–406 (In Russian).
- [8] Sedova N.O., Tokmakov S.V., “On the use of neurocontrol with delayed feedback in the problem of output stabilization”, *Fuzzy systems and soft computing*, **15**:1 (2020), 26–42 (In Russian).
- [9] Zhou R., Quartz T., De Sterck H., Liu J., *Neural Lyapunov Control of Unknown Nonlinear Systems with Stability Guarantees*, arXiv: nucl-ex/2206.01913.
- [10] Khaikin S., *Neural networks: a complete course*, "Williams", Moscow, 2008 (In Russian), 1103 c.
- [11] Virsanski E., *Genetic algorithms in Python*, DMK Press, Moscow, 2020 (In Russian), 286 c.

- [12] Emelyanov V.V., Kureichik V.V. , Kureichik V.M., *Theory and practice of evolutionary modeling*, «Fizmatlit», Moscow, 2003 (In Russian), 432 c.
- [13] Zenkin A.,Peregudin A., Bobtsov, A., *Lyapunov function search method for analysis of nonlinear systems stability using genetic algorithm.*, 2023, arXiv: nucl-ex/2307.03030.
- [14] Belyaev A.S., Sumenkov O.Yu., “Application of a genetic algorithm for synthesizing the parameters of a linear-quadratic regulator of a reference weighting system”, *Youth and modern information technologies*, XXVIII International Scientific and Practical Conference of Students, Postgraduate Students and Young Scientists (Tomsk, March 22-26, 2021), Collection of works, TPU Publishing House, Tomsk, 2021, 405–406 (In Russian).
- [15] Rogachev G.N., “Genetic algorithms in problems of parametric synthesis of optimal control systems”, *Bulletin of Samara State Technical University. Series: Technical Sciences*, 2005, № 33, 67–72 (In Russian).
- [16] Metlitskaya D.V., “Genetic algorithms for searching for optimal control of continuous deterministic systems”, *Proceedings of MAI*, **45**:45 (2011), 3–18 (In Russian).
- [17] O. V. Druzhinina, E. A. Kaledina, O. N. Masina, “Modeling of the Multiply Connected Controlled Dynamical Systems”, *2018 10th International Congress on Ultra Modern Telecommunications and Control Systems (ICUMT)*, Moscow, Russia, 2018, 383–387.
- [18] Lizina E.A., Shchennikov V.N., “Stabilization of a multiply connected controlled hybrid dynamic system with non-overlapping decompositions”, *News of higher educational institutions. Volga region. Physical and mathematical sciences*, **4**:20 (2011), 14–23 (In Russian).
- [19] Zubov V.I., *Lectures on control theory*, «Science», Moscow, 1975 (In Russian), 496 c.
- [20] Petridis V., Petridis S., “Construction of Neural Network Based Lyapunov Functions”, *2006 IEEE International Joint Conference on Neural Network Proceedings*, Vancouver, BC, Canada, 2006, 5059–5065.
- [21] Krasovsky N.N., *Some problems of the theory of motion stability*, «Fizmatlit», Moscow, 1959 (In Russian), 222 c.

Электронная цифровая подпись на основе кодов, определяющих изображения с точностью до аффинных преобразований

В. Н. Козлов¹

Первый и давний вариант защиты документа от подделки (используется и поныне) — это так называемая “живая” подпись (или факсимиле), и канцелярская печать. Однако в наши дни документооборот большей частью электронный, и, зачастую, с очень большим числом документов (электронные торги, банковские платежные системы, сделки в криптовалютах, и пр.). Здесь работает возникшая более сорока лет назад цифровая подпись. Как правило, сердцевиной цифровой подписи является функция, у которой ее значение при заданном значении аргумента вычисляется легко, а обратное, т.е. вычисление значения аргумента при известном значении функции, очень трудно.

В статье описывается аналог цифровой подписи на другой принципиальной основе, с использованием кодов изображений, определяющих их с точностью до аффинных преобразований.

Ключевые слова: электронная цифровая подпись, цифровая подпись, изображение, код изображения, аффинные преобразования, аутентификация изображения.

1. Введение

Здесь описан способ защиты электронного изображения от подделки и проверки подлинности полученной информации на основе аффинных преобразований. Способ предназначен для работы с любыми изображениями (потенциально, в том числе, и с трехмерными), или с информацией, приводимой к изображениям (например, со звуком). Этот способ может быть использован для защиты канала связи между отправителем информации в виде изображения и получателем от попыток третьих сторон отправить получателю информацию под видом информации от отправителя, а также аутентификации полученного изображения. Способ является аналогом известной электронной цифровой подписи, но на иной принципиальной базе. В основе принципа — теорема, доказанная более двадцати лет назад.

Описываемый способ запатентован [1, 2].

¹Козлов Вадим Никитович — профессор каф. математической теории интеллектуальных систем мех.-мат. ф-та МГУ, e-mail: vnkozlov@mail.ru.

Kozlov Vadim Nikitovich — professor, Lomonosov Moscow State University, Faculty of Mechanics and Mathematics, Chair of Mathematical Theory of Intellectual Systems.

2. Теоретические основы способа

Изображением [3, 4, 5] называем конечное (непустое) множество точек в евклидовых пространствах разной размерности. В частности, двумерное изображение — конечное множество точек на плоскости. Обосновываем это тем, что любую фигуру можно “аппроксимировать” конечным множеством точек, которые уже сами по себе делают фигуру вполне узнаваемой. При этом если точек много, то такая совокупность точек практически неотличима от исходной фигуры. Так же можно представлять и полутоновые, черно-бело-серые изображения, при этом разная плотность точек в разных частях изображения дает разные оттенки “серого цвета”. Как известно, цветное изображение можно представлять как наложение трех монохроматических (аналогов черно-бело-серых) изображений. Это означает, что совокупностями точек можно представлять и цветные изображения. Трехмерные изображения — точки в трехмерном евклидовом пространстве. Наконец, трехмерный мир в динамике можно рассматривать как четырехмерное изображение (последовательность трехмерных сцен).

Далее рассматриваются двумерные изображения, но сказанное несложно обобщается и на случаи большей размерности.

Используется код изображения, который определяется следующим образом. Пусть задано изображение A произвольного вида (с числом точек больше трех). Перенумеровываем точки изображения A с единственным условием: разные точки — разные номера. Пусть M_A — множество номеров. Пусть S_{mnu} и S_{ksp} — площади треугольников с вершинами в точках с номерами соответственно m, n, u и k, s, p , и пусть $\rho_{mnu, ksp} = S_{mnu}/S_{ksp}$ (разумеется, S_{ksp} не равно нулю), и все числа ρ для всех пар троек m, n, u и k, s, p составляют множество T_A . При этом множество T_A — это только числа вида $\rho_{mnu, ksp}$, никакой информации об исходных треугольниках и их площадях уже нет. Код изображения A — пара $\langle M_A, T_A \rangle$. Изображения A и B с кодами $\langle M_A, T_A \rangle$ и $\langle M_B, T_B \rangle$ назовем эквивалентными, если существует такая биекция $\psi : M_A \rightarrow M_B$, что для любых m, n, u и k, s, p из M_A выполнено $\rho_{mnu, ksp} = \rho_{\psi(m)\psi(n)\psi(u), \psi(k)\psi(s)\psi(p)}$. Ясно, что эквивалентность изображений содержательно означает одинаковость их кодов с точностью до перенумерации точек. Два изображения называем аффинно эквивалентными, если они переводимы друг в друга аффинными преобразованиями.

Изображение назовем плоским, если все его точки не лежат на одной или двух параллельных прямых. Этим из рассмотрения исключаются два довольно узких и не существенных для практики класса изображений — все точки (именно все) лежат на одной или двух параллельных прямых.

Теорема 1 ([3, 4, 5, 6]). *Два плоских изображения эквивалентны тогда и только тогда, когда они аффинно эквивалентны.*

Отметим, что эта теорема доказана также в варианте для трехмерного пространства, и, в целом, для n -мерного ($n > 3$) [7].

Содержательно аффинные преобразования — это любая комбинация преобразований параллельного переноса, вращений, симметрии, изменений в размерах, сжатий и растяжений по любому направлению. Теорема означает, что любое изображение B , переводимое в A аффинными преобразованиями, имеет тот же (с точностью до перенумерации точек) код, что и A .

И обратное, если восстанавливать изображение по коду $\langle M_A, T_A \rangle$, то получающееся изображение заведомо переводится в A аффинными преобразованиями.

Тем самым, код определяет изображение с точностью до аффинных преобразований.

Возможно использование и укороченного кода, когда в T_A присутствуют числа $\rho_{mnu, ksp}$ только для четырехточечников из A , но всех возможных четырехточечников. Это значит, что из исходного T_A отбираются только те числа $\rho_{mnu, ksp}$, у которых тройки m, n, u и k, s, p разнятся только одним номером. Такой код тоже представляет изображение с точностью до аффинных преобразований.

Возможное сокращение состоит еще в том, что, как следует из определения кода, если в нем есть элемент $\rho_{mnu, ksp}$ с двумя тройками индексов, то есть и обратный элемент — единица, деленная на $\rho_{mnu, ksp}$, поэтому можно оставить в коде только те элементы, которые меньше единицы, поскольку то, что больше единицы однозначно восстанавливается.

Помимо кода $\langle M_A, T_A \rangle$ изображения A рассматриваем еще и стертый код: это множество Q_A чисел, получающихся из чисел множества T_A , но с удаленными индексами при них. Назовем Q_A стертым кодом для A . Восстановить по стертому коду Q_A код $\langle M_A, T_A \rangle$, а значит и изображение A , практически невозможно [8].

Возможно и некое огрубление стертого кода. Для этого промежуток от нуля до единицы разбиваем на N интервалов, где каждый интервал длины 2Δ , и рассматривают центры интервалов. Все числа из кода попадают каждое в свой интервал. После чего оставляют только числа, являющиеся серединами интервалов (возможно, с учетом кратности, то есть количества попавших в данный интервал элементов исходного кода). Так получается возможный огрубленный стертый код.

3. Описание алгоритма

Исходим из наличия схемы с тремя взаимодействующими субъектами: отправителем информации A , получателем информации B , и третьей стороной, злоумышленником C . Между A и B имеется канал связи для передачи сообщений (от A к B). Злоумышленник C может подключаться к каналу связи и, имитируя отправителя A , посылать к получателю B свои (фальшивые) сообщения. Задача состоит в том, чтобы получатель имел возможность отсортировать фальшивые сообщения от истинных. Для этой цели отправитель имеет закрытый ключ — некоторое изображение Z , получатель — открытый ключ, представляющий собой множество чисел, являющихся стертым кодом закрытого ключа Z . Злоумышленнику закрытый ключ не известен, открытый — известен. Посылаемое к B сообщение — изображение X с числом точек не меньше четырех (по смыслу задачи — существенно больше четырех). Злоумышленник предполагается предельно информированным: он имеет возможность читать посланные сообщения, в том числе, располагает архивом всех ранее посланных сообщений, знает о существовании закрытого ключа (сам ключ не знает), знает о процедуре формирования сообщений (в том числе о вводимых ниже проверочных изображениях), все о процедуре проверки полученного сообщения у получателя, знает, как уже упоминалось, открытый ключ, т.е. знает все, кроме закрытого ключа.

Далее — более подробно об этой схеме, об алгоритме работы описываемой модели, и о его этапах (пункты I – VII).

- I. У отправителя формируют закрытый ключ безопасности Z в виде электронного точечного изображения. Закрытый ключ имеется у отправителя и исходно представляет собой преимущественно физический носитель, такой как магнитная карта, листовая документ с магнитными метками, документ с печатными метками или символами (буквами, цифрами и пр.). Таким образом, закрытый ключ исходно — это любой объект, содержащий информацию, эквивалентом которой может быть электронное изображение. Каждая магнитная частица может быть представлена в виде точки и каждый пиксель отсканированного документа также может быть представлен в виде точки или совокупности точек, образуя электронные изображения в виде конечного множество точек на декартовой плоскости. Для дальнейшей работы необходимо иметь указанное электронное изображение Z . Преобразование пикселя в точку и магнитной частицы в точку осуществляется очевидными простыми алгоритмами. В качестве примера, пусть дано электронное пиксельное чёрно-белое изображение, у пикселей есть градации темноты нуль (белый пиксель) и единица (черный пиксель). В этом

случае действуют следующим образом: пиксель — это маленький квадрат, и имея черно-белое изображение, белый пиксель оставляют пустым, а вместо черного пикселя ставят точку. И изображение из пикселей преобразуется в изображение из точек. Если изображение полутоновое чёрно-белое, тогда пиксели имеют градацию темноты от нуля до $K - 1$. Преобразовывают это в изображение из точек следующим образом: снова ставят в квадрат точки, если с градацией единица (не белый квадрат) — ставят одну точку, если градация два (уже темнее) ставят 2 точки в этом квадрате и так далее вплоть до $K - 1$ (до самого темного), и изображение из пикселей превращается в изображение из точек, где разная плотность точек в разных частях изображения обеспечивает светлый или темный тон. Градация определяется тем, что в пиксельных изображениях каждый квадрат уже изначально имеет одну из K — градаций, в этом состоит суть пиксельного изображения, т.е. градации определяются любыми доступными графическими программами.

Множествами точек можно представлять и цветные изображения, тогда необходимо применить преобразование пикселей в точки для каждого из цветных каналов RGB отдельно. Либо преобразовывают изображение в черно-белое, а только затем в точечное.

Предполагается, что магнитные частицы-точки изначально распределены хаотично, неважно откуда — с бумажного носителя или с магнитного. По аналогии с пиксельным преобразованием, если есть частица, в квадрате ставят точку, если нет — оставляют пустой, если есть скопления точек, обращаются к градациям.

Слова “изображение”, “электронное изображение”, “электронное точечное изображение” с очевидностью довольно близки по смыслу (исключая электронное пиксельное изображение), поэтому далее они рассматриваются как практически синонимы.

- II. Далее формируют из подлинной исходной информации для отправления электронное пиксельное изображение. Фото, документы, иные материалы для отправления оцифровывают любыми известными аппаратными средствами, например, сканируют, так получают электронное пиксельное изображение для следующего этапа. Либо данные для передачи могут быть у отправителя в виде электронного пиксельного изображения изначально.
- III. Из полученного электронного пиксельного изображения формируют сообщение X в виде электронного точечного изображения по аналогии с п. I.

- IV. Кодируют закрытый ключ Z , формируя для получателя открытый ключ — конечное множество чисел Q_Z кода закрытого ключа Z .
- V. Формируют как минимум одно проверочное аффинно-эквивалентное части закрытого ключа Z электронное точечное изображение Y у отправителя. В начале этапа у отправителя есть электронное точечное изображение X , которое он должен отправить получателю. У получателя есть открытый ключ. Ценность этапа состоит в том, что отправитель посылает получателю информацию, и предусмотрено средство, которое не позволит злоумышленнику, даже располагая подробной информацией о модели, получить данные о закрытом ключе и выдать себя за отправителя. Для этого отправляют сообщение X получателю, но для защиты снабжают его одним или несколькими дополнительными изображениями Y — проверочными. Поэтому в конце этапа у отправителя должны быть изображение X — сообщение, и некоторые изображения $Y_1 - Y_k$ — проверочные, сформированные у отправителя с помощью закрытого ключа, как будет описано ниже. Изображение Y — проверочное, тест и защита, его роль — участвовать в проверке подлинности сообщения X . Полагаем, изображение Y аффинно эквивалентно с некоторыми подизображениями (т.е. подмножествами точек) одновременно как в X , так и в Z , причем это максимальное по количеству точек такое изображение, т.е. при добавлении хотя бы одной точки аффинная эквивалентность нарушится.

Обозначают через $Y + 1$ изображение “с добавленной точкой”, т.е. это изображение в сравнении с Y имеет добавленную точку, и оно аффинно-эквивалентно с некоторым подмножеством точек в X . По построению, $Y + 1$ уже не аффинно-эквивалентно ни с каким подизображением в Z (это и означает максимальность исходного изображения Y). Множество всех таких $Y + 1$ обозначим через $\{Y + 1\}$. Для каждого $Y + 1$ из $\{Y + 1\}$ проверяем условие: стертый код для $Y + 1$ не является подмножеством стертого кода Q_Z . Если условие не выполняется, то $Y + 1$ называем особым. Отправителю нет нужды направлять получателю информацию о множестве $\{Y + 1\}$ — оно легко строится по X и Y , исключая особые изображения, о них получателю сообщается. Здесь описана ситуация с одним проверочным изображением, но их может быть и несколько.

Отметим для корректности, что полагаем исключенными очевидно вырожденные и легко проверяемые случаи, когда X совпадает с Z , или Z является частью X , или X является частью Z .

- VI. Посылают получателю сообщение X и как минимум одно полученное проверочное электронное точечное изображение Y .
- VII. Осуществляют аутентификацию полученного сообщения X по проверочному изображению Y и по стертому коду Q_Z .

Проверку по аффинной эквивалентности электронного точечного изображения Y подмножеству точек сообщения X и по Q_Z проводят следующим образом:

- 1) проверяется, является ли изображение Y аффинно-эквивалентным с некоторыми подизображениями (подмножествами точек) сообщения X ,
- 2) является ли стертый код Q_Y проверочного изображения Y подмножеством стертого кода Q_Z закрытого ключа Z (т.е. подмножеством открытого ключа),
- 3) а) является ли стертый код каждого из изображений из $\{Y + 1\}$, исключая особые, подмножеством открытого ключа,
б) является ли стертый код каждого из особых изображений в $\{Y + 1\}$ подмножеством открытого ключа.

Ответы “да” на вопросы 1 и 2, “нет” на 3а и “да” на 3б означают, что сообщение X подлинное и получено именно от авторизованного отправителя.

4. Возможные дополнения (опции)

Далее описаны (пункты 1 – 5) некоторые возможные дополнения для использования в алгоритме, представленном этапами I – VII.

1. Посылаемое изображение X может оказаться чрезмерно большим по количеству составляющих его точек, поэтому электронные точечные изображения подлинной исходной информации сообщения X могут быть сжаты в меньшие по объему с сохранением защиты данных. Для этого после формирования сообщения X в виде электронного точечного изображения на него накладывают виртуальную сетку из квадратных ячеек, задают каждому квадрату градацию по количеству точек в каждой ячейке и образуют электронное точечное изображение с меньшим числом точек (аналогично пункту 1 в описании этапов алгоритма, т.е. через своеобразное прореживание сообщения X). Возможен при этом вариант, когда новые точки в каждой ячейке (числом меньшим, чем их было прежде) выбираем из множества прежних точек. Это означает

в этом случае, что стертый код нового изображения будет подмножеством стертого кода прежнего. Получившееся новое сообщение X_1 используем далее по способу вместо сообщения X , а получателю посылаем сообщение X с наложенной виртуальной сеткой и изображение X_1 .

2. Стертый код Q упрощают и сокращают следующими возможными способами:

- а) убирают в нем повторение чисел, т.е. от каждого множества одинаковых чисел (если они есть) оставляют в нем только одно число,
- б) оставляют в нем только числа меньше или равные единице,
- в) разбивают сегмент от 0 до 1 на N частей длины $1/N = 2\Delta$.

Затем заменяют каждое число в стертом коде на середину части, в которую он попадает. Обозначают через Q'_Z множество получившихся чисел.

Эти три пункта меняют вопрос 2 при проверке аутентификации в преимущественном описании способа выше следующим образом:

- 1) имеется ли для каждого числа, меньшего или равного единице, из стертого кода Q_Y проверочного изображения Y , число из Q'_Z , отличающееся от него не более, чем на Δ . Вопросы в пункте 3 меняются на следующие:
- 2) а) имеется ли для каждого числа, меньшего или равного единице, из стертого кода каждого из изображений в $\{Y + 1\}$, исключая особые, число из Q'_Z , отличающееся от него не более, чем на Δ .
б) имеется ли для каждого числа, меньшего или равного единице из кода особого изображения в $\{Y + 1\}$, число из Q'_Z , отличающееся от него не более, чем на Δ .

В целом это дополнение к основному описанию сокращает стертый код (открытый ключ), и упрощает алгоритм проверки сообщения X на достоверность.

3. Особые изображения среди изображений множества $\{Y + 1\}$ из пункта V описания способа возникают по следующим причинам. Если некоторое изображение B аффинно эквивалентно части изображения A , то стертый код Q_B есть подмножество стертого кода

Q_A , но обратное неверно, т.е. стертый код Q_B может быть подмножеством стертого кода Q_A , но B при этом не аффинно эквивалентно части изображения A . Элементы стертых кодов — просто обезличенные числа, поэтому среди большого множества чисел Q_A могут “разрознено”, “случайно” найтись все числа из небольшого Q_B , хотя вместе они в A не составляли код некоторого подизображения. Такое маловероятно, но возможно, поэтому такое изображение и называется особым.

4. Поиск на изображении Z (закрытом ключе) и X (сообщении) общего подизображения Y , а также проверка получателем совпадения (разумеется, с точностью до аффинных преобразований) проверочного изображения Y с каждым из некоторого множества частей сообщения X , имеют нечто общее, заключающееся в том, что нужно определить, является ли некоторое изображение B частью (с точностью до аффинных преобразований) некоторого изображения A . Конечно, это можно сделать тривиальным перебором по всем подизображениям, но это не рационально. Делают так: на изображении A выбирают три точки a_1, a_2, a_3 (произвольные, но не лежащие на одной прямой). Выбирают на B три точки b_1, b_2, b_3 (произвольные, но не лежащие на одной прямой). Совмещают аффинным преобразованием точки b_1, b_2, b_3 с точками соответственно a_1, a_2, a_3 , такое преобразование существует и единственно. Если B аффинно эквивалентно с частью изображения A , причем с соответствием точек b_1, b_2, b_3 точкам a_1, a_2, a_3 , то и остальные точки изображения B совмещаются с соответствующими точками изображения A . Если нет, то проделывается описанное со всеми тройками точек на A и всеми вариантами соответствия между точками в тройках из A и B . Либо получается совмещение B с частью A , либо делается вывод, что B аффинно не эквивалентно никакой части в A .
5. В пункте V описания алгоритма в рамках поиска на изображении Z (закрытом ключе) и X (сообщении) общего для них подизображения Y , а также проверке получателем совпадения (с точностью до аффинных преобразований) проверочного изображения Y с каждым из некоторого множества частей сообщения X , присутствовала общая для них подзадача, заключающаяся в том, что нужно определить, является ли некоторое изображение B частью (с точностью до аффинных преобразований) некоторого изображения A . Но может статься, что такого общего подизображения Y , с числом точек больше трех, нет. Тогда для Y , по-прежнему аффинно эквивалентного части X , полного аффинного совпадения с частью в Z можно не требовать, а требовать только “приблизительного совпа-

дения”, с “зазором ε ”, и называть это ε -эквивалентностью. Выяснение, является ли некоторое изображение B частью (с точностью до ε -эквивалентности) некоторого изображения A , выглядит следующим образом.

Выбирают на B три точки b_1, b_2, b_3 (произвольные, но не лежащие на одной прямой), ставят им в соответствие три произвольные (но не на одной прямой) точки a_1, a_2, a_3 изображения A . Берут на B точку x_1 , сопоставляют ей некоторую точку y_1 на A . Если теперь элементы кода изображения из точек b_1, b_2, b_3 и x_1 отличаются от соответствующих элементов кода изображения из точек a_1, a_2, a_3 и y_1 на A не более, чем на ε , то точку y_1 называют приемлемой при соответствии с точкой x_1 . Если точка y_1 оказалась не приемлемой, то берут другую точку из A в качестве точки y_1 и повторяют рассуждение. Затем то же делают для точки x_2 , и т.д. Пусть B состоит из точек $b_1, b_2, b_3, x_1, \dots, x_k$, и для точек x_1, \dots, x_k найдены соответствующие приемлемые точки y_1, \dots, y_k на A . Тогда для изображения B из точек $b_1, b_2, b_3, x_1, \dots, x_k$ и изображения A' из точек $a_1, a_2, a_3, y_1, \dots, y_k$ (при указанном соответствии) проверяют в целом различие всех соответствующих элементов их кодов не более, чем на ε . При положительном результате проверки подизображение A' изображения A называют искомым.

Если первоначально выбранные тройки точек a_1, a_2, a_3 и b_1, b_2, b_3 не дали возможность для каждой из точек x_1, \dots, x_k подобрать приемлемые точки y_1, \dots, y_k , то проделывается описанное со всеми тройками точек на A и B и всеми вариантами соответствия между точками в тройках. В результате либо находят искомое изображение, либо делают вывод, что такого изображения нет.

Разумеется, при сравнении чисел из Q_Y с числами из Q_Z для Y , подобранного в рамках описанной процедуры, нужно учитывать величину ε .

В частном случае при $\varepsilon = 0$ процедура с очевидностью превращается в поиск на A части, аффинно эквивалентной с B , но, в отличие от предыдущего пункта 4, только по коду, без использования геометрических преобразований.

5. Заключение. Атаки на подпись

В заключение представим три типа возможных “атак” на модель со стороны третьих лиц и ее “реакцию” на атаки. Атака может состоять в том, что третье лицо со стороны, не имеющее закрытого ключа, пытается отправить получателю сообщение под видом сообщения от отправителя.

Полагаем, он располагает максимально подробной информацией о модели, ее алгоритмах, включая примеры ранее посланных сообщений с проверочными изображениями.

Первый и простой случай состоит в том, что вместе с X в качестве проверочного отправляется изображение Y , взятое наугад из числа подизображений сообщения X . Поскольку Y не является подизображением для Z , то проверка на открытом ключе выявит это. Конечно, возможно случайное совпадение Y с подизображением на Z , однако это маловероятно. Действительно, по смыслу предполагаемых разными изображениями X и Z у них сравнительно немного аффинно совпадающих подизображений. При этом если, например, X содержит 100 точек (не очень много), то оценка сверху для числа всех его подизображений — два в степени 100. Это число с тридцатью знаками. Маловероятно, чтобы наугад взятое из этого множества изображение совпало с одним из общих для X и Z .

Более сложный случай — взять одно из ранее посланных проверочных изображений Y , присоединить его как часть к фальшивому изображению, все вместе назвать сообщением X , и Y сделать при нем проверочным изображением. Тогда Y пройдет проверку и на то, чтобы быть подизображением сообщения X , и на то, что его стертый код есть подмножество открытого ключа. Но он не выдержит проверку на максимальность как подизображения в “составном” сообщении X , то есть проверку на стертые коды изображений из множества $\{Y + 1\}$ на открытом ключе.

Если, наконец, полагать, что это третье лицо, собрав статистику всех ранее посланных проверочных изображений, попытается по ней восстановить закрытый ключ, то оно столкнется с комбинаторной проблемой, некоторым аналогом которой, например, можно считать гипотезу Улама из теории графов [9]. Эта проблема сложна, стоит с 1945 года и до сих пор не решена.

Список литературы

- [1] Козлов В. Н., Способ аутентификации электронного изображения. Патент на изобретение №2779379. Дата государственной регистрации в Государственном реестре изобретений Российской Федерации 06 сентября 2022 года.
- [2] Козлов В. Н., Способ защиты электронного изображения на основе аффинных преобразований. Патент на изобретение №2791834. Дата государственной регистрации в Государственном реестре изобретений Российской Федерации 13 марта 2023 года.

- [3] Козлов В. Н., *Введение в математическую теорию зрительного восприятия*, М.: Издательство Центра прикладных исследований при механико-математическом факультете МГУ, 2007, 136 с.
- [4] Kozlov V.N., “Conclusiveness and Heuristics in Visual Recognition”, *Pattern Recognition and Image Analysis: Advances in Mathematical Theory and Applications*, **24**:4 (2014), 1–7
- [5] Kozlov V.N., “Image Coding and Recognition and Some Problems of Stereovision”, *Pattern Recognition and Image Analysis: Advances in Mathematical Theory and Applications*, **7**:4 (1997), 448–466
- [6] Кудрявцев В. Б., Гасанов Э. Э., Подколзин А. С., *Введение в теорию интеллектуальных систем*, МАКС Пресс, 2006, 208 с.
- [7] Агниашвили П. Г., “О восстановлении изображений по кодам в некоторых вырожденных случаях”, *Интеллектуальные системы. Теория и приложения*, **17**:1-4 (2013), 11-15
- [8] Алексеев Д. В., “Необходимые и достаточные условия существования изображения с заданным кодом”, *Интеллектуальные системы. Теория и приложения*, **24**:2 (2020), 55–66
- [9] Емеличев В. А., О.И.Мельников О. И.В., Сарванов В. И., Тышкевич Р. И., *Лекции по теории графов*, Наука, 1990, 384 с.

Electronic digital signature based on codes, defining images up to affine transformations

Kozlov V.N.

The first and long-standing variant of protecting a document from forgery (it is still used today) is the so-called “living” signature (or facsimile), and a clerical seal. However, nowadays the document flow is mostly electronic, and often with a very large number of documents (electronic trading, bank payment systems, transactions in cryptocurrencies, etc.). The digital signature that emerged more than forty years ago works here. As a rule, the core of a digital signature is a function whose value is easily calculated for a given argument value, and the reverse, i.e. calculating the value of an argument given the value of a function is very difficult.

The article describes an analogue of a digital signature on a different fundamental basis, using image codes that define them up to affine transformations.

Keywords: digital signature, image, image code, affine transformations, image authentication.

References

- [1] Kozlov V. N., A method for authenticating an electronic image. Patent for invention No. 2779379. Date of state registration in the State Register of Inventions of the Russian Federation September 06, 2022.
- [2] Kozlov V. N., A method for protecting an electronic image based on affine transformations. Patent for invention No. 2791834. Date of state registration in the State Register of Inventions of the Russian Federation March 13, 2023.
- [3] Kozlov V. N., *Vvedenie v matematicheskuyu teoriyu zritel'nogo vospriyatiya [Introduction to the mathematical theory of visual perception]*, Publishing House of the Center for Applied Research at the Faculty of Mechanics and Mathematics of Moscow State University, Moscow, 2007 (in Russian), 136 c.
- [4] Kozlov V. N., “Conclusiveness and Heuristics in Visual Recognition”, *Pattern Recognition and Image Analysis: Advances in Mathematical Theory and Applications*, **24**:4 (2014), 1–7
- [5] Kozlov V. N., “Image Coding and Recognition and Some Problems of Stereovision”, *Pattern Recognition and Image Analysis: Advances in Mathematical Theory and Applications*, **7**:4 (1997), 448–466
- [6] Kudryavtsev V. B., Gasanov E. E., Podkolzin A. S., *Vvedenie v teoriyu intellektual'nykh sistem [Introduction to the theory of intelligent systems]*, MAKS Press, 2006 (in Russian), 208 c.
- [7] Agniashvili P. G., “On image restoration from codes in some degenerate cases”, *jour Intelligent systems. Theory and applications*, **17**:1-4 (2013), 11-15 (in Russian)
- [8] Alekseev D. V., “Necessary and sufficient conditions for the existence of an image with a given code”, *Intelligent Systems. Theory and Applications*, **24**:2 (2020), 55–66 (in Russian)
- [9] Emelichev V. A., Melnikov O. I., Sarvanov V. I., Tyshkevich R. I., *Lectures on graph theory*, Science, 1990 (in Russian), 384 c.

Пять принципов многоагентной автоматизированной системы искусственного интеллекта

А. В. Чечкин¹

Цель. Исследовать эффективность взаимодействия естественного и искусственного интеллекта (ИИ) ноосферы Земли.

Методы. Математическое, информационное и виртуальное моделирования. Системный и ультрасистемный анализ и синтез.

Результаты. Обосновывается необходимость использования понятия многоагентной автоматизированной системы как массовой типовой человеко-машинной системы ИИ ноосферы Земли. Формулируются и обосновываются пять концептуальных принципов: принцип автоматизации и человекоцентризма ИИ; принцип информационно-системной избыточности и безопасности ИИ; принцип разумной целостности ИИ; принцип нештатного целеполагания ИИ; принцип пяти уровней языкового развития ИИ.

Ключевые слова: естественный и искусственный интеллект, автоматизированная система, многоагентная система, цифровые двойники, система интеллектуального планирования и ситуационного группового управления, язык интерфейса.

*Не надо бояться ИИ,
надо его возглавить*
(В.В. Путин)

Слова академика В.И. Вернадского: «...Все человечество, вместе взятое, представляет ничтожную массу вещества планеты. Мощь его связана не с его материей, но с его мозгом, разумом и направленным этим разумом его трудом. *Ноосфера* есть новое геологическое явление на нашей планете. В ней впервые человек становится крупнейшей геологической силой. Он может и должен перестраивать своим трудом и мыслью область своей жизни, перестраивать коренным образом по сравнению с тем, что было раньше...» [1].

Деятельный, сознательный интеллект человека целиком и полностью определяется его естественным *языком*. Приведем слова академика И.П.

¹ Чечкин Александр Витальевич — доктор физ.-мат. наук, профессор, Военная академия РВСН имени Петра Великого Финансовый университет при правительстве Российской Федерации, e-mail: a.chechkin@mail.ru

Chechkin Alexander Vitalievich — Doctor of Phys.-Math. Sci., Professor, Strategic Missile Forces Military Academy named after Peter the Great Financial University under the Government of the Russian Federation.

Павлова: «... *Вторая сигнальная система* представляет собой чрезвычайную прибавку к высшей нервной деятельности человека, это наше лишнее, добавочное, социальное и есть человеческое ... Слово сделало нас людьми...» [2].

Эти слова подтверждают важность выбранной нами темы исследования. Язык человека имеет дело с *данными, знаниями* и с *информацией*. Информационные системы различного назначения являются предметом *информатики* [3]. Сегодня, в связи с появлением и бурным развитием интеллектуальных помощников (агентов) человека, технических систем с элементами *искусственного интеллекта (ИИ)* [4], многоагентных интеллектуальных систем группового управления [5], цифровых двойников бизнес проектов, сложных технических, организационных и биологических систем, [6, 7], дадим следующее определение: *Всякую многоагентную автоматизированную программно-техническую систему с элементами искусственного интеллекта будем называть Автоматизированной Системой Планирования и Управления Искусственным Интеллектом (АСПУ ИИ), если для нее выполняются следующие пять принципов.*

1. Принцип автоматизации и человеко-центризма ИИ

Любая система ИИ должна функционировать всегда и только в рамках автономного агента АСПУ ИИ, где главным является человек - хозяин, оператор, начальник, командир. Человек-хозяин каждый раз планирует и выдаёт автономное тактико - техническое задание (ТТЗ) каждому агенту и жестко контролирует его выполнение. Без ТТЗ от АСПУ ИИ система ИИ не функционирует. Действует правило: «Смочь – то система ИИ сможет, но кто же ей позволит?»

АСПУ ИИ является сервисной многоагентной системой массового обслуживания (СМО). Агентами АСПУ выступают отдельные системы ИИ различного назначения от сенсоров, датчиков, роботов, цифровых двойников до систем поддержки выбора решений, планирования и эффективного исполнения. АСПУ ИИ планирует свои коллективные, групповые действия и обеспечивает групповое управление, контроль, коррекцию, согласованность и безусловное выполнение частных и общее ТТЗ. Согласно принципу, верхняя часть дерева целей каждой автономной системы ИИ всегда составляет его автономное частное ТТЗ от человека-

хозяина, а тактическую часть и оперативную часть каждая система ИИ вырабатывает и исполняет самостоятельно. В любой момент любой частный план действий каждой системы ИИ или цифрового двойника находится под контролем человека - хозяина и может быть им скорректирован.

2. Принцип информационно-системной избыточности и безопасности ИИ

Каждая система ИИ и вся АСПУ ИИ должны быть информационно – системно избыточными и безопасными в форме среды радикалов [11, 12]. Среда радикалов – это одновременно эффективная форма для избыточности сервисной системы массового обслуживания (СМО) и для избыточности адаптивной распределенной модели всего театра действий АСПУ ИИ, включая модели самих систем ИИ как действующих участников театра. Системы ИИ в АСПУ ИИ постоянно обновляются и развиваются, что должно отражаться в моделях АСПУ ИИ, в каждой самой системе ИИ и всего театра действий АСПУ ИИ.

Все что необходимо и достаточно для решения любой штатной для АСПУ ИИ задачи в штатных ситуациях должно быть представлено в форме среды радикалов. Среда радикалов – это такая форма распределенного избыточного информационно – системного ресурса АСПУ ИИ, когда каждая целостность этого ресурса представлена радикалом, а вместе среда радикалов – это сеть взаимосвязанных радикалов. При этом **радикал** – это функциональная система, которая имеет **назначение и два внешнедоступных типа состояний**: радикал активен (включен) и радикал пассивен (выключен). **Активный радикал** – это функционирующий (действующий) радикал. **Пассивный радикал** – это не функционирующая система, но готовая к активации. АСПУ ИИ каждый раз под очередную штатную задачу в штатной ситуации планирует и активирует рабочий системоквант, кластер среды радикалов. Все остальные радикалы при этом составляют потенциальные возможности АСПУ ИИ и находятся **в резерве, в ожидании, в готовности** к своей активации. Требование избыточности информационно-системного ресурса АСПУ ИИ необходимо для попыток эффективного парирования (учета) возможных изменений штатных ситуаций во всем театре

действий АСПУ ИИ и возможного появления нештатных задач. Уровень информационно-системной избыточности ИИ определяет уровень информационно – системной безопасности ИИ [11. 12]. Форма избыточности в виде среды радикалов имеет двойное качество. *Во-первых, автономность и семантическая согласованность радикалов*, когда все активные радикалы взаимно согласованы, а не востребуемые в данный момент пассивные радикалы АСПУ ИИ выключены и не влияют на требуемое поведение систем ИИ. Пассивные радикалы находятся все в ожидании, в резерве, в хранении.

Во-вторых, индивидуальная доступность радикалов. Каждый радикал АСПУ ИИ в пассивном состоянии доступен для автономной активации и независимого обновления. Для эффективности активации и обновления отдельных радикалов среда радикалов снабжена и структурирована *координатной навигационной системой*. При этом не требуется каждый раз общее согласование, перенастраивание и корректирование всей среды радикалов АСПУ ИИ целиком.

3. Принцип разумной целостности ИИ

Каждая система ИИ в АСПУ ИИ постоянно мониторит (тестирует) себя, свою среду радикалов, свой театр действий и сертифицирует их на целостность и гомеостаз. При этом целостность восстанавливается полностью или частично с массовым использованием цифровых и других двойников физических, технических, биологических, социальных и других подсистем с учетом оптимизации затрат временных, материальных, финансовых, энергетических и др. ресурсов подобно протезированию человека.

Роль и эффективность цифровых, биологических и других двойников систем постоянно возрастает в экономике, в финансах, в технике, в биологии, в социальных, в организационных системах различного назначения на Земле и в космосе.

4. Принцип нештатного целеполагания ИИ

Системы ИИ в АСПУ ИИ постоянно обновляются и развиваются, отслеживая естественные и искусственные изменения в самих системах ИИ и во всем



Рис. 1. Функционирование АСПУ ИИ в штатных и нештатных ситуациях

театре действий АСПУ ИИ. Точками роста, развития являются появления нештатных ситуаций и нештатных задач. Для эффективного обновления АСПУ ИИ и систем ИИ используются свои попытки самообучения, коммуникационные возможности обучения от человека – хозяина или поиск опыта других АСПУ ИИ, включая помощь от дружественных пользователей самой АСПУ ИИ.

В случае штатной ситуации и штатного целеполагания АСПУ ИИ функционирует штатно в рамках воспринимаемого ею штатными сенсорами мира, *сенсорного мира АСПУ ИИ*. В противном случае в АСПУ ИИ организуется процесс *синтеза нового, включая создание новых сенсорных измерительных и исполнительных систем*, и далее происходит *интеллектуальное развитие АСПУ ИИ, включая осознание новых миров АСПУ ИИ, микро-, макро- и др. миров, рис.1.*

5. Принцип пяти уровней языкового развития ИИ

Каждая система ИИ в АСПУ ИИ должна воспринимать голосовые и письменные сообщения человека-хозяина, других систем ИИ и самой себя, находясь в режиме постоянной информированности и интеллектуального развития. В режиме нештатного целеполагания АСПУ ИИ проходит пять уровней своего развития.

На практике этот принцип частично проявляется в рамках интернета вещей, языкового интерфейса людей и компьютеров, голосового и языкового управления, предобучения нейронных искусственных сетей в рамках chatGPT, [4] и цифровых двойников, [7]. Принцип требует, чтобы АСПУ ИИ и каждая система ИИ в АСПУ должны иметь специальные подсистемы, *ультрасистемы языкового общения, обеспечения информационно-системной безопасности (ИСБ) АСПУ ИИ, адаптации к новому, нештатному* [11, 12].

Ультрасистема АСПУ ИИ подобна *центральной нервной системе, мозгу человека*. Ультрасистема АСПУ ИИ отвечает за интеллектуальные качества АСПУ ИИ: *Сбор, хранение, преобразование и использование данных, знаний, информации и умений о самой «Я-АСПУ», а также об объектах и связях всего театра действий АСПУ ИИ с целью обеспечения двух сторон, информационной и системной, требования ИСБ АСПУ ИИ*, рис.2.

Информационная сторона ИСБ. Каждая штатная задача жизненного цикла АСПУ ИИ должна быть безусловно и эффективно решена, независимо от формы и полноты оперативной информации, от наличия помех, путем избыточного (радикального) моделирования расширенной проблемной области, театра действий АСПУ ИИ, защиты модели от несанкционированного доступа (НСД), логической обработки модели с целью обеспечения очередной решаемой задачи достаточной информацией.

В случае нештатной ситуации в АСПУ ИИ должна быть исследована возможность решения задачи, путем использования избыточности радикальной модели и попыток решения задачи в режиме «проб и ошибок», самообучения, развития АСПУ ИИ.

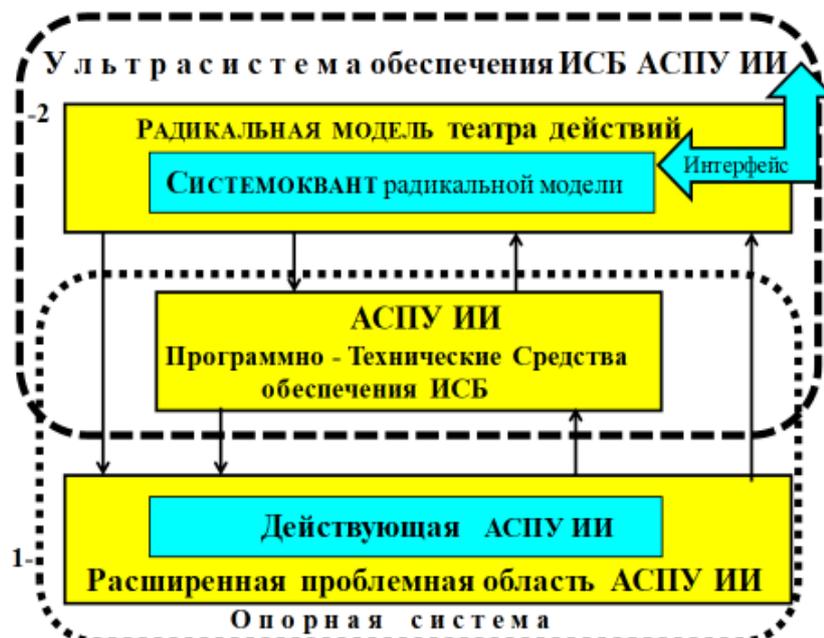


Рис. 2. Ультрасистема АС ПУ ИИ обеспечения ИСБ

Системная сторона ИСБ. Каждая задача в штатном и нештатном случае в процессе своего решения требует постоянного учета последствий решения в расширенной проблемной области АСПУ ИИ, постоянного сертифицирования радикальной модели, устранения конфликтов, сохранения системной целостности, гомеостаза АСПУ ИИ.

Принцип требует соблюдения определяющей роли для технических систем, как и для человека, первой и второй сигнальных систем [8-10]. Что, в конечном счете, обеспечивает адаптацию АСПУ ИИ к любым изменениям.

Системы первичного и языкового сенсориумов в АСПУ ИИ определяют *подсознательное* и *сознательное* поведения [9,10].

Ультрасистема АСПУ ИИ имеет пять иерархических уровней, на каждом из которых она является радикальной (избыточной) системой и находится в режиме постоянного интеллектуального обновления.

1. **Первый уровень** связан с проблемами сенсорных сетей, технического зрения, акустических сетей, проблемами сбора и хранения больших массивов метрологических измерений, с проблемами развития *пер-*

вичного сенсорiums АСПУ ИИ. Первичный сенсорium АСПУ является хранителем сенсорных *образных моделей мира [9,10].* Этот уровень исходно является *подсознательным* для АСПУ, он не осознан, не познан, не структурирован.

Второй уровень интеллектуального развития АСПУ связан с процессом *языкового осознания части первичного сенсорiums*, выделения в первичном сенсорiums *отдельных, основных, главных для АСПУ ИИ, доминирующих объектов и отношений* путем их номинирования, присвоения каждому из них языкового индивидуального, уникального символа (*мультиграф термов*), рис.3. Во втором уровне появляется *вторая сигнальная система АСПУ ИИ, языковой сенсорiums АСПУ.* В первичном сенсорiums присваиваются уникальные индивидуальные *слова-имена* выделенных объектов и отношений. Язык - средство выделения в первичном сенсорiums *осознанной (основной) его части.* Одновременно с этим язык является *механизмом познания* этой осознанной части первичного сенсорiums. В ультрасистеме АСПУ второй иерархический уровень - это *языковой сенсорiums, дубликат осознанной части первичного сенсорiums.* Языковой сенсорiums позволяет в АСПУ ИИ умозрительно изучать (активировать) эту осознанную часть первичного сенсорiums когнитивными (умственными) технологиями, минуя явного наличие реальных объектов перед сенсорами АСПУ ИИ. Среди объектов мира «Я-АСПУ» имеются объекты: «программное обеспечение (ПО) АСПУ», «программно-технические средства (ПТС) АСПУ» и др., рис.2. Ультрасистема АСПУ ИИ сама оценивает свои ПО и ПТС. В ней происходит изучение на основе языкового сенсорiums объективной структуры окружающего мира.

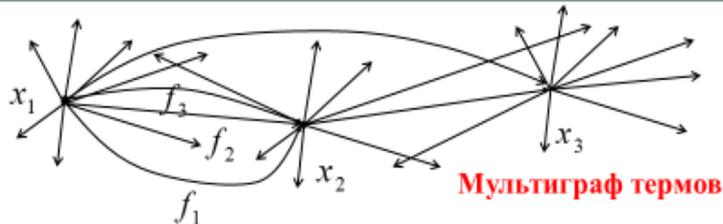
Третий уровень развития АСПУ ИИ связан с появлением *лексики, слов – понятий, опорных множеств и опорных операторов.* Развивается *языковая семантическая координатная опорная сеть [3, 10].* Лексика завершает выделение в осознанной части первичного сенсорiums структуры классов *сходных объектов и отношений*, которые *объективно похожи и эквивалентны* между собой, рис.3 (*категория SET*).

Далее в ультрасистеме АСПУ ИИ, рис.2, в осознанной части первичного сенсорiums, рис.3 должны реализоваться *два взаимодополняющих* друг друга процесса В результате чего развивается

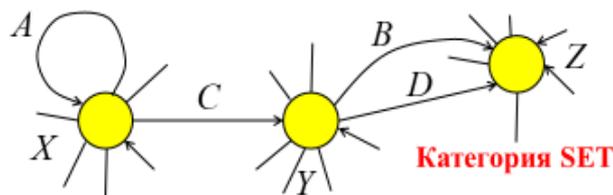
Четвертый информационно-системный иерархический уровень над опорной сетью, когда происходит внутренняя координатизация опорных множеств до *ультрамножеств*, рис.4 и опорных операторов до *ультраоператоров*, рис.5. В ультрасистеме происхо-

Переход к третьему уровню ИИ - лексика

Область языковых термов-имен сенсорных образных моделей



Координатная локализация (классификация) термов и их связей



Лексика - опорная сеть понятий, множества и операторы

Рис. 3. Математические модели осознанной части первичного сенсорiums

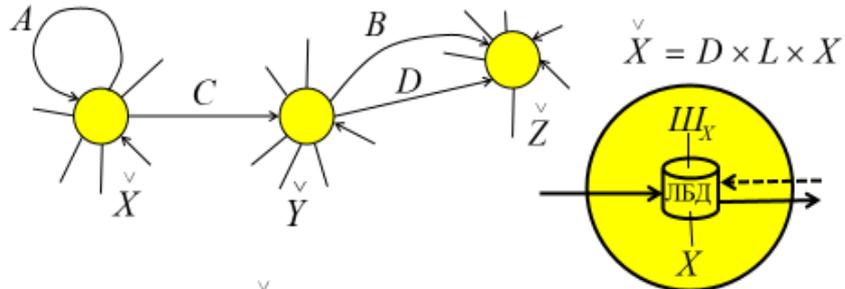
дят два *ультраоснащения* опорной координатной сети понятий, *ультрамножественное* и *ультраоператорное*. При этом каждое *опорное множество* оснащается до *ультрамножества*, т.е. до *локальной базы данных (ЛБД)*, рис.4. Каждый *опорный оператор* оснащается до *ультраоператора*, т.е. до *локальной базы знаний (ЛБЗ)*, рис.5.

Категория SET, рис.3 преобразуется в *категорию ULTRASET*, рис.5. В ультрасистеме АСПУ ИИ появляется информационно-системная избыточная *математическая модель мира АСПУ ИИ* в форме *распределенной базы данных (БД)*, рис.4 и *распределенной базы знаний (БЗ) осознанной части первичного сенсорiums*, рис.5. На четвертом этапе развития АСПУ решаются сначала проблемы создания *локальных баз данных (ЛБД)* и организация из них *больших данных в распределенную БД АСПУ ИИ*. В АСПУ ИИ *ультраоператорное оснащение* координатной сети понятий происходит до формы *распределенной базы знаний (БЗ) осознанной части первичного сенсорiums*, рис.5.

Отметим, что для человека в жизни подобный процесс происходит в форме создания, формализации и реализации на практике частных профессиональных технологий. Проблемы создания *локальных баз*

Ультраоснащение опорной сети в распределенную БД ИИ

Ультраоснащение опорных множеств в ультрамножества – это их локальная координатизация и учет достоверностей

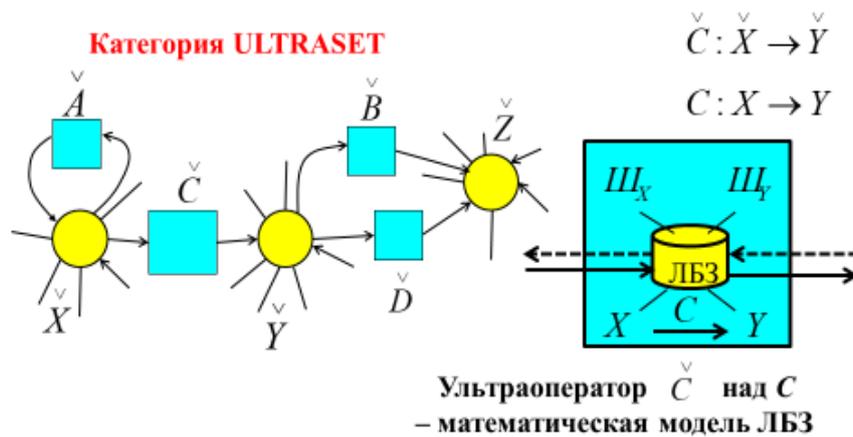


Ультрамножество \check{X} над X , математическая модель ЛБД триад вида $(d)P(x)$, $d \in D$, $P \in L$, $x \in X$

Ультраоснащенная опорная сеть – распределенная БД

Рис. 4. Развитие четвертого иерархического уровня АСПУ ИИ

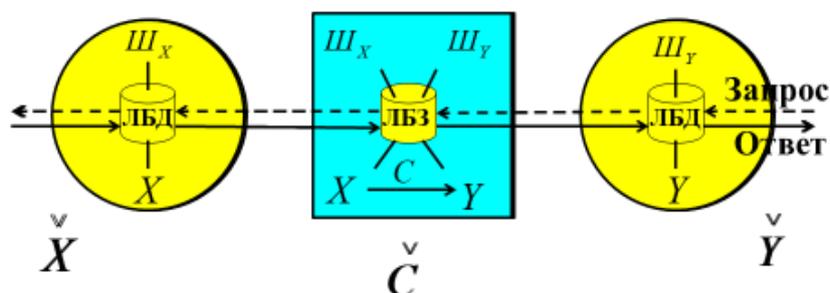
Ультраоснащение опорных операторов в распределенную БЗ ИИ



Ультраоснащенная опорная сеть – распределенная БД и БЗ, т.е. информационно-системная избыточность

Рис. 5. Завершение развития четвертого уровня ИИ до ядра ИИ.

Согласование ЛБД и ЛБЗ в ядре ИИ



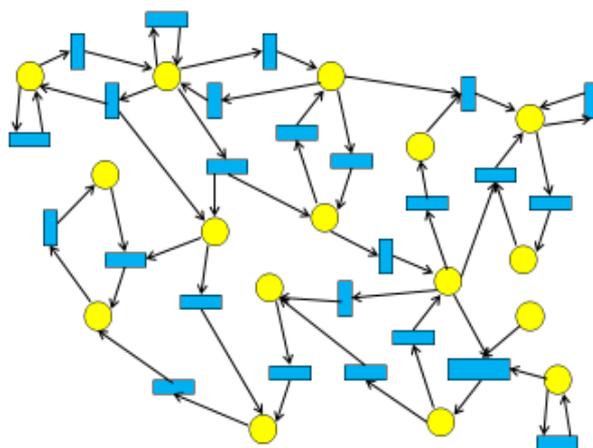
$$\begin{array}{l} \text{Ультрасистема} \quad \overset{\vee}{C} : \overset{\vee}{X} \rightarrow \overset{\vee}{Y} \\ \text{Система} \quad C : X \rightarrow Y \end{array}$$

Рис. 6. Семантическое согласование в распределенных БД и БЗ ядра АСПУ ИИ

знаний (ЛБЗ) в АСПУ ИИ решаются средствами машинного обучения, прогнозирования и принятия решений в рамках распределенного информационно-системного ресурса АСПУ в форме среды радикалов-ЛБД и радикалов-ЛБЗ, формируется *категория ULTRASET*, рис.5. На практике у человека в жизни ядро естественного интеллекта связано с внешней бумажной и электронной формой разнообразных частных профессиональных знаний и технологий (в учебниках), на основе которых обучают специалистов в средних и высших учебных заведениях, рис. 6.

ВЫВОД. На четвертом уровне развития АСПУ ИИ оформляется языковая информационная радикальная модель сенсорного мира АСПУ ИИ в виде распределенной БД и распределенной БЗ. Такая модель является ядром АСПУ ИИ, рис.5, в котором каждый радикал автономен и семантически согласован с соседними радикалами, рис.6, рис.7.

Ядро ИИ как распределенная БД и БЗ ИИ



Согласованная радикальная модель мира ИИ в рамках своего театра действий

Рис.5.

Ядро АСПУ ИИ - модель видимой части мира АСПУ ИИ, отражающая осознанную часть видимого только штатными сенсорами первичного сенсорiums АСПУ ИИ.

Замечание: модель сенсорного мира АСПУ ИИ или ядро АСПУ ИИ – это только сознательная часть, осознанная часть, самой «Я-АСПУ» и только осознанная часть, театра действий АСПУ ИИ. Большая часть первичного сенсорiums не охвачена языком, остается в подсознательной, неосознанной форме. Отметим еще, что ядро АСПУ ИИ – это только модель сенсорного мира АСПУ ИИ, видимого имеющимися штатными для АСПУ ИИ сенсорами, средствами. На рис. 7 изображена только сенсорная часть ядра АСПУ ИИ, доступная и обслуживаемая имеющимися наличными (штатными) сенсорами АСПУ ИИ. Например, в ядре АСПУ ИИ нет микро и макро виртуальных уровней ядра АСПУ ИИ, доступных специальным дополнительным измерительным инструментам, микро- и макро-сенсорам АСПУ ИИ, и соответствующим им математическим моделям, рис.7.

5. Пятый когнитивный иерархический уровень ультрасистемы АСПУ ИИ, [10] – это языковая операционная система интеллектуального планирования и ситуационного управления действиями АСПУ ИИ в рамках своего театра действий с условием обязательного и эффективного выполнения своего общего ТТЗ и

целевых указаний автономных ТТЗ со стороны человека – хозяина, рис. 2.

Ультрасистема АСПУ ИИ, на своем когнитивном, верхнем, пятом уровне отвечает за *адаптацию к любым изменениям*, к нештатности, за *планирование и активирование системоквантов ядра АСПУ ИИ* для эффективного управления функционированием АСПУ ИИ в рамках своего театра действий. При этом языковая операционная система планирования и управления поведением АСПУ на основе радикальной модели АСПУ ИИ осуществляет одновременно три следующих информационных управляющих процесса: 1) *Текущий целевой* - решение очередной тактической задачи во исполнение целевого указания ТТЗ человека-хозяина. Правило: «целенаправленность поведения АСПУ ИИ»; 2) *текущий сенсорный мониторинг* - непрерывный сбор ситуационной информации. Правило: «что здесь и сейчас» и «что нового появилось и как отражено в ядре АСПУ ИИ и, что в глобальном виртуальном информационно-системном пространстве»; 3) *текущий сертификационный процесс* – обеспечение достаточной целостности, гомеостаза ядра АСПУ ИИ, самой «Я-АСПУ» и всего театра действий АСПУ ИИ. Правило: «заккрытие гештальтов», рис.8.

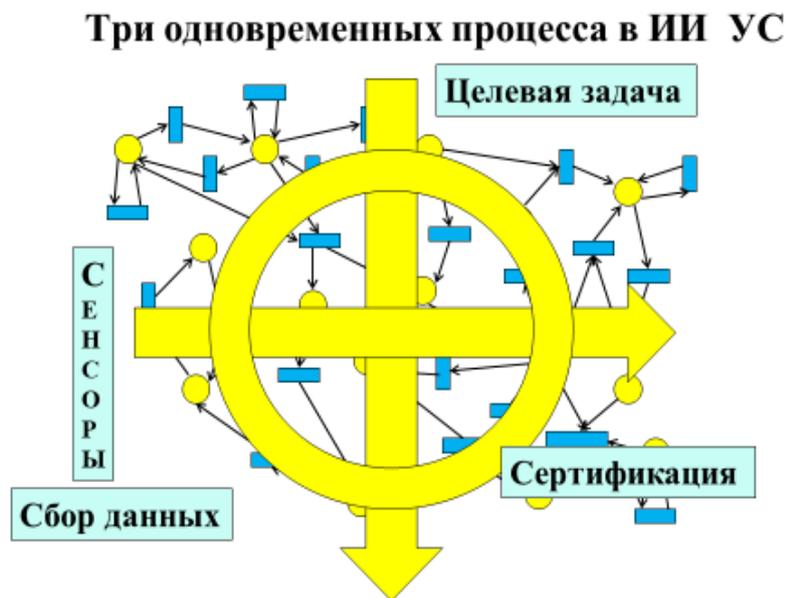


Рис.6.
Управляемые процессы, *целевой, сенсорный и сертификационный*, в языковой радикальной единой модели сенсорного мира АСПУ ИИ

6. Заключение. Техносфера АСПУ ИИ

В ноосфере Земли, с появлением искусственных технических умных систем с элементами ИИ и признанием обязательной ведущей роли человека в социуме Земли появляется новый революционный системный эффект, **техносфера АСПУ ИИ**. Вот некоторые виртуальные стороны новой техносферы.

Представим себе близкое будущее, когда во дворах жилых домов, на обочинах наших дорог, на площадках у магазинов и офисов, у мест отдыха совсем не будет стоянок припаркованных личных автомобилей, которые часами ожидают своих хозяев. Автомобили будут только при делах, в движении. Они будут умными беспилотниками, будут без водителей, но с эффективными элементами ИИ. Все они будут в подчинении своих АСПУ ИИ с многокритериальной оптимизацией сервисных массовых заказов людей. Им не требуется стоять в ожидании своих пассажиров. Каждый человек всегда может вызвать себе беспилотный автомобиль требуемого достоинства и в любое время.

Другой виртуальный пример картины будущего. Любая сложная техническая, социальная, организационная или биологическая система, включая человека будет снабжена и оберегаема цифровыми двойниками контроля, ремонтного сервиса и эффективного поддержания данной системы в нормативном состоянии. Например, человек будет жить долгие годы в активном бодром жизнедеятельном состоянии. Такая же новая картина жизни людей в окружении умных помощников будет на Земле, в космосе, в логистике, в любом сервисном обслуживании, в медицине, в любом транспорте, на любом производстве, в спорте и т.д. и т.п. Сейчас даже не все новое можно себе представить, но оно идет. Приведем в заключении слова академика В.И. Вернадского: «Научное мировоззрение, проникнутое естествознанием и математикой, есть величайшая сила не только настоящего, но и будущего» [1]. По-видимому, придёт время единого для техники и человека делового языка интеллектуального общения, *радикального языка интерфейса глобального социума сетевой техносферы АСПУ ИИ*.

Список литературы

- [1] В.И. Вернадский, *Биосфера и ноосфера*, Наука, М., 1989.
- [2] И.П. Павлов, «Лекции о работе больших полушарий головного мозга. Полное собрание трудов в 5-и т.: Т.1, 2, 5.», 1973.
- [3] А.В. Чечкин, «Математическая информатика», 1991.

- [4] ВИКИПЕДИЯ. Generative pre-trained transformer. СНАТ-GPT.ru.
- [5] Городецкий В.И., Карсаев О. В., Самойлов В. В., Серебряков С.В., “Инструментальные средства для открытых сетей агентов”, *Известия РАН. Теория и Системы Управления*, 2008, № 3, 106-124..
- [6] Г.И. Андреев, П.А. Созинов, В.А. Тихомиров, “Методология моделирования сложных технических систем. Основы системных исследований”, 2020.
- [7] П.А. Созинов, *Цифровые двойники*, РАДИОТЕХНИКА, М., 2022.
- [8] Г.С. Воронков, А.В. Чечкин, “Проблемы моделирования сенсориума и языковой системы естественного интеллекта индивидуума”, *Интеллектуальные системы*, **2**:1-4 (1997), 35-54.
- [9] А.В. Чечкин, “Тезис о наличии искусственного интеллекта”, *Интеллектуальные системы*, **25**:1 (2021), 29-49.
- [10] А.В. Чечкин, “Когнитивный уровень искусственного интеллекта”, *Интеллектуальные системы*, **26** (2022), 5-24.
- [11] В.А. Васенин, М.В. Пирогов, А.В. Чечкин, “Информационно-системная безопасность критических систем”, 2018, 348.
- [12] А.Ю. Потюпки, А.В. Чечкин, “Искусственный интеллект на базе информационно-системной избыточности”, 2019, 384.

Five principles multi-agent automated artificial intelligence systems

A.V. Chechkin

Goal. To investigate the effectiveness of the interaction of natural and artificial intelligence (AI) of the Earth’s noosphere.

Methods. Mathematical, informational and virtual modeling. System and ultrasystem analysis and synthesis.

Results. The necessity of using the concept of a multi-agent automated system as a very typical human-machine AI system of the Earth’s noosphere is substantiated. Five conceptual principles are formulated and substantiated: the principle of automation and human-centrism of AI; the principle of information system redundancy and security of AI; the principle of reasonable integrity of AI; the principle of abnormal goal setting of AI; the principle of five levels of language development of AI.

Key words: natural and artificial intelligence, automated system, multi-agent system, digital twins, intellectual planning and situational group management system, interface language.

References

- [1] V.I. Vernadsky, *The biosphere and the noosphere*, Nauka, M., 1989.
- [2] I.P. Pavlov, "Lectures on the work of the cerebral hemispheres. The complete collection of works in 5 volumes: Vol.1, 2, 5", 1973.
- [3] A.V. Chechkin, "Mathematical computer science", 1991.
- [4] WIKIPEDIA. Generative pre-trained transformer. CHAT-GPT.ru.
- [5] Gorodetsky V.I., Karsaev O. V., Samoilov V. V., Serebryakov S.V., "Tools for open agent networks", *Izvestia RAS. Theory and Control Systems*, 2008, №3, 106-124..
- [6] G.I. Andreev, P.A. Sozinov, V.A. Tikhomirov, "Methodology Modeling of complex technical systems. The basics of the system research", 2020.
- [7] P.A. Sozinov, *Digital doubles*, RADIO ENGINEERING, M., 2022.
- [8] G.S. Voronkov, A.V. Chechkin, "Problems of modeling the sensorium and the language system of the natural intelligence of the individual", *Intelligent Systems Magazine*, **2**:1-4 (1997), 35-54.
- [9] A.V. Chechkin, "The thesis about the presence of artificial intelligence", *Intelligent Systems*, **25**:1 (2021), 29-49.
- [10] A.V. Chechkin, "The cognitive level of artificial intelligence", *Intelligent Systems Magazine*, **26** (2022), 5-24.
- [11] V.A. Vasenin, M.V. Pirogov, A.V. Chechkin, "Information systems the security of critical systems", 2018, 348.
- [12] A.Y. Potyupkin, A.V. Chechkin, "Artificial intelligence based on information and system redundancy", 2019, 384.

Часть 2.
Специальные вопросы теории
интеллектуальных систем

Алгоритм распознавания эмоций на основе линейной регрессии

Е. С. Ковалёва¹

В работе предлагается метод распознавания эмоций на изображениях лиц при помощи линейной регрессии с использованием двух нейронных сетей (MediaPipe[1] и Dlib[2]), которые располагают ключевые точки на изображениях лиц людей. В данной статье будут показаны: создание базы данных для обучающей выборки, признаки, которые будут использоваться для обучения классификатора, построение линейного классификатора и результат его работы на тренировочной выборке. Целью работы является выявление ключевых признаков по ключевым точкам, по которым можно распознать ту или иную эмоцию. Предлагаемые подходы могут оказаться полезными, в частности, при обучении многослойных нейронных сетей, обучение которых повысит качество распознавания эмоций на изображениях.

Ключевые слова: ключевая точка, MediaPipe, Dlib, линейный классификатор.

1. Введение

Основной целью данной работы является разработка линейного классификатора для распознавания эмоций по фотографии человека.

Для её достижения были проделаны следующие этапы:

- создание базы данных для обучающей выборки из видео данных;
- применение нейронных сетей MediaPipe [1] и Dlib [2] для извлечения координат ключевых точек;
- создание векторов признаков, состоящих из расстояний и углов между ключевыми точками и характеризующих эмоции наилучшим образом;
- создание линейных классификаторов для отнесения каждого изображения к той или иной эмоции;
- обучение линейных классификаторов и поиск весовых коэффициентов;

¹ Ковалёва Елена Сергеевна — аспирантка каф. математической теории интеллектуальных систем мех.-мат. ф-та МГУ, e-mail: e.kovaleva.msu@yandex.ru.

Kovalyova Elena Sergeevna — graduate student, Lomonosov Moscow State University, Faculty of Mechanics and Mathematics, Chair of Mathematical Theory of Intellectual Systems.

- тестирование классификаторов на тренировочной выборке;
- тестирование алгоритма на локальную и общую точность распознавания.

Последующей целью является выявление слабых и сильных сторон разработанного алгоритма и построение нейронной сети по найденным в работе векторам признаков.

2. Создание базы данных

Для создания обучающей выборки необходимо создать базу изображений лиц людей с различными эмоциями. В данной работе речь пойдет о распознавании семи эмоций: радость, грусть, удивление, гнев, страх, отвращение, презрение. Поскольку для выявления признаков впоследствии будут использоваться нейронные сети MediaPipe и Dlib, то для того, чтобы эти нейронные сети могли благополучно извлечь координаты ключевых точек, необходимы изображения с разрешением большим, чем 480×480 пикселей. Поэтому была поставлена цель создать базу данных с изображениями высокого качества. Так как такой базы данных нет в открытом доступе, появилась необходимость создавать её вручную. Есть два способа, которыми можно было решить данную проблему:

- Скачивать отдельные изображения.
- Скачивать видео, которые позже можно будет нарезать на отдельные кадры, которые будут представлять собой изображения с лицами людей.

Было принято решение создать базу данных вторым способом, поскольку это оптимальнее по времени. Видео были собраны на ресурсе YouTube, так как данный сайт обладает не только безграничным количеством видеoinформации, но и качественной поисковой системой, позволяющей быстрее найти нужные для работы видео.

Видео были отобраны с учетом следующих параметров:

- Видео должно иметь качество 4k.
- Видео должно быть с участием людей.
- Лица людей на видео должны быть четкими и сняты крупным планом.
- В течение минимум минуты видео должно быть непрерывно по содержанию и повороту камеры.

- Видео должны содержать лица людей, выражающих все семь эмоций, которые необходимо будет впоследствии распознавать.

Количество изображений (кадров) из загруженных видео [3] для каждой эмоции было отрегулировано вручную. Для загрузки видео в высоком качестве был использован youtube-dl [4]. Итоговая база данных составила 523747 изображений.

3. Создание векторов признаков

3.1. Извлечение ключевых точек

Для создания вектора признаков для каждого изображения, применим к каждому изображению две нейронные сети MediaPipe и Dlib для извлечения координат ключевых точек. Эти две нейронные сети применяются для данной цели по причине большого числа преимуществ каждой из них.

MediaPipe содержит 468 ключевых точек и является самой подробной по количеству точек нейронной сетью. Эта нейронная сеть очень подробно описывает область носа, щёк и глаз. Пример работы данной нейронной сети изображен на рисунке 1, а маска с ключевыми точками - на рисунке 2.

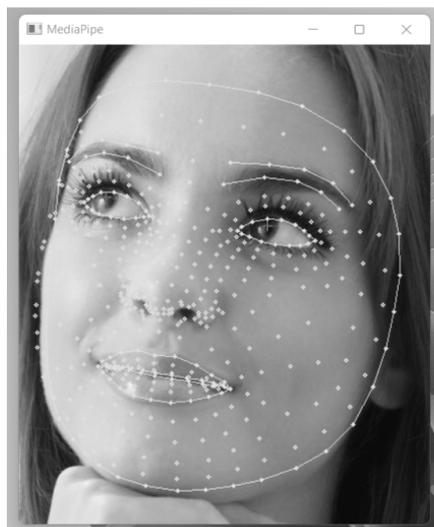


Рис. 1. Пример работы MediaPipe.

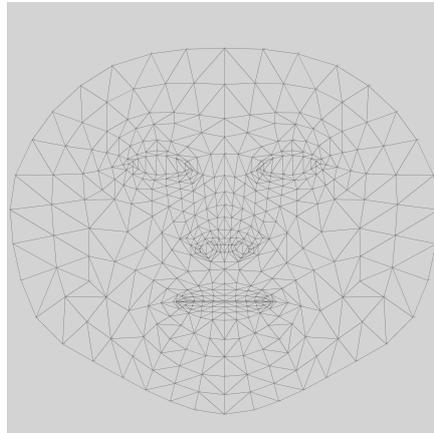


Рис. 2. Ключевые точки MediaPipe.

Преимущества Dlib в сравнении с MediaPipe заключаются в точности расположения ключевых точек на лицах. Визуальный анализ показал, что Dlib очень точно располагает на изображениях ключевые точки глаз, бровей, носа и рта. Однако, в отличие от MediaPipe он не умеет располагать ключевые точки на щеках, в области носа и в области лба. Именно по этой причине (по причине наличия преимуществ в каждой из этих двух нейронных сетей) было принято решение извлекать координаты ключевых точек, используя и MediaPipe и Dlib.

Пример работы Dlib на изображении показан на рисунке 3. На рисунке 4 продемонстрированы непосредственно сами ключевые точки.



Рис. 3. Пример работы Dlib.

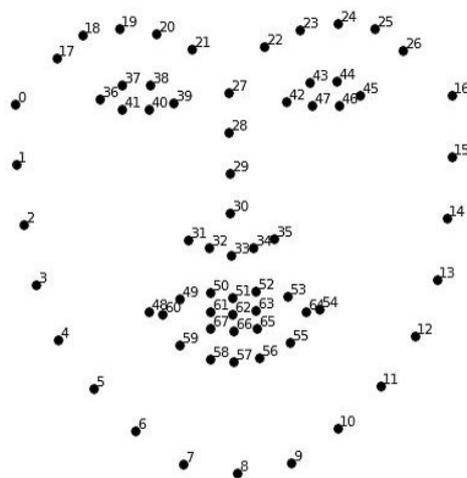


Рис. 4. Ключевые точки Dlib.

3.2. Вектор признаков

Визуальный анализ результатов работы MediaPipe и Dlib на изображениях показал, что положение точек в области глаз, рта, переносицы и щек влияют на показываемую человеком эмоцию. Однако для более точного

вывода необходимо понять, какие именно точки, расстояния и углы между ними влияют на изображение той или иной эмоции. Для этой цели необходимо создать векторы признаков, которые впоследствии будучи приняты в классификатор покажут насколько то или иное расстояние или тот или иной угол между ключевыми точками влияют на показанную на изображении эмоцию.

Для начала условно обозначим все точки от 0 до 467 из MediaPipe как M_0, M_1, \dots, M_{467} соответственно, а все точки от 0 до 67 из Dlib как D_0, D_1, \dots, D_{67} соответственно. Для входных значений классификатора понадобятся признаки в виде нормированных расстояний и углов (в радианах) между ключевыми точками и отношений этих углов друг к другу, изображенных на рисунке 5. Для того, чтобы нормировать расстояния, их необходимо разделить на длину отрезка $M_{127}M_{356}$. Нормировать расстояния необходимо, чтобы данные исследования не зависели от размера изображения и размера лица на изображении.

Таким образом, входная база данных для каждого изображения i состоит из вектора признаков x_i (включающего в себя нормированные расстояния отрезков и размер углов в радианах, изображенных на рисунке 5) и вектора вероятностей y_i ($y_{ij} = 0$, если на изображении i нет эмоции j , и $y_{ij} = 1$ в противном случае).

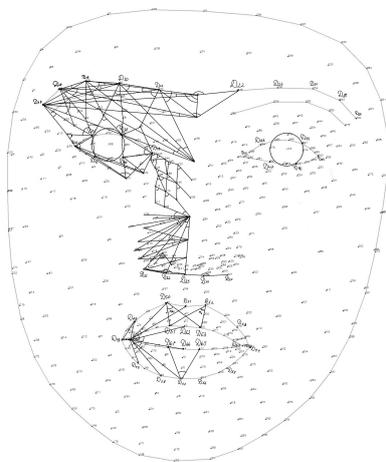


Рис. 5. Расстояния и углы для вектора признаков.

4. Линейная регрессия для распознавания эмоций

4.1. Математическая формулировка задачи

Для идентификации одной из семи эмоций необходимо построить семь классификаторов, определяющих вероятность изображения эмоций на фото, следующим образом:

- 1) Разделяем случайным образом базу данных изображений (из параграфа 2) на тестовую и тренировочную выборки по следующим правилам:
 - Размер тренировочной выборки $m_{train} = 366625$ (по 52375 на каждую эмоцию).
 - Размер тестовой выборки, соответственно, $m_{test} = 157122$ (по 22446 на каждую эмоцию).
- 2) Для каждого из m изображений:
 - Извлекаем признаки x_{ij} (согласно алгоритму, изложенному в параграфе 3.2), где i - номер признака, j - номер изображения. Строим матрицу характеристик $X = \{x_{ij}\}_{m_{train} \times n+1}$ с первым столбцом из единиц и с величинами признаков во всех остальных столбцах.
 - Строим вектор вероятностей y_j для каждого изображения j размера 7 (по количеству эмоций) таким образом, чтобы $y_{jt} = 1$, если на изображении есть эмоция под номером t , и $y_{jt} = 0$ в противоположном случае.
Пример: если на изображении вторая эмоция, то вектор будет выглядеть следующим образом: $y_j = (0, 1, 0, 0, 0, 0, 0)$.
- 3) Для каждого изображения i зададим уравнения следующего вида:
 $y_{it} = w_{0t} + w_{1t}x_{1t} + w_{2t}x_{2t} + \dots + w_{nt}x_{nt} + e_{it}$, где
 $w_{0t}, w_{1t}, \dots, w_{nt}$ - веса признаков, которые необходимо найти,
 e_{it} - погрешность модели,
 y_{it} - вероятность того, что на фото изображена эмоция под номером t ,
 t - номер эмоции.
- 4) Таким образом наша задача сводится к поиску решения следующих СЛАУ для каждой из 7 эмоций:
 $X_{m_{train} \times n} w_{n \times 1} + e_{m_{train} \times 1} = y_{m_{train} \times 1}$,
где w - вектор весов, e - вектор погрешностей, y - вектор вероятностей.

4.2. Поиск вектора весов методом наименьших квадратов

Построим решение для поиска вектора весов в классификаторе эмоции с номером t в тренировочной выборке. Для этой цели:

- 1) Строим СЛАУ $X_{m_{train} \times n} w_t + e_t = y_t$ для векторов y_{it} , где i - номер изображения.
- 2) Для решения данной СЛАУ применяем метод наименьших квадратов [5] и получаем решение для вектора весов:
$$w_t = (X^T X)^{-1} X^T y_t. \quad (1.1)$$
- 3) Решаем систему (1.1) для $t = 0, 1, \dots, 6$.
- 4) Находим значения весов признаков w_{it} для $i = 0, 1, \dots, m_{train} - 1$ и $t = 0, 1, \dots, 6$.

4.3. Применение классификатора на тестовой выборке

Теперь, когда веса признаков для каждой эмоции известны, можем провести эксперимент по работе классификатора на тренировочной выборке. Для этого на вход каждого классификатора подаются следующие данные:

- Матрица признаков с единичным первым столбцом:
 $X = \{x_{ij}\}_{m_{test} \times n+1}$, где $i = 0, 1, \dots, m_{test} - 1$ (номер изображения), $j = 0, 1, \dots, n$ (номер признака).
- Значения весов для весового вектора: w_{it} , где $i = 0, 1, \dots, m_{test} - 1$ (номер изображения), а $t = 0, 1, \dots, 6$ (номер классификатора).

Затем для каждого классификатора вычисляется произведение матрицы признаков и его вектора весов:

$f_t = X w_t$ - вектор вероятностей, показывающий, с какой вероятностью на фото изображена эмоция под номером t .

Например, если величина $f_{it} = 0.75$, то это означает, что на фото с номером i с вероятностью 0.75 изображена эмоция под номером t .

Таким образом, применив все 7 классификаторов ко всем изображениям тестовой выборки, на выходе алгоритма получаем матрицу вероятностей $F = \{f_{ij}\}_{m_{test} \times 7}$, где $i = 0, 1, \dots, m_{test} - 1$ (номер изображения), $t = 0, 1, \dots, 6$ (номер классификатора). После этого для каждого изображения $i = 0, 1, \dots, m_{test} - 1$ вычисляем:

$k_i = \arg t(\max_{t \in [0,6]}(f_{it}))$ - номер эмоции, которая изображена на фото с номером i с наибольшей вероятностью;

$p_i = \max_{t \in [0,6]}(f_{it})$ вероятность того, что на фото с номером i изображена эмоция k_i .

Таким образом вычисляется эмоция, показанная на изображении.

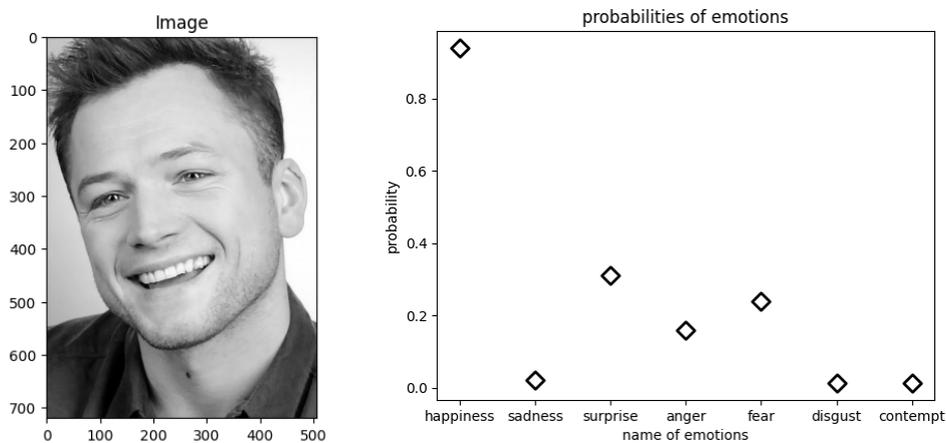


Рис. 6. Пример работы алгоритма на одном изображении. График вероятностей.

5. Примеры работы алгоритма и результаты

На рисунке 6 показан пример работы алгоритма на одном изображении из тестовой выборки. График отображает вычисленную каждым из семи классификаторов вероятность изображения на фотографии эмоции.

На данном примере можно заметить, что первый классификатор, отвечающий за вероятность эмоции радости дал наибольший результат среди всех классификаторов, что является правильным ответом.

На рисунке 7 приведен пример работы первого классификатора, отвечающего за вероятности эмоции радости, примененного ко всем изображениям тестовой выборки. На оси x расположены номера изображений. На оси y расположены вероятность идентификации эмоции радости на изображениях.

Вертикальные полосы на графике отделяют изображения, относящиеся к одной группе. Например, во втором столбце на оси x расположены номера изображений с эмоцией грусти.

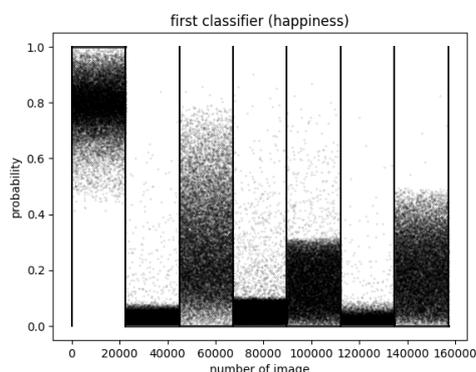


Рис. 7. Результат работы первого классификатора на тестовой выборке.

На рисунке 8 приведена таблица с долями изображений, в которых алгоритм семи классификаторов успешно и правильно распознал эмоции.

Emotion	Probability
happiness	0.87
sadness	0.93
surprise	0.86
anger	0.91
fear	0.94
disgust	0.98
contempt	0.95

Рис. 8. Доли правильно распознанных эмоций в тестовой выборке.

Общая доля ошибок составила во всей тестовой выборке составила 8%.

По результатам вероятностей успешного распознавания, приведенного в таблице на рисунке 8, можно заметить, что хуже всего распознаются эмоции радости и удивления, а эмоции отвращения и грусти распознаются лучше всего.

Данная аномалия связана с тем, что у некоторых пар эмоций существует большее количество общих признаков, чем у других пар эмоций. Например, на одном изображении могут быть одновременно эмоция удивления и эмоция радости. Также на одном изображении одновременно могут быть эмоции удивления и страха. В некоторых случаях трудно отличить

эмоцию презрения и эмоцию радости из-за наличия улыбки в обоих случаях. По этой причине лучше всего распознаются эмоции с более уникальным набором признаков, чем у других.

6. Выводы

В данной работе были проделаны следующие этапы:

- Создана база видеоданных.
- На её основе создана база данных с 523747 изображениями.
- Создана база входных данных с признаками на основе двух нейронных сетей (MediaPipe и Dlib).
- Математически поставлена задача распознавания эмоций с помощью классификаторов на основе линейной регрессии.
- Разработан алгоритм распознавания эмоций на основе линейной регрессии и метода наименьших квадратов.
- Алгоритм был обучен на тренировочной выборке изображений.
- Алгоритм был протестирован на тестовой выборке, включающей в себя 157122 изображения.
- Были вычислены вероятности правильных ответов и общая доля ошибки алгоритма.
- Был вычислен успех работы алгоритма, который составил 92%.

В дальнейшем планируется построение многослойной нейронной сети для достижения более высокого результата правильно распознанных изображений.

Список литературы

- [1] Руководство по использованию MediaPipe,
https://github.com/googlesamples/mediapipe/tree/main/examples/face_landmarker/python.
- [2] Руководство по использованию Dlib,
<http://dlib.net/python/index.html>.

- [3] Плейлист с видео для обучающей выборки, <https://www.youtube.com/playlist?list=PLbvXU2Yespkgvzwk400999CKrVV1PZYCI>.
- [4] Руководство по использованию youtube-dl, <https://github.com/ytdl-org/youtube-dl/blob/master/README.md#readme>.
- [5] Линник Ю.В., “Метод наименьших квадратов и основы математико-статистической теории обработки наблюдений”, *Государственное издательство Физико-математической литературы*, 1958.

Emotion recognition algorithm based on linear regression Kovalyova E.S.

The paper proposes a method for recognizing emotions in facial images using linear regression using two neural networks (MediaPipe[1] and Dlib[2]), which locate key points in images of people’s faces. This article will show: creating a database for the training set, features that will be used to train the classifier, building a linear classifier and the result of its work on the training set. The goal of the work is to identify key features at key points by which one or another emotion can be recognized. The proposed approaches may be useful, in particular, when training multilayer neural networks, the training of which will improve the quality of recognition of emotions in images.

Keywords: keypoint, MediaPipe, Dlib, linear classifier.

References

- [1] MediaPipe user manual, https://github.com/googlesamples/mediapipe/tree/main/examples/face_landmarker/python.
- [2] Dlib user manual, <http://dlib.net/python/index.html>.
- [3] Playlist with videos for the training set, <https://www.youtube.com/playlist?list=PLbvXU2Yespkgvzwk400999CKrVV1PZYCI>.
- [4] Youtube-dl user manual, <https://github.com/ytdl-org/youtube-dl/blob/master/README.md#readme>.
- [5] Linnik Yu.V., “The least squares method and the fundamentals of the mathematical and statistical theory of observation processing (In Russian)”, *State Publishing House of Physical and Mathematical Literature*, 1958.

О единственности решения одной системы полиномиальных уравнений над конечным полем

И.С. Фаерштейн¹

Рассматривается система полиномиальных уравнений специального вида над полем $GF(2^m)$ и исследуется единственность её решения с точностью до перестановки переменных.

Ключевые слова: система полиномиальных уравнений, конечные поля, единственность решения.

1. Введение

Пусть задан некоторый класс функций K . Будем говорить, что функция f , зависящая от того же множества переменных, что и функции из класса K , порождает функцию g (при условии $g \in K$), если существует такое множество точек X , что $g(x)$ является единственной функцией, принадлежащей классу K и удовлетворяющей соотношению $f(x) = g(x)$ для любого x из множества X . Функция f называется универсальной для класса K , если она порождает любую функцию из данного класса [1].

При решении задачи о построении универсальной функции класса линейных функций, существенно зависящих от фиксированного числа переменных, была поставлена вспомогательная задача исследования систем полиномиальных уравнений определенного вида над полем $GF(2^m)$ на единственность имеющегося решения, состоящего из разных элементов, с точностью до перестановки переменных [2, 3]. Данные системы уравнений представляют собой равенства параметрам сумм нечетных степеней переменных.

Для систем уравнений вида

$$\begin{cases} x_1 + x_2 = a, \\ x_1^3 + x_2^3 = b. \end{cases}$$

¹Фаерштейн Игорь Семенович — аспирант кафедры математической кибернетики факультета ВМК МГУ, e-mail: isfaer@rambler.ru.

Faershtein Igor Semenovich — postgraduate student, Lomonosov Moscow State University, Faculty of Computer Science, Chair of Mathematical Cybernetics.

и

$$\begin{cases} x_1 + x_2 + x_3 + x_4 = a, \\ x_1^3 + x_2^3 + x_3^3 + x_4^3 = b, \\ x_1^5 + x_2^5 + x_3^5 + x_4^5 = c, \\ x_1^7 + x_2^7 + x_3^7 + x_4^7 = d, \end{cases}$$

где $a, b, c \in GF(2^m)$ – константы, $x_1, x_2, x_3 \in GF(2^m)$, $x_i \neq x_j$ при $i \neq j$ – попарно различные переменные.

2. Основные результаты

Рассмотрим систему уравнений над полем Галуа $GF(2^m)$:

$$\begin{cases} x_1 + x_2 + x_3 + x_4 = a, \\ x_1^3 + x_2^3 + x_3^3 + x_4^3 = b, \\ x_1^5 + x_2^5 + x_3^5 + x_4^5 = c, \\ x_1^7 + x_2^7 + x_3^7 + x_4^7 = d, \end{cases} \quad (1)$$

где $a, b, c, d \in GF(2^m)$ – константы, $x_1, x_2, x_3, x_4 \in GF(2^m)$ – переменные.

Решение (x'_1, x'_2, x'_3, x'_4) системы (1) назовём невырожденным, если оно не содержит равных элементов. Решения системы (1), получающиеся друг из друга перестановкой переменных, называются эквивалентными. Будем говорить, что в системе (1) имеется коллизия, если в ней есть два неэквивалентных невырожденных решения. Невырожденное решение называется нулевым (единичным), если оно содержит нулевой (единичный) элемент.

В системе (1) коллизия невозможна.

Доказательство. Выразим уравнения системы (1) через элементарные симметрические многочлены и будем пытаться выразить элементарные симметрические многочлены через правые части системы (1). Рассмотрим уравнение

$$x^4 + s_1x^3 + s_2x^2 + s_3x + s_4 = 0, \quad (2)$$

где элементарные симметрические многочлены s_1, s_2, s_3, s_4 выражены через правые части системы (1). По обратной обобщённой теореме Виета [4] корнями уравнения (2) являются переменные x_1, x_2, x_3, x_4 системы (1). Поскольку уравнение степени 4 может иметь в поле не более 4 корней [5], то система (1) не может иметь невырожденные решения и, как следствие, не может иметь коллизии.

После выражения уравнений системы (1) через элементарные симметрические многочлены получим следующую систему уравнений:

$$\begin{cases} s_1 = a, \\ s_1^3 + s_1 s_2 + s_3 = b, \\ s_1^5 + s_1^3 s_2 + s_1 s_2^2 + s_1^2 s_3 + s_2 s_3 + s_1 s_4 = c, \\ s_1^7 + s_1^5 s_2 + s_1 s_2^3 + s_1^4 s_3 + s_1^2 s_2 s_3 + s_2^2 s_3 + s_1 s_3^2 + s_1^3 s_4 + s_3 s_4 = d, \end{cases} \quad (3)$$

где

$$\begin{aligned} s_1 &= x_1 + x_2 + x_3 + x_4, \quad s_2 = x_1 x_2 + x_1 x_3 + x_1 x_4 + x_2 x_3 + x_2 x_4 + x_3 x_4, \\ s_3 &= x_1 x_2 x_3 + x_1 x_2 x_4 + x_1 x_3 x_4 + x_2 x_3 x_4, \quad s_4 = x_1 x_2 x_3 x_4. \end{aligned}$$

Лемма 1. Пусть $a = 0, b \neq 0$. Тогда в системе (1) коллизия невозможна.

Доказательство. Пусть $a = 0$. Тогда система уравнений (3) принимает вид:

$$\begin{cases} s_1 = 0, \\ s_1^3 + s_1 s_2 + s_3 = b, \\ s_1^5 + s_1^3 s_2 + s_1 s_2^2 + s_1^2 s_3 + s_2 s_3 + s_1 s_4 = c, \\ s_1^7 + s_1^5 s_2 + s_1 s_2^3 + s_1^4 s_3 + s_1^2 s_2 s_3 + s_2^2 s_3 + s_1 s_3^2 + s_1^3 s_4 + s_3 s_4 = d. \end{cases}$$

Подставим всюду $s_1 = 0$ и получим:

$$\begin{cases} s_3 = b, \\ s_2 s_3 = c, \\ s_2^2 s_3 + s_3 s_4 = d. \end{cases}$$

Откуда с учетом условия $b \neq 0$ имеем:

$$\begin{cases} b s_2 = c, \\ b s_2^2 + b s_4 = d. \end{cases}$$

$$\begin{cases} s_2 = b^{-1} c, \\ s_4 = b^{-2} c^2 + b^{-1} d. \end{cases}$$

$$\begin{cases} s_1 = 0, \\ s_2 = b^{-1} c, \\ s_3 = b, \\ s_4 = b^{-2} c^2 + b^{-1} d. \end{cases}$$

Элементарные симметрические многочлены однозначно выражаются через правые части системы уравнений, что означает невозможность неэквивалентных решений и, как следствие, невозможность коллизии. \square

Лемма 2. Пусть $a = 0, b = 0$. Тогда в системе (1) коллизия невозможна.

Доказательство. Пусть $a = 0, b = 0$. Тогда система уравнений (3) примет вид:

$$\begin{cases} a = b = c = d = 0, \\ s_1 = 0, \\ s_3 = 0, \\ x_1 + x_2 + x_3 + x_4 = 0, \\ x_1x_2x_3 + x_1x_2x_4 + x_1x_3x_4 + x_2x_3x_4 = 0. \end{cases}$$

$$\begin{cases} s_1 = 0, \\ s_1^3 + s_1s_2 + s_3 = 0, \\ s_1^5 + s_1^3s_2 + s_1s_2^2 + s_1^2s_3 + s_2s_3 + s_1s_4 = c, \\ s_1^7 + s_1^5s_2 + s_1s_2^3 + s_1^4s_3 + s_1^2s_2s_3 + s_2^2s_3 + s_1s_3^2 + s_1^3s_4 + s_3s_4 = d. \end{cases}$$

Подставим всюду $s_1 = 0$ и получим:

$$\begin{cases} s_3 = 0, \\ s_2s_3 = c, \\ s_2^2s_3 + s_3s_4 = d. \end{cases}$$

Подставим всюду $s_3 = 0$ и получим:

$$\begin{cases} c = 0, \\ d = c, \end{cases}$$

Таким образом, в этом случае всегда

$$\begin{cases} a = b = c = d = 0, \\ s_1 = 0, \\ s_3 = 0, \\ x_1 + x_2 + x_3 + x_4 = 0, \\ x_1x_2x_3 + x_1x_2x_4 + x_1x_3x_4 + x_2x_3x_4 = 0. \end{cases}$$

Выразим s_1 и s_3 через исходные переменные системы. Имеем

$$\begin{cases} x_1 + x_2 + x_3 + x_4 = 0, \\ x_1x_2x_3 + x_1x_2x_4 + x_1x_3x_4 + x_2x_3x_4 = 0. \end{cases}$$

Пусть $x_1 = \alpha, x_2 = \beta, \alpha \neq \beta$. (Иначе получается вырожденное решение). Тогда

$$\begin{cases} \alpha + \beta + x_3 + x_4 = 0, \\ \alpha\beta x_3 + \alpha\beta x_4 + \alpha x_3x_4 + \beta x_3x_4 = 0. \end{cases} \quad (4)$$

Выразим $x_3 + x_4$. $x_3 + x_4 = \alpha + \beta$ и подставим во второе уравнение системы (4). Получим $\alpha\beta(\alpha + \beta) + (\alpha + \beta)x_3x_4 = 0$. Так как $\alpha \neq \beta$, то $\alpha + \beta \neq 0$. Тогда $x_3x_4 = \alpha\beta$. Имеем:

$$\begin{cases} x_3 + x_4 = \alpha + \beta, \\ x_3x_4 = \alpha\beta. \end{cases}$$

Переменные x_3 и x_4 однозначно выражаются через свои сумму и произведение, то есть

$$\begin{cases} x_3 = \alpha, \\ x_4 = \beta. \end{cases}$$

или

$$\begin{cases} x_3 = \beta, \\ x_4 = \alpha. \end{cases}$$

То есть система (1) может иметь только вырожденные решения, а поэтому не может иметь коллизий. \square

Лемма 3. Пусть $a^5 + a^2b + a^{-1}b^2 + c \neq 0$. Тогда в системе (1) коллизия невозможна.

Доказательство. Так как выражение $a^5 + a^2b + a^{-1}b^2 + c$ имеет смысл, то $a \neq 0$. Подставим значение $s_1 = a$ в остальные уравнения системы (3) и получим систему уравнений:

$$\begin{cases} a^3 + as_2 + s_3 = b, \\ a^5 + a^3s_2 + as_2^2 + a^2s_3 + s_2s_3 + as_4 = c, \\ a^7 + a^5s_2 + as_2^3 + a^4s_3 + a^2s_2s_3 + s_2^2s_3 + as_2^2 + a^3s_4 + s_3s_4 = d. \end{cases} \quad (5)$$

Выразим s_3 . $s_3 = a^3 + as_2 + b$ и подставим в остальные уравнения системы (5). Получим систему уравнений:

$$\begin{cases} a^2b + a^3s_2 + bs_2 + as_4 = c, \\ a^4b + a^5s_2 + a^2bs_2 + bs_2^2 + a^7 + a^3s_2^2 + ab^2 + as_2s_4 + bs_4 = d. \end{cases} \quad (6)$$

Так как $a \neq 0$, выразим s_4 . $s_4 = ab + a^2s_2 + a^{-1}bs_2 + a^{-1}c$. Подставим значение s_4 в оставшееся уравнение системы (6) и получим

$$s_2(a^5 + a^2b + a^{-1}b^2 + c) = d + a^4b + a^7 + a^{-1}bc.$$

Так как $a^5 + a^2b + a^{-1}b^2 + c \neq 0$, то

$$s_2 = (a^5 + a^2b + a^{-1}b^2 + c)^{-1}(d + a^4b + a^7 + a^{-1}bc).$$

Тогда

$$\begin{cases} s_3 = a^3 + a(a^5 + a^2b + a^{-1}b^2 + c)^{-1}(d + a^4b + a^7 + a^{-1}bc) + b, \\ s_4 = ab + (a^2 + a^{-1}b)(a^5 + a^2b + a^{-1}b^2 + c)^{-1}(d + a^4b + a^7 + a^{-1}bc) + a^{-1}c. \end{cases}$$

Таким образом,

$$\begin{cases} s_1 = a, \\ s_2 = (a^5 + a^2b + a^{-1}b^2 + c)^{-1}(d + a^4b + a^7 + a^{-1}bc), \\ s_3 = a^3 + a(a^5 + a^2b + a^{-1}b^2 + c)^{-1}(d + a^4b + a^7 + a^{-1}bc) + b, \\ s_4 = ab + (a^2 + a^{-1}b)(a^5 + a^2b + a^{-1}b^2 + c)^{-1}(d + a^4b + a^7 + a^{-1}bc) + a^{-1}c. \end{cases}$$

Элементарные симметрические многочлены однозначно выражаются через правые части системы уравнений, что означает невозможность неэквивалентных решений и, как следствие, невозможность коллизии. \square

Лемма 4. Пусть для правых частей системы (1) выполнены соотношения $a \neq 0$ и

$$a^6 + a^3b + b^2 + ac = 0. \quad (7)$$

Тогда в системе (1) коллизия невозможна.

Доказательство. Сделаем в условии (7) замены на левые части системы (1). После приведения подобных членов получим условие

$$\sum_{i \neq j, i \neq k, j \neq k} x_i^3 x_j^2 x_k = 0. \quad (8)$$

Пусть невырожденное решение является нулевым. Тогда в равенстве (8) останется шесть слагаемых из двадцати четырех. Пусть $x_4 = 0$. Вынесем в (8) произведение $x_1 x_2 x_3$ и сократим на него. Имеем

$$\sum_{i \neq j} x_i^2 x_j = 0. \quad (9)$$

Сделав в (9) замену $x_1 = x, x_2 = xp, x_3 = xq$, и сократив на x^3 , получим

$$p(1+p) + q(1+q) + pq(p+q) = 0$$

или

$$(p+q)(p+1)(q+1) = 0.$$

Последнее противоречит невырожденности решения.

Заметим, что если неэквивалентные решения (x'_1, x'_2, x'_3, x'_4) и $(x''_1, x''_2, x''_3, x''_4)$ являются невырожденными, то при $\mu \neq 0$ наборы $(x'_1\mu, x'_2\mu, x'_3\mu, x'_4\mu)$ и $(x''_1\mu, x''_2\mu, x''_3\mu, x''_4\mu)$ являются решениями системы

$$\begin{cases} x_1 + x_2 + x_3 + x_4 = a\mu, \\ x_1^3 + x_2^3 + x_3^3 + x_4^3 = b\mu^3, \\ x_1^5 + x_2^5 + x_3^5 + x_4^5 = c\mu^5, \\ x_1^7 + x_2^7 + x_3^7 + x_4^7 = d\mu^7, \end{cases}$$

Положив при этом $\mu = (x'_4)^{-1}$, получим, что из наличия невырожденной коллизии в системе (1) следует наличие невырожденной единичной коллизии для некоторой системы, полученной из (1) изменением правых частей уравнений.

Докажем невозможность выполнения условия (7) в системе (1) при наличии единичного решения. Пусть $x_4 = 1$. Сделаем в (8) замену

$$x_1 = z, x_2 = zp, x_3 = zq. \quad (10)$$

При делении на z^3 условие (8) принимает вид

$$\begin{aligned} & z^3(p^3q^2 + p^2q^3 + p^3q + pq^3 + p^2q + pq^2) + z^2(p^3q^2 + p^2q^3 + p^3 + q^3 + p^2 + q^2) + \\ & + z(p^3q + pq^3 + p^3 + q^3 + p + q) + (p^2q + pq^2 + p^2 + q^2 + p + q) = 0. \end{aligned}$$

Разделив левую часть последнего равенства на $(p+q)(p+1)(q+1)$, получим

$$z^3pq + z^2(pq + p + q) + z(p + q + 1) + 1 = 0.$$

Сделав в последнем условии замену, обратную (10), и сгруппировав слагаемые, получим

$$(x_1 + 1)(x_2 + 1)(x_3 + 1) = 0.$$

Последнее соотношение противоречит предположению невырожденности единичного решения при $x_4 = 1$. Лемма доказана. \square

Таким образом, во всех случаях коллизия невозможна, что доказывает теорему. \square

Автор выражает благодарность профессору Вороненко А.А. за постановку задачи.

Список литературы

- [1] Вороненко А.А. Об универсальных частичных функциях для класса линейных функций. // Дискретная математика. Т. 24. Вып. 3. 2012. С. 62–65.
- [2] Вороненко А.А., Окунева А.С. Универсальные функции для классов линейных функций двух переменных. // Дискретная математика. Т. 32. Вып. 1. 2020. С. 3–7.
- [3] Вороненко А.А., Окунева А.С. Универсальные функции для классов линейных функций трех переменных. // Прикладная математика и информатика. – Т. 51. – М.: Макс Пресс, 2020. С. 114–121.
- [4] Винберг Э.Б. Алгебра многочленов. Учебное пособие для студентов-заочников III-IV курсов физико-математических факультетов педагогических институтов. – М.: Просвещение, 1980. 176 с.
- [5] Винберг Э.Б. Курс алгебры. 2-е изд., стереотип. – М.: МЦНМО, 2013. 590 с.

On the uniqueness of the solution of a system of polynomial equations over a finite field

Faershtein I.S.

A system of polynomial equations of a special form over the field $GF(2^m)$ is considered and the uniqueness of its solution up to a permutation of variables is investigated.

Keywords: system of polynomial equations, finite fields, uniqueness of solution.

References

- [1] Voronenko A.A. On universal partial functions for a class of linear functions. // Discrete Mathematics. Vol. 24. Issue. 3. 2012. P. 62–65. (In Russian).
- [2] Voronenko A.A., Okuneva A.S. Universal functions for classes of linear functions of two variables. // Discrete Mathematics. Vol. 32. Issue. 1. 2020. P. 3–7. (In Russian).
- [3] Voronenko A.A., Okuneva A.S. Universal functions for classes of linear functions of three variables. // Applied Mathematics and Computer Science. – Vol. 51. – Moscow: Max Press, 2020. P. 114–121. (In Russian).

- [4] Vinberg E.B. Algebra of polynomials. Textbook for part-time students of the III-IV courses of physics and mathematics faculties of pedagogical institutes. – Moscow Prosveshchenie, 1980. 176 p. (In Russian).
- [5] Vinberg E.B. Course of algebra. 2nd ed., stereotype. – Moscow: MCCME, 2013. 590 p. (In Russian).

Проблема оценки позы человека: задачи, методы, решения

И. А. Шергин¹, А. П. Рыжов²

В статье рассматривается проблема оценки позы человека, возникающая во многих приложениях систем компьютерного зрения. Рассмотрены различные постановки задач и подходов к решению проблемы, приводится обзор существующих методов решения и наборов данных. Дополнительно исследуется применение методов в задаче обучения физическим навыкам.

Ключевые слова: машинное обучение, глубокое обучение, оценка позы человека, компьютерное зрение.

1. Введение

Компьютерное зрение - это быстро развивающаяся область компьютерных наук. Человеческое тело является одним из самых непредсказуемых элементов для этой области.

Оценка позы человека (*Human pose estimation, HPE*) - это подобласть компьютерного зрения которая, позволяет идентифицировать и прогнозировать положение и ориентацию человеческого тела на изображении или видео. Последние достижения в развитии сверточных нейронных сетей значительно ускорили прогресс в этой области.

Различные процессы жизнедеятельности человека, такие как ходьба, бег, прыжки, танцы, или иные физические упражнения являются объектами изучения систем оценки позы человека - Рис. 1.

Целью данной работы является обзор задач в области Human Pose Estimation, методов их решения, и реализованных систем. Работа организована следующим образом: в разделе 1 приводится общая постановка задачи HPE; в разделе 2 обсуждаются вариации оценки позы человека;

¹Шергин Иван Андреевич — аспирант кафедры интеллектуальных информационных технологий ф-та ВМК МГУ, e-mail: i.a.shergin@gmail.com.

Shergin Ivan Andreevich — graduate student, Lomonosov Moscow State University, Faculty of Computational Mathematics and Cybernetics, Chair of Intelligent Information Technology.

²Рыжов Александр Павлович — доктор технических наук, профессор кафедры интеллектуальных информационных технологий ф-та ВМК МГУ, e-mail: ryjov@mail.ru.

Rylov Alexander Pavlovich — Sc.D. in engineering, Ph.D. in mathematics, MBA, professor, Moscow State University, Faculty of Computational Mathematics and Cybernetics, Chair of Intelligent Information Technology.



Рис. 1. Пример моделирования и оценки позы человека

раздел 3 посвящен анализу метрик качества, известных наборов данных и методов решения задач.

Обзор некоторых приложений систем НРЕ с упором на освоение физических навыков представлен в разделе 4; здесь же обсуждается задача персонализации обучения физическим навыкам с использованием методов оценки позы человека.

1.1. Постановка задачи

Целью оценки позы человека является определение расположения всех частей тела человека, учитывая отношения друг с другом и окружающей средой вокруг. Human pose estimation - это процесс выявления и классификации суставов человеческого тела и определения того, являются ли связи между этими суставами корректными. Другими словами, это процесс определения ключевых точек и достоверных пар связей между ними.

Ключевая точка - это 2 или 3-размерная координата в пространстве, описывающая расположение анатомического сустава, а также которая может быть использована для описания позы человека. Эти точки соединяются, образуя пары, при этом парой становится только существующее соединение. Связь, которая образуется между точками, должна быть осмысленной: любые две случайные точки не могут образовать пару.

Таким образом, методам НРЕ необходимо получить представление о человеческом теле подобное скелету, и затем использовать структуру для обработки человеческого тела для конкретных приложений. Именно это дает компьютерам возможность лучше понимать поведение человека.

1.2. Области применения

Уже существует ряд современных решений по применению технологий НРЕ в области фитнеса, реабилитации, физиотерапии, захвата движения, продвинутых игр, робототехники, видеонаблюдения и др.

1.2.1. Персональные тренеры на базе искусственного интеллекта

НРЕ может помочь поддерживать физическое самочувствие и достичь желаемого уровня физической подготовки без хорошего тренера. Пользователи могут открыть мобильное приложение, которое использует камеру для определения позы человека и оценивает, насколько точна поза во время определенного упражнения. Приложение может сообщить пользователю, что было сделано не правильно, и дать рекомендацию, как лучше тренироваться.

Например, Zenia [1] - это приложение для йоги на базе искусственного интеллекта, которое использует GPS, чтобы направлять пользователя к достижению правильной осанки во время занятий йогой.

1.2.2. Реабилитация и физиотерапия

Еще одной областью, где оценка позы человека может быть полезна, является физиотерапевтия. Это одно из наиболее перспективных приложений искусственного интеллекта для оценки позы в секторе здравоохранения. Цифровые консультации на дому повышают гибкость и доступность для пациента, а также производительность терапевта. Технологии ИИ [2] позволяют использовать новые и более сложные способы доставки лечения в цифровом виде.

1.2.3. Захват движения и дополненная реальность

Оценка позы человека была включена в виртуальную и дополненную реальность, например, в игры или фильмы для захвата движения. НРЕ может автоматически извлекать ключевые точки и визуализировать их для создания компьютерной графики. Использование данной технологии в AR и VR позволяет более естественно взаимодействовать с виртуальной средой.

1.2.4. Определение позы спортсмена

В наши дни почти все виды спорта в значительной степени зависят от анализа данных [3]. Определение позы может помочь игрокам улучшить свою технику и добиться лучших результатов. Помимо этого, НРЕ

может быть использовано для анализа и получения информации о преимуществах и недостатках противника.

1.2.5. Робототехника

Роботы являются неотъемлемой частью нашей повседневной жизни и их роль будет только возрастать. Существует множество областей реального мира, где робототехника, основанная на компьютерном зрении, значительно расширила возможности человека, например, медицинские вмешательства, такие как биопсия, автоматическое обнаружение и визуализация опухолей, применение роботов для обучения определенным навыкам и многое другое. С помощью компьютерного зрения и, в частности, оценки позы человека, есть шанс в будущем сделать робототехнические системы более отзывчивыми, гибкими и точными.

2. Вариации оценки позы человека

Оценка позы человека делится на два типа: оценка single-pose используется для оценки поз одного человека, в то время как оценка multi-pose при определении поз для нескольких человек в одном кадре. Последняя является более сложным из-за необходимости успешно идентифицировать каждого человека отдельно друг от друга.

Существует два способа с помощью, которых может быть выполнена оценка позы человека: двумерный (2D) и трехмерный (3D). 2D подход реализует методы извлечения признаков для прогнозирования ключевых точек человеческого тела по изображениям на основе значений пикселей. 3D подход связан с предсказанием пространственного положения конкретного человека или объекта по изображениям и видео аналогично 2D. Данный подход сложнее из-за фоновой сцены, условий освещения или других помех. Кроме того, имеется меньше доступных 3D наборов данных.

2.1. Представление человеческого тела

Существует 2 различных варианта моделирования человеческого тела для описания позы, позволяющие решать различные задачи на основе ее оценки .

2.1.1. Представление на основе ключевых точек

2D или 3D координаты ключевых точек человеческого тела являются простыми и интуитивно понятными представлениями для скелета тела, которые имеют несколько форм представления.

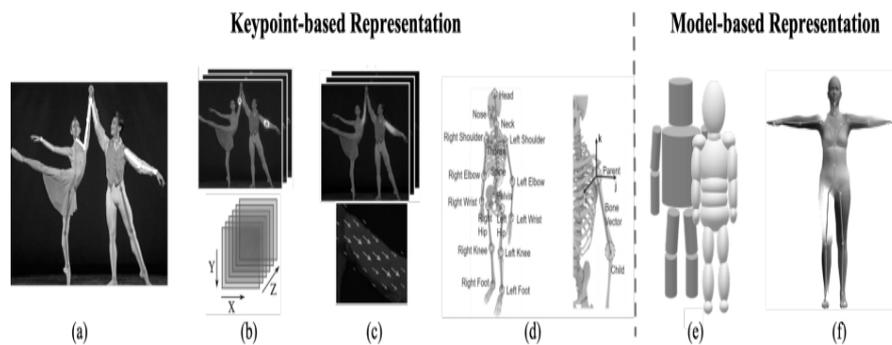


Рис. 2. Формы представления человеческого тела

- **2D/3D координаты ключевой точки.** Ключевые точки тела могут быть явно описаны 2D/3D координатами. Как показано на Рис. 2 (а), ключевые точки соединены в соответствии с присущей корпусу структурой.
- **2D/3D тепловые карты.** Чтобы сделать координаты более подходящими для регрессии сверточной нейронной сетью, многие методы представляют координаты ключевых точек в виде тепловой карты. Как показано на Рис. 2 (b), тепловая карта Гаусса каждой ключевой точки имеет высокое значение отклика на соответствующих 2D/3D координатах и низкое значение отклика в других положениях.
- **Ориентационные карты.** Этот метод использует карту ориентации ключевых точек тела в качестве вспомогательного представления тепловых карт с помощью «part affinity fields» (PAFs) для представления 2D ориентации между конечностями. Как показано на Рис. 2 (c), PAF представляет собой 2D векторное поле, которое связывает две ключевые точки конечности. Каждый пиксель в поле содержит 2D вектор, который указывает от одной конечности к другой.
- **Иерархические векторы моделирования костей.** 2D версия иерархического представления костей представляет собой комбинацию суставов и костных векторов. Она расширяется на трехмерный случай. Как показано на Рис. 2 (d), 3D человеческий скелет представлен набором костных векторов. Каждый вектор кости направлен от родительской ключевой точки к дочерней ключевой точке, следуя кинематическому дереву. Каждая родительская ключевая

точка связана с локальной сферической системой координат. Вектор кости может быть представлен сферическими координатами в этой системе.

2.1.2. Представление на основе модели

Представление на основе модели разработано в соответствии с присутствующими человеческому телу структурными характеристиками. Оно обеспечивает более информативное представление человеческого тела, по сравнению с предыдущим.

- **Объемная модель на основе деталей.** Объемная модель на основе деталей разрабатываются для решения реальных задач. Как показано на синей модели на Рис. 2 (е), каждая конечность представлена в виде цилиндра, модель была разработана для создания меток прегражденных частей. Каждый цилиндр находится путем совмещения центров верхней и нижней поверхностей с 3D ключевыми точками конечности. Как показано на розовой модели на Рис. 2 (е), предлагается модель эллипсоида, в которой эллипсоид используется в качестве базовой единицы частей тела для большей гибкости.
- **Подробная статистическая модель человеческого тела.** По сравнению с объемной моделью, основанной на деталях, статистическая трехмерная сетка человеческого тела описывает более подробную информацию, включая позу и форму тела.

Human pose estimation выполняется путем определения суставов тела. Различные алгоритмы могут идентифицировать различное количество ключевых точек человеческого тела. На Рис. 3 показан пример доступных ключевых точек.

2.2. Подходы к определению позы человека

2.2.1. «Снизу-вверх» и «сверху-вниз»

Существует 2 подхода к определению позы человека:

- Подходы **снизу-вверх** (bottom-up) сначала определяют каждый сустав тела, а затем группируют их, чтобы сформировать уникальную позу для разных людей, Рис. 4 (а).
- Подходы **сверху-вниз** (top-down) сначала определяют области ограничивающих рамок человека, а затем определяют суставы тела в пределах обнаруженных ограничивающих рамок для каждого человека по отдельности, Рис. 4 (b).

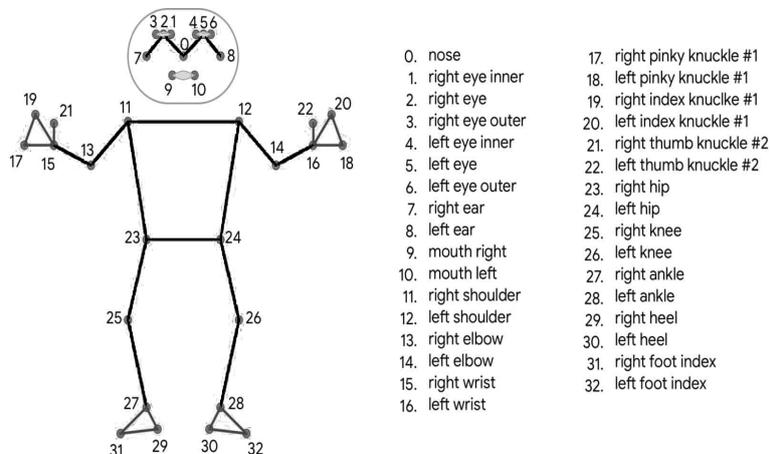


Рис. 3. Пример ключевых точек суставов человеческого тела

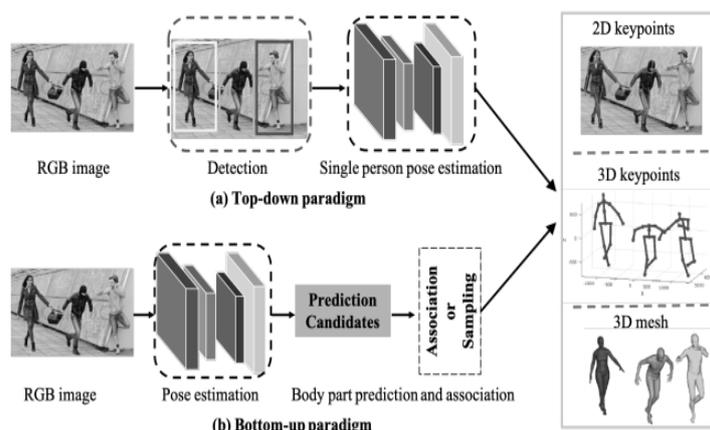


Рис. 4. Типы методов для оценки позы человека

2.2.2. Моделирование человеческого тела

Исследователи выделяют следующие модели человеческого тела:

Кинематическая модель (модель на основе скелета) используется для оценки 2D и 3D поз, Рис. 5 (а). Эта интуитивная модель человеческого тела содержит набор положений суставов и ориентаций конечностей для представления структуры человеческого тела.

Плоская Модель (модель на основе контуров) используется для оценки 2D позы, Рис. 5 (b). Эта модель используется для представления

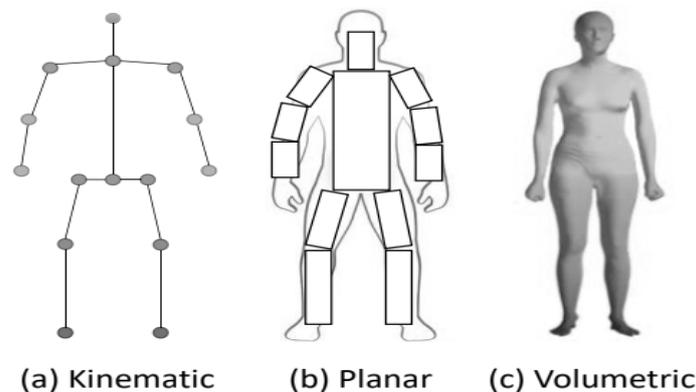


Рис. 5. Типы скелетона

внешнего вида и формы человеческого тела. Части тела представлены несколькими прямоугольниками, приблизительно повторяющими контуры человеческого тела.

Объемная модель используется для оценки позы в 3D, Рис. 5 (с). Эта модель предоставляет соответствующую информацию для оценки позы в 3D с помощью сканирования всего тела для получения сетки человеческого тела различных форм и поз.

3. Инструменты для задачи

3.1. Метрики оценки качества

Существует несколько показателей, используемых для оценки производительности моделей НРЕ:

Intersection Over Union (IoU) - это метрика которая определяет разницу между аннотациями истинными и прогнозируемыми ограничивающими рамками, Рис. 6. Удаляет все ненужные рамки на основе присвоенного порогового значения, которое обычно равно 0.5.

Percentage of Correct Parts (PCP) (процент правильных частей) - это мера, которая оценивает предсказания, чтобы сообщить о точности локализации конечностей. Локализация конечностей определяется, когда расстояние между прогнозируемым суставом и истинным суставом составляет менее доли длины конечности (от 0.1 до 0.5). Эта метрика используется для оценки single-person НРЕ. Однако, PCP не была широко

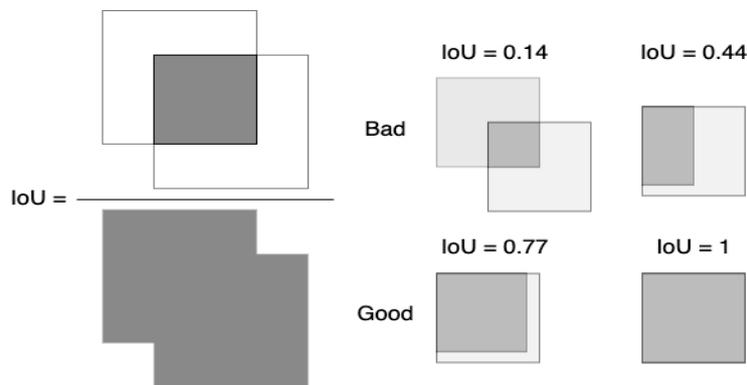


Рис. 6. Intersection over Union

применена в последних работах, потому что она наказывает конечности с короткой длиной, которые трудно обнаружить.

Percentage of Detected Joints (PDJ) (процент правильных суставов) - это метрика, в которой прогнозируемый сустав считается обнаруженным, если расстояние между прогнозируемым суставам и истинным суставом находится в пределах определенной доли диаметра туловища. Введена для того, чтобы устранить недостатки PCP

Percentage of Correct Key-points (PCK) (процент правильных ключевых точек) - используется для измерения точности локализации различных ключевых точек в пределах определенного порога, обычно устанавливаемым равным 50% от длины сегмента головы каждого тестового изображения.

Object Keypoint Similarity (OKS) (сходство ключевых точек объекта) - среднее сходство ключевых точек по всем ключевым точкам объекта. Рассчитывается на основе масштаба объекта и расстояния между прогнозируемой и истинной точками. Масштаба и константа ключевой точки необходимы для того, чтобы придать равную важность каждой ключевой точке.

Average Recall (AR) используется для измерения точности определения ключевых точек в соответствии с показателем *recall* - сколько из всех истинных положительных результатов было найдено моделью. Метрика представляет собой средние значения отзыва по всем значениям *recall* в диапазоне от 0 до 1 при различных пороговых значениях IoU. **Mean Average Recall (MAR)** - это среднее значение *recall* по всем классам при нескольких пороговых значениях по IoU всей модели.

Average Precision (AP) используется для измерения точности определения ключевых точек в соответствии с показателем *precision* - на-

сколько точны прогнозы. Следовательно, показатель представляет собой средние значения точности по всем значениям precision в диапазоне от 0 до 1 при различных пороговых значениях IoU. **Mean Average Precision (MAP)** - это среднее значение precision по всем классам при нескольких пороговых значениях по IoU всей модели.

3.2. Наборы данных

Наборы данных являются одним из наиболее важных аспектов алгоритмов машинного обучения. Задача НРЕ не является исключением. Алгоритмы должны быть сначала обучены, а затем протестированы, чтобы убедиться что они правильно интерпретируют данные для выполнения поставленной задачи. Таким образом, необходимы хорошие обучающие и даже тестовые наборы данных. Существует несколько широко используемых наборов данных для задачи оценки позы человека.

Набор данных **Сосо** - наиболее широко используемый набор 2D данных, в первую очередь для multi-person НРЕ. Используется также для обнаружения объектов и содержит изображения, помеченные ключевыми точками, до 17 по всему телу. Существует 2 версии набора данных Сосо с разделением тренировочного, валидационного и тестового поднаборов данных. Разработан в 2014 году [16].

Набор данных **МРП** - 2D набор данных, используемый в основном для single-pose моделирования. Он содержит изображения, помеченные вручную до 16 аннотированными суставами тела с расширенными обозначениями, такими как окклюзии частей тела и ориентации туловища, 3D ориентации торса и головы. Изображения охватывают 410 различных видов человеческой деятельности. Каждое изображение было взято из видео на платформе YouTube предыдущими и последующими кадрами, которые не были аннотированы. Разработан в 2014 году [17].

Набор данных **AI Challenger Human Keypoint Detection** - самый большой набор данных для двумерной оценки позы человека. Он содержит более 300 тысяч изображений с высоким разрешением, аннотированных по 14 ключевым точкам для обнаружения каждого человека и более 600 тысяч тестовых изображений. Все изображения были собраны из поисковых систем интернета и посвящены повседневной деятельности людей в различных позах [18].

PoseTrack - 2D набор данных о телосложении на основе видео, в основном используемый для multi-pose НРЕ, который содержит около 1356 видеорядов, 46 тысяч аннотированных кадров, и 276 тысяч аннотаций поз тела, содержащих до 15 ключевых точек. Предполагается, что у каждого пользователя есть уникальный идентификатор трека с аннотациями. Разработан в 2017 году [19].

Human3.6M - самый популярный и самый большой набор данных, используемый для 3D HPE. Он содержит 3.6 миллиона 3D поз человека, помеченных ключевыми 24 ключевыми точками, и использует 3 протокола с различным разделением данных для обучения и тестирования, а также 11 профессиональных актеров, выполняющих 17 действий с 4 разных точек обзора в закрытых лабораторных условиях. Разработан в 2014 году [20].

3.3. Популярные методы оценки позы человека

На сегодняшний день исследователи разработали следующие популярные модели оценки позы человека.

DeepPose [5] захватывает все соединения с помощью глубоких нейронных, добавляет слой объединения, слой свертки и полносвязный слой, чтобы сформировать части этих слоев.

OpenPose [6] - подход «снизу-вверх» для multi-person HPE с архитектурными функциями для режима реального времени. Преимущество Open Pose заключается в том, что это API, который предоставляет пользователям гибкость при выборе исходных изображений с камеры, веб-камер и других, что более важно для встроенных системных приложений.

DeepCut [7] - подход «снизу-вверх» для multi-person HPE. Модель работает путем определения количества людей на изображении, а затем прогнозирования совместных местоположений для каждого изображения. Deep Cut можно применить к видео или изображениям с несколькими объектами, например, футбол, баскетбол и другими.

Mask R-CNN - популярный алгоритм сегментации, архитектура которого может быть легко расширена для решения задач HPE. В этой модели используется сверточная нейронная сеть, которая извлекает объекты и их представление из входного изображения. Извлеченные объекты используются для генерации ряда возможных позиций для объекта, который известен как сеть региональных предложений (RPN).

MoveNet может определить 17 ключевых точек человеческого тела. Существует две версии модели. Lightning версия предназначена для приложений с низкими требованиями к задержке. Версия Thunder предназначена для приложений, которые нацелены на достижение более высокой точности. Обе модели способны обнаруживать данные в режиме реального времени и доказали свою эффективность для обнаружения приложений, работающих в режиме реального времени в фитнесе, спорте или здравоохранении. Совместно с MediaPipe достигли передовых результатов в предсказании как изображений, так и видео различных видов человеческой деятельности [15].

PoseNet [8] может использоваться для single-person или multi-person HPE для запуска на легких устройствах, таких как браузеры или мобильные устройства.

DCPose [9] или Deep Dual Consecutive Network определяет позу человека из нескольких кадров. Модель пытается преодолеть критические проблемы HPE, такие как размытость изображения при движении, расфокусированное видео и перекрытия, возникающие из-за зависимости от каждого видеокadra. Модель использует различные временные привязки между этими кадрами видео для облегчения точного определения ключевых точек.

DensePose [10] отображает различные пиксели на основе человека из RGB изображения относительно 3D поверхности человеческого тела. Модель использует крупный набор данных, содержащий аннотированную информацию от изображения к поверхности. Модель поддерживает single-person и multi-person HPE.

High-Resolution Net (HRNet) [11] находит ключевые точки относительно конкретного объекта или человека на изображении. Одним из преимуществ этой архитектуры является то, что большинство существующих моделей сопоставляют представления поз с высоким разрешением с представлениями с низким разрешением.

Regional Multi-Person Pose Estimation (AlphaPose) [12] - подход «сверху-вниз» для задачи HPE как для single-person так и multi-person случаев на изображении или видео. Модель используется для оценки человеческих поз по оптимально определенным ограничивающим рамкам при наличии их неточности.

TransPose [13] реализует основанный на сверточной нейронной сети подход к извлечению признаков, трансформер энкодер и возможности предсказания. Модель имеет встроенные функции, такие как преобразователь, который может собирать информацию из пространственных соотношений на большом расстоянии между различными ключевыми точками.

MediaPipe Pose предназначена для высокоточного отслеживания положения тела: доступно определение 33 трехмерных ориентиров и маски сегментации фона на всем теле из RGB видеокadров с использованием модели BlazePose [14], которая точно локализует ключевые точки, что делает его уникальным для фитнес приложений. Этот подход обеспечивает производительность в реальном времени на мобильных телефонах с процессорными вычислениями.

4. Применение задачи для обучения физическим навыкам

4.1. Приложения для фитнеса

Индустрия цифрового фитнеса переживает невероятный рост после COVID-19. Количество загрузок для здоровья и фитнеса выросло на 46% по всему миру 2020 году, и ожидается, что к 2025 году индустрия вырастет до \$16 млрд. [4].

Рассмотрим несколько компаний, которые разрабатывают подходы к фитнес приложениям с помощью искусственного интеллекта:

- 1) **Mirror** [21] программное обеспечение, анализирующее движения на тренировках и обеспечивающее обратную связь в режиме реального времени, основанную на предпочтениях и целях пользователя.
- 2) **Kaia Health** [22] использует компьютерное зрение и селфи камеру для предоставления звуковых подсказок и визуальной обратной связи для более эффективного выполнения упражнений.
- 3) **Vay** [23] предлагают приложение, анализирующее упражнения с помощью захвата камеры. Им удалось быстро расширить базу упражнений и добиться независимости от положения камеры.
- 4) **TwentyBN** [24] - это фитнес-киоск с ИИ тренером, который сочетает в себе видео понимание и NLP, ведет и корректирует тренировку в режиме реального времени. Их система искусственного интеллекта опирается на данные реальных пользователей и отслеживает различные движения при выполнении упражнений.

4.2. Персонализация

Современные фитнес решения с искусственным интеллектом основаны на анализе прогнозов алгоритмов оценки позы человека. Они получают координаты частей тела человека и сравнивают их с «эталонными». Но это не надежный и универсальный подход из-за отсутствия интерпретации разницы между двумя позами.

Чтобы преодолеть эту проблему, можно разработать решение, основанное на системе нечетких правил [25], которое может учитывать правила экспертной области для принятия решений, чтобы персонализированно следить за выполнением различных физических упражнений. Такие правила позволят обеспечивать более понятные и надежные решения. Например, правило может состоять из сравнения двух человеческих поз и установления фиксированных границ корректности позы.

Это может стать основой для системы рекомендаций, основанной на оценке позы человека, для различных видов деятельности от фитнеса до массовых профессий.

5. Заключение

В работе представлены достижения в области оценки позы человека, а также рассмотрены фундаментальные понятия постановки задачи, ее разновидности и существующие решения. Также проведен анализ применения данной технологии для обучения физическим навыкам на примере фитнес приложений.

Итоги исследования показывают, что решения в области оценки позы человека могут быть применены для построения инструмента персонализации обучения физическим навыкам. Такой инструмент может быть реализован в виде системы управления процессом тренировки различным физическим навыкам. Подобные персонализированные системы будут полезны в спорте (от фитнеса до подготовки спортсменов высшей квалификации) [26], здравоохранении (процессы реабилитации) [27], обучении выполнению физических навыков (рынок массовых профессий).

The problem of human pose estimation: tasks, methods, solutions Shergin I.A., Ryjov A.P.

The article deals with the problem of estimating a person's pose, which arises in many applications of computer vision systems. Various problem statements and approaches to solving the problem are considered, an overview of existing solution methods and data sets is provided. Additionally, the application of methods in the task of teaching physical skills is investigated.

Keywords: machine learning, deep learning, human posture estimation, computer vision.

References

- [1] Shubham Sharma, “Meet Zenia, an AI that can guide your yoga sessions”, <https://www.newsbytesapp.com/news/science/zenia-ai-assistant-will-guide-your-yoga-sessions/story>.
- [2] Yuan Niu, Jinhua She, Chi Xu, “A Survey on IMU-and-Vision-based Human Pose Estimation for Rehabilitation”, 2022 <http://dx.doi.org/10.23919/CCC55666.2022.9901525>.

- [3] Rui Liu, “A Study of Athlete Pose Estimation Techniques in Sports Game Videos Combining Multiresidual Module Convolutional Neural Networks”, 2021 <http://dx.doi.org/10.1155/2021/4367875>.
- [4] Konstantin Kalinin, “Pose Estimation: How to Track Human Movements in Real-Time Using Machine Learning”, 2021, <https://topflightapps.com/ideas/pose-estimation/>.
- [5] Alexander Toshev, Christian Szegedy, “DeepPose: Human Pose Estimation via Deep Neural Networks”, 2014 <http://dx.doi.org/10.1109/CVPR.2014.214>.
- [6] Zhe Cao, Tomas Simon, Shih-En Wei and Yaser Sheikh, “OpenPose: Realtime Multi-Person 2D Pose Estimation using Part Affinity Fields”, 2019.
- [7] Leonid Pishchulin, Eldar Insafutdinov, Siyu Tang, “DeepCut: Joint Subset Partition and Labeling for Multi-Person Pose Estimation”, 2016.
- [8] Pose Detection in the Browser: PoseNet Model <https://github.com/tensorflow/tfjs-models/tree/master/posenet>.
- [9] Zhenguang Liu, Haoming Chen, “Deep Dual Consecutive Network for Human Pose Estimation”.
- [10] Rıza Alp Guler, Natalia Neverova, Iasonas Kokkinos, “DensePose: Dense Human Pose Estimation In The Wild”, 2018.
- [11] Bowen Cheng, Bin Xiao, Jingdong Wang, “HigherHRNet: Scale-Aware Representation Learning for Bottom-Up Human Pose Estimation”, 2020.
- [12] Hao-Shu Fang, Jiefeng L, “AlphaPose: Whole-Body Regional Multi-Person Pose Estimation and Tracking in Real-Time”, 2022.
- [13] Sen Yang, Zhibin Quan, “TransPose: Keypoint Localization via Transformer”.
- [14] Valentin Bazarevsky, Ivan Grishchenko, Karthik Raveendran, “BlazePose: On-device Real-time Body Pose tracking”, 2020 <https://doi.org/10.48550/arXiv.2006.10204>.
- [15] Jen-Li Chung, Lee-Yeng Ong and Meng-Chew Leow, “Comparative Analysis of Skeleton-Based Human Pose Estimation”, 2022.
- [16] Tsung-Yi Lin, “Coco dataset”, <https://cocodataset.org/home>.
- [17] Mykhaylo Andriluka, “2d human pose estimation: New benchmark and state of the art analysis”, 2014.

- [18] Jiahong Wu, “Large-scale datasets for going deeper in image understanding”, 2019.
- [19] Mykhaylo Andriluk, “Posetrack: A benchmark for human pose estimation and tracking”, 2018.
- [20] Catalin Ionescu, “Human3.6m: Large scale datasets and predictive methods for 3d human sensing in natural environments”, 2014.
- [21] DAMI LEE, “Mirror launches live training sessions that let coaches see you at home”, <https://www.theverge.com/2019/10/8/20899151/mirror-live-personal-training-interactive-home-gym-fitness-session-price>.
- [22] Kaia Health Team, “Digital Exercise Performance Tracking with Kaia Health’s Motion Coach”, <https://kaiahealth.com/blog/motion-coach-exercise-tracking>.
- [23] Robin Wauters, “University of Zurich spin-off Vay Sports has launched its AI-powered digital fitness coach in beta”, <https://tech.eu/2019/05/06/university-of-zurich-spin-off-vay-sports-has-launched-its-ai-powered-digital-fitness-coach-in-beta/>.
- [24] Twenty Billion Neurons Team, “2 fit-tech trends are leading us towards interactive AI fitness trainers”, <https://fityourself.club/2-fit-tech-trends-are-leading-us-towards-interactive-ai-fitness-trainers-2c74d4da4890>.
- [25] Malinka Ivanova, Petya Petkova, Nikolay Petkov, “Machine Learning and Fuzzy Logic in Electronics: Applying Intelligence in Practice”, 2021.
- [26] N. Mahendran, “Deep Learning for Fitness”, 2021.
- [27] J. Stenum, “Applications of Pose Estimation in Human Health and Performance across the Lifespan”, 2021.

Часть 3.
Математические модели

О числе состояний автомата, модифицирующего свою диаграмму

Д. О. Маслеников¹

Вводятся понятия правила - пары слов одинаковой длины, и результата применения его к конечному инициальному автомату, когда в случае вывода первого слова из этой пары оно заменяется на диаграмме на второе.

В работе исследуется результат применения правила к автомату, как словарная функция и как реализующий её конечный автомат. Показана эквивалентность двух определений результата применения правила к автомату. Показано, что он реализует ограниченно-детерминированную функцию. Получены оценки на наибольшее и наименьшее число состояний соответствующего ей приведённого конечного инициального автомата.

Ключевые слова: конечный инициальный автомат, самомодифицирующийся конечный автомат, диаграмма Мура.

1. Введение

В этой работе рассмотрим модификацию детерминированного конечного автомата: пусть по диаграмме автомата перемещается головка машины Тьюринга, как по ленте, а сама машина работает так, что её поведение совпадает с обычной работой автомата, пока выводом не окажется определённое слово, тогда машина, вернувшись назад, исправляет на рёбрах его на другое. Будет доказано, что такая конструкция будет эквивалента некоторому автомату, и будут даны оценки наибольшего и наименьшего числа состояний этого автомата.

Изменяющиеся автоматы рассматривались и ранее. Например, Я. М. Барздинь в работах [1, 2] рассмотрел логические сети, в которых соединения между элементами меняются со временем.

Рой С. Рубинштейн и Джон Н. Шатт в [3] предложили модель самомодифицирующегося конечного автомата (self-modifying finite automaton)— недетерминированного автомата, который может модифицировать себя во время перехода из одного состояния в другое, добавляя или удаляя переходы и состояния.

М. Костер и Дж. Тейч в [4] рассмотрели реконфигурирующийся конечный автомат (reconfigurable finite state machine)— детерминирован-

¹Маслеников Денис Олегович — студент каф. математической теории интеллектуальных систем мех.-мат. ф-та МГУ, e-mail: denismaslenikov01@mail.ru.

Maslenikov Denis Olegovich — student, Lomonosov Moscow State University, Faculty of Mechanics and Mathematics, Chair of Mathematical Theory of Intellectual Systems.

ный автомат, который при переходе из одного состояния в другое может изменить входное состояние и выводимый символ этого перехода, причём то, какой переход будет выполнен и какие изменения произойдут, зависит не только от входного символа, но и от состояния реконфигурации, которое подаётся извне. Если же оно генерируется частью системы, то такой автомат называется самореконфигурирующимся конечным автоматом (self-reconfigurable finite state machine).

В данной работе будет рассмотрена модель наиболее похожая на самореконфигурирующийся конечный автомат, однако, в отличие от автомата, рассмотренного в статье Костера и Тейча, в автомате, описанном в этой статье, возможно изменение выходных символов на нескольких переходах, но, с другой стороны, не допускается изменение самих переходов.

2. Определения и основные результаты

Введем ряд определений.

2.1. Определения

Правилom длины l над алфавитом B называется пара слов $(b_1b_2 \dots b_l, b'_1b'_2 \dots b'_l)$ из символов алфавита A . Также введём обозначения: $l(r) = l$, $R_B = \bigcup_{l \in \mathbb{N}} (B^l \times B^l)$ - множество правил над алфавитом A .

$Q(V) = Q$, где $V = (A, Q, B, \psi, \phi, q(1))$.

$\Psi_{A,B} = \{\psi : Q \times A \rightarrow B\}$ - множество функций вывода, где Q - некоторое фиксированное множество состояний.

$V_{A,B}$ - множество автоматов с входным и выходным алфавитом A и B соответственно и множеством состояний Q .

Модифицируем работу инициального автомата V следующим образом: пусть есть правило $r = (\alpha, \beta)$ длины n над выходным алфавитом. Тогда если в какой-то момент суффикс длины n выходного слова равен α , то в диаграмме Мура на последних n пройденных рёбрах заменяем выходные символы $\alpha_1, \dots, \alpha_n$ на β_1, \dots, β_n соответственно, начиная с первого и заканчивая последним. Далее функционирование автомата продолжается из последнего состояния по обновлённой диаграмме Мура. Полученное "устройство" отображает входные слова в выходные, то есть реализует некоторую словарную функцию, которую будем называть *результатом применения правила r к диаграмме автомата V* .

Получившуюся функцию вывода будем обозначать $\chi(V, r, \gamma)$, где γ - входное слово.

Приведём два примера.

Пример 1. Пусть автомат V имеет диаграмму, как на Рис. 1, начальное состояние - q_1 (здесь и дальше будем обозначать текущее состояние "звёздочкой"), $r = (101, 110)$, а входное слово - (111) , тогда будет выведено слово (101) . При этом будут пройдены рёбра, отмеченные на Рис. 2 нижними индексами, обозначающими порядок их прохода, и диаграмма примет вид, как на Рис. 3.

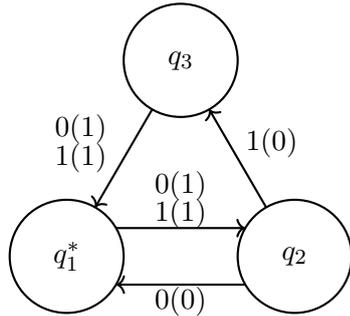


Рис.1

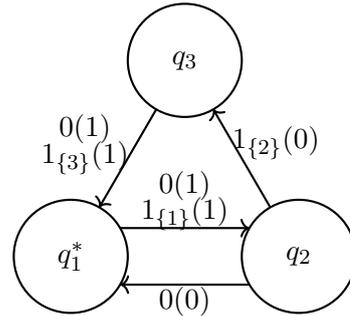


Рис.2

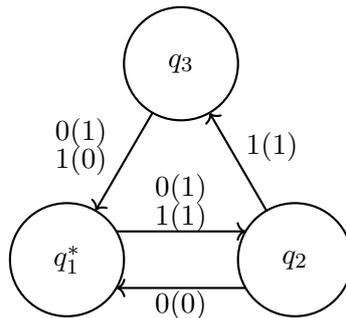


Рис.3

Пусть теперь входное слово - (11111) . В соответствии с Рис. 3, будет выведено (10111) .

Пример 2. Пусть автомат и правило те же, что и в примере 1 (Рис. 4), а входное слово (000) . Заметим, что $\psi(q_1, 0)$ переписывается два раза подряд (Рис. 5). Поскольку β записывается, начиная с первого символа, $\psi(q_1, 0)$ в итоге равно 0, поэтому выводится слово (101) , и диаграмма принимает вид, как на Рис. 6. Пусть теперь входное слово - (00011) , тогда будет выведено (10101) . Далее будем нумеровать рёбра так, чтобы были занумерованы только те рёбра, которые дали последнее 101 (Рис. 7). Видим, что диаграмма после вывода пятого символа выходного слова примет вид, как на Рис. 8.

Множества, записанные в нижние индексы и нумерующие рёбра по порядку вывода первого слова правила, будем обозначать $I_{q,x}^\alpha$, где α -

входное слово. Если же ребро не было пройдено за последние l шагов, где l - длина правила, то считаем их пустыми. В примере их запись опущена.

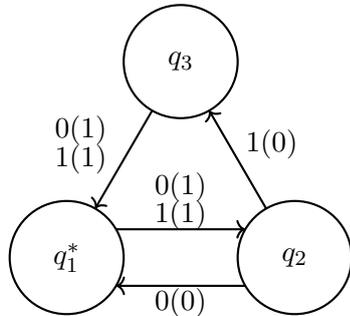


Рис.4

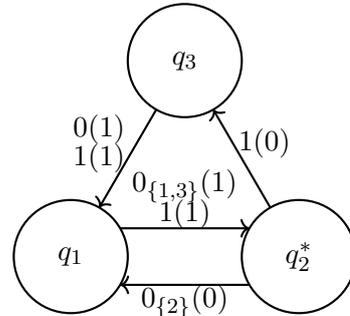


Рис.5

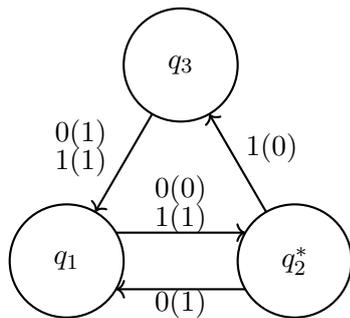


Рис.6

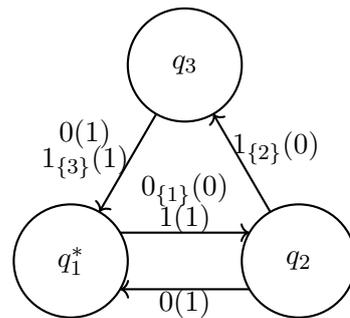


Рис.7

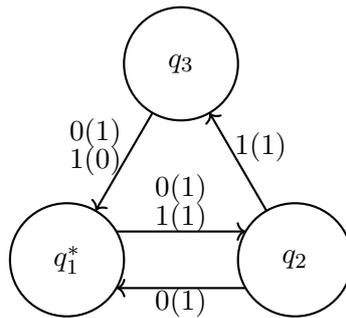


Рис.8

Приведём формальное определение эквивалентное данному.

Результатом применения правила $r \in B$ к инициальному автомату $V = (A, Q, B, \phi, \psi, q(1))$ называется словарная функция $\omega(V, r)$, где $\omega : V_{A,B} \times R_B \rightarrow P_{A,B}$ такая, что $\omega_t(V, r)(x_1 \dots x_t) = \chi(V, r, x_1 \dots x_{t-1})(q(t), x_t)$; $q(t) = \phi(q(t-1), x_{t-1})$;

$$\begin{aligned} \chi: V_{A,B} \times R_B \times A^* &\rightarrow \Psi_{A,B}; \chi(V, r, \Lambda) = \psi; \\ \chi(V, r, x_1 \dots x_t)(q, x) &= \\ &= \begin{cases} b'_{\max I_{q,x}^{x_1 \dots x_t}}, & \text{если } t \geq l, \omega(V, r)(x_1 \dots x_t) = \dots b_1 \dots b_l \text{ и } I_{q,x}^{x_1 \dots x_t} \neq \emptyset; \\ \chi(V, r, x_1 \dots x_{t-1})(q, x), & \text{иначе,} \end{cases} \end{aligned}$$

где Λ - пустое слово, и в нижний индекс b равен максимуму множества $I_{q,x}^{x_1 \dots x_t} = \{i \in 1 \dots l \mid q(t-l+i) = q, x_{t-l+i} = x\}$.

Имеем два определения - строгое и более наглядное. Далее покажем их эквивалентность.

Утверждение 1. *Результат применения правила r к автомату V равен результату применения правила r к диаграмме автомата V .*

Доказательство. Заметим, $q(t)$ совпадает с состоянием автомата V в t -й момент времени. Далее, по предположению индукции, ω совпадает с результатом применения правила r к диаграмме автомата V . Одновременно $\chi(V, r, x_1 \dots x_t)$ совпадает с функцией вывода ψ после того, как на вход поступило слово $x_1 \dots x_t$. Последнее, очевидно, верно при $\gamma = \Lambda$. Также несложно заметить, что $I_{q,x}^{x_1 \dots x_t}$ из обоих определений - одно и то же. Пусть значение результата применения правила r к диаграмме автомата V оканчивается на $b_1 \dots b_l$, что по предположению индукции означает $\omega(V, r)(x_1 \dots x_t) = \dots b_1 \dots b_l$. Тогда для всех q и x таких, что $I_{q,x}^{x_1 \dots x_t} \neq \emptyset$ на соответствующем ребре поочерёдно записываются выходные символы b'_i , где i из $I_{q,x}^{x_1 \dots x_t}$ возрастает. То есть, для таких q и x выполняется $\psi = b'_{\max I_{q,x}^{x_1 \dots x_t}} = \chi(V, r, x_1 \dots x_t)(q, x)$. В остальных случаях функция вывода не меняется, то есть $\psi = \chi(V, r, x_1 \dots x_{t-1}) = \chi(V, r, x_1 \dots x_t)$. Итак, мы доказали предположение индукции для χ . Отсюда следует, что t -й выходной элемент результата применения правила r к диаграмме автомата V равен $\chi(V, r, x_1 \dots x_{t-1})(q(t), x_t)$. Таким образом, предположение индукции верно для ω , то есть результат применения правила к автомату совпадает с результатом применения правила r к диаграмме автомата V . ■

2.2. Верхняя оценка наибольшего числа различных состояний

Теорема 1. *Пусть α и β - слова длины l . Результат применения правила $r = (\alpha, \beta)$ к автомату $V = (A, Q, B, \phi, \psi, q(1))$ является ограниченно-детерминированной функцией, имеющей не более чем $Cn|B|^{n|A|}$ остаточных функций, где $n = |Q|$ и*

$$C = \begin{cases} \frac{|A|^l - 1}{|A| - 1}, & |A| \neq 1; \\ l, & \text{иначе.} \end{cases}$$

Доказательство. Возьмём автомат $\tilde{V} = (A, \tilde{Q}, B, \tilde{\phi}, \tilde{\psi}, \tilde{q}(1))$, где

$$\tilde{Q} = \bigcup_{i=0}^{l-1} A^i \times Q \times \Psi;$$

$\tilde{\phi}((a_1, \dots, a_i, q, \psi'), a) = (a_m, \dots, a_i, a, \phi(q, a_1 \dots a_{m-1}), \chi'(a_1, \dots, a_i, a, q, \psi'))$;
 m такое, что слово

$\alpha_m \dots \alpha_i \tilde{\psi}((a_1, \dots, a_i, q, \psi'), a)$ - наибольший суффикс слова

$\alpha_1 \dots \alpha_i \tilde{\psi}((a_1, \dots, a_i, q, \psi'), a)$, являющийся собственным префиксом α ;

$$\begin{aligned} & \chi'(a_1, \dots, a_i, a, q, \psi')(q', x) = \\ & = \begin{cases} \beta_{\max I_{q',x}}, & \text{если } i = l - 1, \tilde{\psi}((a_1, \dots, a_i, q, \psi'), a) = \alpha_l \text{ и } I_{q',x} \neq \emptyset; \\ \psi'(q', x), & \text{иначе;} \end{cases} \end{aligned}$$

$I_{q',x} = \{i \in 1 \dots l \mid \phi(q, a_1 \dots a_{i-1}) = q', a_i = x\}$; $\tilde{\psi}((a_1, \dots, a_i, q, \psi'), a) = \psi'(q, a_1 \dots a_i a) = \psi'(\phi(q, a_1 \dots a_i), a)$; $\tilde{q}(1) = (q(1), \psi)$. (В определении $I_{q',x}$ считаем, что $\phi(q, a_1, a_0) = q$, и $a_l = a$.)

Возьмём в качестве предположения индукции, что автомат соответствует результату применения правила r к автомату V . Заметим, что в таком случае также по индукции получаем

$$\tilde{\phi}(\tilde{q}(1), \gamma) = (\gamma_{t-i+1}, \dots, \gamma_t, \phi(q(1), \gamma_1 \dots \gamma_{t-i}), \psi'),$$

где t - длина γ ; i - длина наибольшего суффикса выходного слова, являющегося собственным префиксом α ; $\psi' = \chi(V, r, \gamma)$ из определения результата применения правила к автомату.

Исходя из этого, верно предположение индукции, так как $\tilde{\psi}(\tilde{q}(1), \gamma a) = \tilde{\psi}(\tilde{\phi}(\tilde{q}(1), \gamma), a) = \tilde{\psi}((\gamma_{t-i+1}, \dots, \gamma_t, \phi(q(1), \gamma_1 \dots \gamma_{t-i}), \psi'), a) = \psi'(\phi(q(1), \gamma_1 \dots \gamma_{t-i}), \gamma_{t-i+1} \dots \gamma_t a) = \psi'(\phi(\phi(q(1), \gamma_1 \dots \gamma_{t-i}), \gamma_{t-i+1} \dots \gamma_t), a) = \psi'(\phi(q(1), \gamma_1 \dots \gamma_t), a) = \psi'(q(t+1), a) = \chi(V, r, \gamma)(q(t+1), a) = \omega_{t+1}(V, r)(\gamma a)$.

Следовательно, число различных состояний не больше, чем число состояний \tilde{V} , то есть

$$n|B|^{n|A|} \sum_{i=0}^{l-1} |A|^i = n \frac{|A|^l - 1}{|A| - 1} |B|^{n|A|} \text{ при } |A| > 1 \text{ и } n|B|^{n|A|} \text{ при } |A| = 1. \quad \blacksquare$$

В силу того, что результатом является ограниченно детерминированная функция, будем считать, что результатом также является конечный автомат приведённого вида. Обозначим его V_r .

2.3. Нижняя оценка наибольшего числа состояний

Перед тем, как рассмотреть вторую теорему о максимальном числе состояний, приведём пример.

Пример 3. Рассмотрим автомат на Рис. 9:

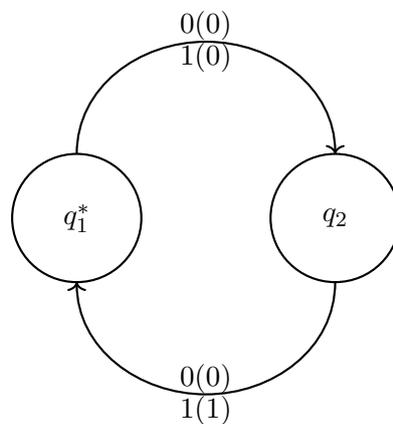


Рис.9

Применим к нему правило $r = (0, 1)$. И построим в явном виде V_r . Заметим, что, поскольку r - правило длины 1, состояние V_r определяется текущими функцией вывода (диаграммой) и состоянием V с точностью до перестановки состояний. Например, если мы имеем диаграммы, как на Рис. 10, то в автомате V_r мы будем иметь одно и то же состояние.

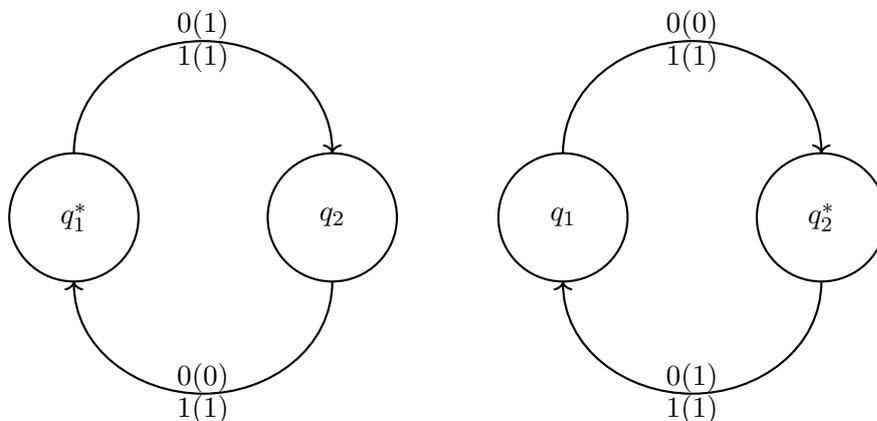


Рис.10

Выпишем все возможные диаграммы с точностью до перестановки состояний, считая, что текущее состояние на каждой расположено слева, и обозначим их, как состояния V_r (Рис. 11). Устанавливаем функцию переходов и получаем диаграмму на Рис. 12.

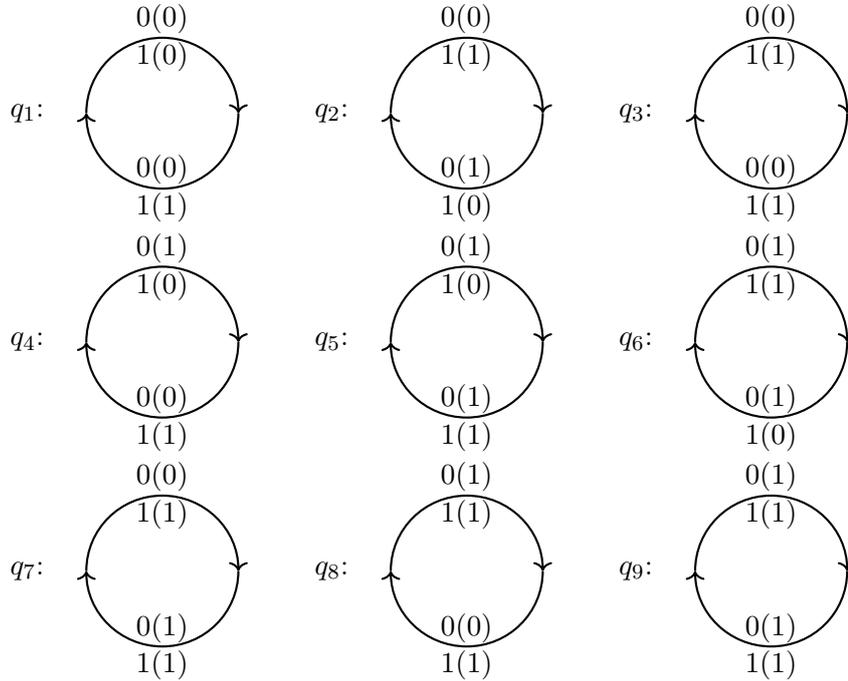


Рис.11

Рассмотрим более общий случай. Возьмём автомат V^n на Рис. 13 и правило $r = (0, 1)$.

Теорема 2. V_r^n имеет $3^n + 2^{n-1} - 2$ состояний.

Доказательство. Заметим, что в общем случае так же, как и в примере выше, состояние V_r^n определяется диаграммой и текущим состоянием с точностью до перестановок вида $\sigma_k(q_i) = q_{i+k \bmod n}$ состояний автомата V^n . Так, если текущее состояние - q_i , а диаграммы отличаются значением $\psi(q_{i+j \bmod n}, a)$, то отличающим словом будет a^{j+1} .

Пусть $t \leq n - 1$, тогда в t -й момент времени автомат V^n пришёл в состояние q_t . Есть ровно $n - t$ состояния q таких, что $\psi(q, x) \equiv 0$, и могло быть получено на вход 2^{t-1} различных слов, каждое из которых приводит к разной диаграмме. Итого, за первые $n - 1$ момента времени можем получить $\sum_{t=1}^{n-1} 2^{t-1} = 2^{n-1} - 1$ различные диаграммы.

Начиная с момента n диаграмма удовлетворяет условию: $\nexists q : \psi(q, x) \equiv 0$. Поэтому начиная с него автомат V_r^n принимает состояния отличные от тех, что мог принять ранее. Более того, автомат V^n может прийти из состояния в состояние, не изменив диаграмму. Также выполнено второе условие на диаграмму: $\exists q : \psi(q, 1) = 1$. Зафиксировав с помощью перестановок текущее состояние q_1 автомата V^n (состояние как бы

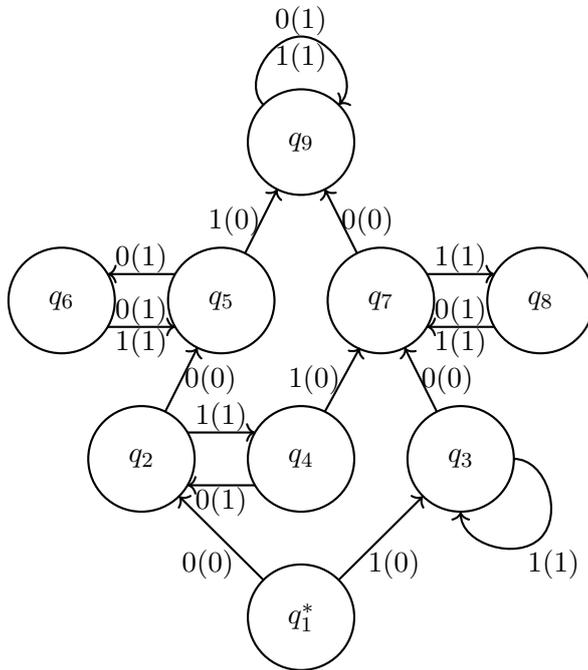


Рис.12

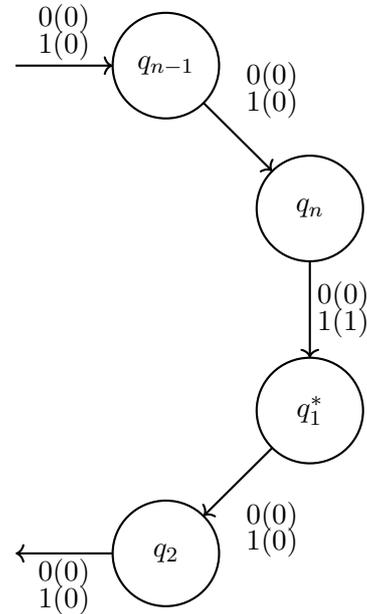


Рис.13

не движется по диаграмме по кругу, а диаграмма «поворачивается» вокруг центра против часовой стрелки), можем получить любую диаграмму, удовлетворяющую этим двум условиям. Пройдя по соответствующим ребрам, можно получить требуемую диаграмму, а единственную единицу на выходе в исходной диаграмме поставить «поворотами» диаграммы после состояния q в соответствии со вторым условием. Существует 3^n диаграмм, удовлетворяющих первому условию. Из них не удовлетворяет при этом второму ровно одна, так как имеем $\psi(q, 1) \equiv 0$ и, исходя из этого равенства и первого условия, $\psi(q, 0) \equiv 1$.

Итого, имеем $(2^{n-1} - 1) + (3^n - 1) = 3^n + 2^{n-1} - 2$ различных состояния автомата V_r^n . ■

Следствие 1. Пусть $|A| = |B| = 2$, тогда

$$3^n + 2^{n-1} - 2 \leq \max_{|\mathcal{Q}(V)|=n, l(r)=1} |\mathcal{Q}(V_r)| \leq n4^n.$$

2.4. Наименьшее число состояний

Для любого $n \in \mathbb{N}$ существует такой автомат V с n состояниями и правилом r такие, что V_r имеет одно состояние. Достаточно взять в качестве V автомат с одним достижимым состоянием, выводящий всегда 1, а в

качестве r правило $(0, 1)$. Теперь будем требовать, чтобы автомат V был приведённый.

Теорема 3. $\min_{|Q(V)|=n} |Q(V_r)| \leq \lfloor \frac{n}{2} \rfloor + 1$, где минимум берётся по автоматам приведённого вида, и всем правилам.

Доказательство. Возьмём автомат V , как на Рис. 14, и $r = (0^l 1, 10^l)$, $2l < n$, $m = l + 1$. Заметим, что автомат примет состояние q_n через $n - l - 1 > l - 1$ момент, то есть в $(n - l)$ -й момент автомат выведет 1, и диаграмма примет вид, как на Рис. 15.

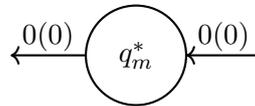
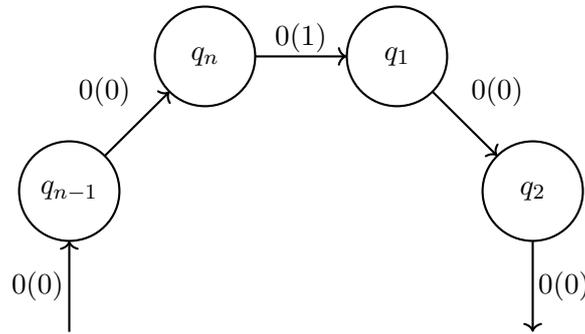


Рис.14

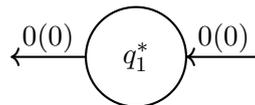
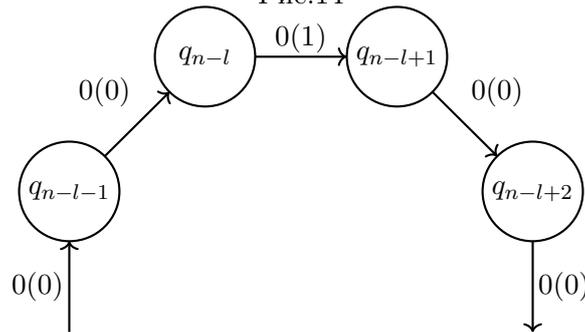


Рис.15

Автомату потребуется $n-l-1$ момент, чтобы прийти в состояние q_{n-l} , то есть он вернулся к изначальному положению. Таким образом, автомат V_r выводит слово $(0^{n-l-1}1)^\infty$. Другими словами, он представляет собой автомат аналогичный исходному, но с числом состояний равным $n-l$. Возьмём $l = \lfloor \frac{n-1}{2} \rfloor$, тогда число состояний V_r равно $\lfloor \frac{n}{2} \rfloor + 1$. ■

3. Заключение и выводы

В этой работе были получены оценки на наибольшее и наименьшее число состояний результата применения правила к автомату. Наибольшее число состояний экспоненциально зависит от числа состояний исходного автомата. При этом приведён пример автомата и правила, что после применения правила возникает экспоненциальное количество отличимых состояний. Также было показано, что возможны невырожденные случаи, когда результат имеет меньшее количество состояний.

4. Благодарность

Выражаю благодарность своему научному руководителю А. П. Соколову.

Список литературы

- [1] Барздинь Я. М., “Универсальные пульсирующие элементы”, *Докл. АН СССР*, **157**:2 (1964), 291–294.
- [2] Барздинь Я. М., “Проблемы универсальности в теории растущих автоматов”, *Докл. АН СССР*, **157**:3 (1964), 542–545.
- [3] Roy S. Rubinstein, John N. Shutt, “Self-Modifying Finite Automata”, *IFIP Congress*, 1993.
- [4] M. Koster, J. Teich, “(Self-)reconfigurable Finite State Machines: Theory and Implementation”, *Proceedings 2002 Design, Automation and Test in Europe Conference and Exhibition*, 2002, 559–566.

About number of states of automaton modifying its diagram Maslenikov D. O.

Definitions of a rule and a result of applying it to a finite automaton are introduced.

The present paper considers the result of applying the rule to the automaton as a boundedly deterministic function and a finite automaton that realizes this function.

Equivalence of two definitions of a result of applying a rule to an automaton is proved. The paper provides estimations of largest and smallest number of states of the automaton.

Keywords: finite automaton, self-modifying finite state machine, Moore diagram.

References

- [1] Barzdin Ya. M., “Universal pulsating elements”, *Report of Academy of Sciences of the Soviet Union*, **157**:2 (1964), 291–294 (in Russian).
- [2] Barzdin Ya. M., “Problems of universality in theory of growing automata”, *Report of Academy of Sciences of the Soviet Union*, **157**:3 (1964), 542–545 (in Russian).
- [3] Roy S. Rubinstein, John N. Shutt, “Self-Modifying Finite Automata”, *IFIP Congress*, 1993.
- [4] M. Koster, J. Teich.M. Koster, J. Teich, “(Self-)reconfigurable Finite State Machines: Theory and Implementation”, *Proceedings 2002 Design, Automation and Test in Europe Conference and Exhibition*, 2002, 559–566.

О свойстве линейной реализуемости отображений

С. Б. Родин¹

В данной работе изучается линейная реализуемость отображений на конечном множестве. Данное свойство важно с точки зрения линейной реализуемости автоматов, а именно, линейная реализуемость порождающих внутренней полугруппы автомата является одним из необходимых условий линейной реализуемости автомата. Ранее было показано, что любое отображение на конечном множестве является линейно реализуемой посредством кодирования с длиной кода равной мощности множества. В данной работе этот результат будет усилен и будет показано, что любое отображение является линейно реализуемым посредством кодирования, с длиной кода равной мощности множества минус один.

Ключевые слова: теория автоматов, переходные системы, подстановка, кодирование, сложность, булев оператор

1. Введение

На практике часто приходится решать задачу перехода от автоматного описания функционирования на язык схем. Например, при логическом синтезе чипов на первом этапе функционирование чипа описывается как конечный автомат. Переход к описанию на языке схем осуществляется с помощью кодирования алфавита состояний, входного алфавита и выходного алфавита в алфавите $E_2 = \{0, 1\}$. В результате кодирования возникает булев оператор. При этом автомат может обладать тем свойством, что каждое кодирование порождает оператор, отличный от оператора, порождаемого любым другим кодированием [8].

Возникаемый в результате кодирования булев оператор можно рассматривать как набор булевых функций. Сложность такого оператора можно определить как максимальную сложность получающихся булевых функций. Как известно [1], каждой булевой функции единственным образом соответствует полином Жегалкина. В статье [10] было предложено определить сложность как максимальную из сложностей полиномов Жегалкина функций, задающих этот оператор, т. е. как максимальную

¹ *Родин Сергей Борисович* — старший научный сотрудник каф. математической теории интеллектуальных систем мех.-мат. ф-та МГУ, e-mail: sergei_rodin@mail.ru.

Rodin Sergei Borisovich — Senior research scientist, Lomonosov Moscow State University, Faculty of Mechanics and Mathematics, Chair of Mathematical Theory of Intellectual Systems.

степень полиномов. Тогда простейшими с точки зрения такой сложности являются такие операторы, что соответствующие полиномы Жегалкина имеют первую степень, или линейные булевы функции. Интересно заметить, что максимальность мощности множества возникаемых для автомата посредством кодирований операторов [8] не гарантирует существование «простой», в указанном выше смысле, реализации автомата [9].

В статье [10] был доказан критерий линейной реализуемости нумерованной переходной системы $V = (E_2, E_n, \varphi)$ [2] посредством избыточного кодирования F . Данный критерий был сформулирован в терминах порождающих внутренней полугруппы переходной системы [7]. Обозначим через p_0 отображение на n -элементном множестве [3], индуцированное входным символом 0, а через p_1 отображение на n -элементном множестве, индуцированное входным символом 1. Для линейной реализуемости переходной системы необходимыми и достаточными условиями являются, во-первых линейная реализуемость отображений p_0 и p_1 , во-вторых выполнения свойства «аддитивного сдвига» на отображениях p_0 и p_1 [10]. Данная работа посвящена изучению первого свойства, а именно линейной реализуемости отображений.

В сформулированном критерии длина кода ограничена $\lceil \log_2 n \rceil$. Однако переходные системы, не являвшиеся линейно реализуемыми посредством коротких кодов, могут оказаться линейно реализуемыми посредством более длинных. Соответственно интересно изучить вопрос линейной реализуемости без ограничения на длину кода. В работе [11] было показано, что любое отображение на n -элементном множестве является линейно реализуемым посредством позиционного кодирования, причем длина кода такого кодирования равна n . В данной работе будет показано, для любого отображения на n -элементном множестве, существует кодирование, посредством которого достигается линейная реализуемость отображения и длина кода при таком кодировании равна $n - 1$.

2. Основные понятия и определения

Основным объектом изучения являются отображения на множестве E_n , где $n = 2^k$. В частности, будет изучаться как с помощью кодирования можно перейти от отображения к булеву оператору [1].

2.1. Булев оператор и его сложность

Сначала введем понятия, связанные с булевым оператором, и определим его сложность.

Определение 1. Пусть $\phi : E_2^m \rightarrow E_2^k$ — булев оператор. Его можно рассматривать как набор k булевых функций [1], зависящих от m переменных, а именно, если $\phi(\alpha_0, \alpha_1, \dots, \alpha_{m-1}) = (\beta_0, \beta_1, \dots, \beta_{k-1})$, то $f_j(\alpha_0, \alpha_1, \dots, \alpha_{m-1}) = \beta_j$, где $0 \leq j \leq k-1$. Обозначим этот набор через $\mathcal{F}_\phi = \{f_0, f_1, \dots, f_{k-1}\}$.

Пример 1. Рассмотрим оператор ϕ , заданный таблицей

x_0	x_1	x_2	x_3	y_0	y_1	y_2
0	0	0	0	0	0	1
0	0	0	1	0	1	0
0	0	1	0	0	1	1
0	0	1	1	1	0	0
0	1	0	0	1	0	1
0	1	0	1	1	1	0
0	1	1	0	0	0	0
0	1	1	1	1	1	1
1	0	0	0	0	1	1
1	0	0	1	1	1	0
1	0	1	0	0	0	1
1	0	1	1	1	0	1
1	1	0	0	1	0	0
1	1	0	1	0	1	0
1	1	1	0	1	1	1
1	1	1	1	0	0	0

Тогда последние три столбца y_0, y_1, y_2 можно рассматривать как булевы функции f_0, f_1, f_2 . Эти функции имеют следующий вид

$$f_0(x_0, x_1, x_2, x_3) = x_1 + x_2 \cdot x_3 + x_0 \cdot x_3 + x_1 \cdot x_2 + x_0 \cdot x_2 \cdot x_3 + x_0 \cdot x_1 \cdot x_2$$

$$f_1(x_0, x_1, x_2, x_3) = x_0 + x_2 + x_3 + x_0 \cdot x_1 + x_0 \cdot x_3 + x_1 \cdot x_2 + x_0 \cdot x_1 \cdot x_2 + x_0 \cdot x_1 \cdot x_3$$

$$f_2(x_0, x_1, x_2, x_3) = 1 + x_3 + x_1 \cdot x_2 + x_0 \cdot x_1 + x_0 \cdot x_1 \cdot x_3 + x_0 \cdot x_2 \cdot x_3 + x_0 \cdot x_1 \cdot x_2 + x_0 \cdot x_1 \cdot x_2 \cdot x_3$$

Определение 2. Пусть $\mathcal{F} = \{f_0, f_1, \dots, f_{k-1}\}$ — набор булевых функций, зависящих от m переменных. Данный набор определяет булев оператор $\phi_{\mathcal{F}} : E_2^m \rightarrow E_2^k$ по правилу

$$\begin{aligned} \phi_{\mathcal{F}}(\alpha_0, \alpha_1, \dots, \alpha_{m-1}) = & (f_0(\alpha_0, \alpha_1, \dots, \alpha_{m-1}), \\ & f_1(\alpha_0, \alpha_1, \dots, \alpha_{m-1}), \\ & \dots \\ & f_{k-1}(\alpha_0, \alpha_1, \dots, \alpha_{m-1})), \end{aligned}$$

где $\alpha_i \in E_2$.

Пример 2. Пусть дана пара функций $f_0(x_0, x_1, x_2, x_3) = x_0 + x_1$

x_0	x_1	x_2	x_3	f_0
0	0	0	0	0
0	0	0	1	0
0	0	1	0	0
0	0	1	1	0
0	1	0	0	1
0	1	0	1	1
0	1	1	0	1
0	1	1	1	1
1	0	0	0	1
1	0	0	1	1
1	0	1	0	1
1	0	1	1	1
1	1	0	0	0
1	1	0	1	0
1	1	1	0	0
1	1	1	1	0

и $f_1(x_0, x_1, x_2, x_3) = x_2 + x_3$

x_0	x_1	x_2	x_3	f_1
0	0	0	0	0
0	0	0	1	1
0	0	1	0	1
0	0	1	1	0
0	1	0	0	0
0	1	0	1	1
0	1	1	0	1
0	1	1	1	0
1	0	0	0	0
1	0	0	1	1
1	0	1	0	1
1	0	1	1	0
1	1	0	0	0
1	1	0	1	1
1	1	1	0	1
1	1	1	1	0

Данные функции определяют булев оператор оператор ϕ , задаваемый таблицей

x_0	x_1	x_2	x_3	y_0	y_1
0	0	0	0	0	0
0	0	0	1	0	1
0	0	1	0	0	1
0	0	1	1	0	0
0	1	0	0	1	0
0	1	0	1	1	1
0	1	1	0	1	1
0	1	1	1	1	0
1	0	0	0	1	0
1	0	0	1	1	1
1	0	1	0	1	1
1	0	1	1	1	0
1	1	0	0	0	0
1	1	0	1	0	1
1	1	1	0	0	1
1	1	1	1	0	0

Определение 3. Пусть $\phi : E_2^m \rightarrow E_2^k$ — булев оператор. Сложностью оператора назовем максимальную степень полиномов Жегалкина функций \mathcal{F}_ϕ или $L_{deg}(\phi) = \max_{f_i \in \mathcal{F}_\phi} \{deg f_i\}$

Заметим, что сложность оператора из примера 1 равна 4, а сложность оператора из примера 2 равна 1.

В предыдущих определениях предполагалось, что операторы определены на всех элементах множества E_2^m . Однако, в дальнейшем будут возникать частично-определенные операторы, т.е. операторы, определенные на подмножестве множества E_2^m . Определим понятие доопределения частично-определенного оператора.

Определение 4. Оператор $\hat{\phi} : E_2^m \rightarrow E_2^k$, $m, k \in N$ назовем доопределением оператора $\phi : R \rightarrow E_2^k$, где $R \subseteq E_2^m$, если для каждого $(\alpha_1, \dots, \alpha_m) \in R$ верно

$$\phi(\alpha_1, \dots, \alpha_m) = \hat{\phi}(\alpha_1, \dots, \alpha_m).$$

Пример 3. Рассмотрим частично-определенный оператор ϕ

x_0	x_1	x_2	y_0	y_1
0	0	0	0	0
0	0	1	0	1
0	1	0	1	0
1	0	0	1	0

Примером доопределения является оператор $\hat{\phi}$

x_0	x_1	x_2	y_0	y_1
0	0	0	0	0
0	0	1	0	1
0	1	0	1	0
0	1	1	1	1
1	0	0	1	0
1	0	1	1	1
1	1	0	0	0
1	1	1	0	1

2.2. Реализуемость отображения посредством кодирования

От отображения к булеву оператору можно перейти с помощью кодирования. Сначала определим кодирование, а затем как с помощью кодирования получается булев оператор.

Определение 5. Кодированием множества $E_n = \{0, \dots, n-1\}$ назовем взаимно-однозначное отображение (вложение) $F : \{0, \dots, n-1\} \rightarrow E_2^m$, где $m \geq \lceil \log_2 n \rceil$.

Пример 4. В качестве примера кодирования можно рассмотреть следующее отображение E_8 в E_2^3 :

q	0	1	2	3	4	5	6	7
$F(q)$	001	010	100	011	110	111	101	000

Выделим из всех кодирований «стандартное» кодирование.

Определение 6. Кодирование $F_0 : \{0, \dots, n-1\} \rightarrow E_2^k$ назовем стандартным, если код элемента есть его двоичное представление.

Пример 5. В качестве примера стандартного кодирования можно рассмотреть следующее отображение E_8 в E_2^3 :

q	0	1	2	3	4	5	6	7
$F_0(q)$	000	001	010	011	100	101	110	111

Каждому кодированию F можно сопоставить подстановку s_F на множестве $Q = \{0, \dots, n-1\}$ по правилу $s_F(i) = F_0^{-1}(F(i))$.

Кодированию F из примера 4 соответствует подстановка

$$s_F = \begin{pmatrix} 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 \\ 1 & 2 & 4 & 3 & 6 & 7 & 5 & 0 \end{pmatrix}$$

Определение 7. Пусть $s : E_n \rightarrow E_n$ — отображение множества $E_n = \{0, \dots, n-1\}$ в себя. Кодирование $F : E_n \rightarrow E_2^l$ множества E_n сопоставляет отображению s булев оператор $\phi_s^F : R \rightarrow R$, где $R \subseteq E_2^k$, по правилу

$$\phi_s^F(\alpha_1, \dots, \alpha_{l-1}) = F(s(F^{-1}(\alpha_1, \dots, \alpha_{l-1}))),$$

где $\alpha_1, \dots, \alpha_{l-1} \in E_2$.

Пример 6. Пусть задано отображение

$$p = \begin{pmatrix} 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 \\ 1 & 2 & 3 & 4 & 5 & 6 & 7 & 0 \end{pmatrix}$$

Рассмотрим кодирование

q	0	1	2	3	4	5	6	7
$F(q)$	0000	0010	0100	0111	1000	1010	1100	1111

Построим булев оператор по отображению p с использованием кодирования F . Запишем отображение p в табличном виде.

i	$p(i)$
0	1
1	2
2	3
3	4
4	5
5	6
6	7
7	0

Заменяем в таблице элементы множества E_8 на их коды, определяемые кодированием F . В результате получится следующий частично определенный булев оператор ϕ

x_0	x_1	x_2	x_3	y_0	y_1	y_2	y_3
0	0	0	0	0	0	1	0
0	0	1	0	0	1	0	0
0	1	0	0	0	1	1	1
0	1	1	1	1	0	0	0
1	0	0	0	1	0	1	0
1	0	1	0	1	1	0	0
1	1	0	0	1	1	1	1
1	1	1	1	0	0	0	0

Определение 8. *Отображение $s : E_n \rightarrow E_n$ называется линейно реализуемым посредством кодирования F , если для оператора ϕ_s^F существует такое доопределение $\widehat{\phi}_s^F$, что набор $\mathcal{F}_{\widehat{\phi}_s^F}$ состоит из линейных булевых функций.*

Пример 7. *Заметим, что оператор ϕ из примера 6*

x_0	x_1	x_2	x_3	y_0	y_1	y_2	y_3
0	0	0	0	0	0	1	0
0	0	1	0	0	1	0	0
0	1	0	0	0	1	1	1
0	1	1	1	1	0	0	0
1	0	0	0	1	0	1	0
1	0	1	0	1	1	0	0
1	1	0	0	1	1	1	1
1	1	1	1	0	0	0	0

может быть доопределен до оператора $\widehat{\phi}$, таким образом что $\mathcal{F}_{\widehat{\phi}}^F$ состоит из линейных булевых функций. Жирным шрифтом выделены наборы, на которых оператор ϕ не определен и значения оператора $\widehat{\phi}$ на этих наборах.

x_0	x_1	x_2	x_3	y_0	y_1	y_2	y_3
0	0	0	0	0	0	1	0
0	0	0	1	1	0	1	1
0	0	1	0	0	1	0	0
0	0	1	1	1	1	0	1
0	1	0	0	0	1	1	1
0	1	0	1	1	1	1	0
0	1	1	0	0	0	0	1
0	1	1	1	1	0	0	0
1	0	0	0	1	0	1	0
1	0	0	1	0	0	1	1
1	0	1	0	1	1	0	0
1	0	1	1	0	1	0	1
1	1	0	0	1	1	1	1
1	1	0	1	0	1	1	0
1	1	1	0	1	0	0	1
1	1	1	1	0	0	0	0

Причем множество $\mathcal{F}_{\hat{\phi}}^F$ состоит из функций

$$\begin{aligned} y_0 &= x_0 + x_3 \\ y_1 &= x_1 + x_2 \\ y_2 &= x_2 + 1 \\ y_3 &= x_1 + x_3 \end{aligned}$$

Следовательно подстановка p из примера 6 является линейной реализуемой посредством кодирования F .

2.3. Понятие линейризующего кодирования

Среди всех кодирований выделим специальное кодирование, которое назовем линейризующим.

Определение 9. Пусть задан вектор $(q_0, \dots, q_{k-1}) \in E_2^k$. Множеством единиц $\mathbb{I}_{q_0, \dots, q_{k-1}}$ назовем множество всех индексов $\{i_1, \dots, i_l\}$ таких, что $j \in \{i_1, \dots, i_l\}$ тогда и только тогда, когда $q_j = 1$.

Пример 8. Пусть задан вектор (0110) . Тогда $\mathbb{I}_{(0110)} = \{12\}$.
0123

Определение 10. Полиномиальным оператором назовем отображение $\phi_{pol} : E_2^k \rightarrow E_2^{2^k-1}$, где i -ая компонента результата ϕ_{pol} определяется соотношением

$$\phi_{pol_i}(q_0, \dots, q_{k-1}) = q_{j_1} \cdot q_{j_2} \cdot \dots \cdot q_{j_l},$$

где $(j_1, j_2, \dots, j_l) = \mathbb{I}_{F_0(i+1)}, F_0 : E_n \rightarrow E_2^k$ - стандартное кодирование, $i \in E_{2^k-1}$.

Пример 9. Оператор $\phi_{pol} : E_2^3 \rightarrow E_2^7$ имеет вид

x_0	x_1	x_2	y_0 {2}	y_1 {1}	y_2 {12}	y_3 {0}	y_4 {02}	y_5 {01}	y_6 {012}
0	0	0	0	0	0	0	0	0	0
0	0	1	1	0	0	0	0	0	0
0	1	0	0	1	0	0	0	0	0
0	1	1	1	1	1	0	0	0	0
1	0	0	0	0	0	1	0	0	0
1	0	1	1	0	0	1	1	0	0
1	1	0	0	1	0	1	0	1	0
1	1	1	1	1	1	1	1	1	1

Определение 11. Пусть задано кодирование $F : E_{2^k} \rightarrow E_2^k$. Обозначим через $L_F : E_{2^k} \rightarrow E_2^{2^k-1}$ линеаризующее кодирование, задаваемое соотношением по формуле:

$$L_F(q) = \phi_{pol}(F(q)), q \in E_{2^k}.$$

Заметим, что отображение $L_F : E_{2^k} \rightarrow S$ является взаимнооднозначным, т.е. из $q \neq q'$ следует $L_F(q) \neq L_F(q')$. Другими словами данное отображение может быть использовано в качестве кодирования. Это следует из определения, если $q \neq q'$, то $F(q) \neq F(q')$ в силу определения кодирования, а значит и $L_F(q)$ и $L_F(q')$ отличаются.

Пример 10. Пусть задано стандартное кодирование $F_0 : E_8 \rightarrow E_2^3$

q	0	1	2	3	4	5	6	7
$F_0(q)$	000	001	010	011	100	101	110	111

Тогда кодирование $L_F : E_8 \rightarrow E_2^7$ задается как

q	0	1	2	3	4	5	6	7
$L_F(q)$	0000000	0000001	0000010	0000111	0001000	0011001	0101010	1111111

3. Линейная реализуемость отображений.

Теорема 1. Пусть задано отображение $p : E_n \rightarrow E_n$, где $n = 2^k$ и кодирование $F : E_n \rightarrow E_2^k$. Отображение p является линейно реализуемым посредством линеаризующего кодирования L_F

Доказательство. Построим по отображению p и кодированию L_F булев оператор $\phi_p^{L_F}$.

$$\begin{aligned} \phi_p^{L_F}(q_0, \dots, q_{n-2}) &= L_F(p(L_F^{-1}(q_0, \dots, q_{n-2}))) \\ &= \phi_{pol}(F(p(F^{-1}(\phi_{pol}^{-1}(q_0, \dots, q_{k-1}))))), \end{aligned}$$

где $(q_0, \dots, q_{n-2}) \in R = Im(L_F)$. Схематически данное отображение можно представить следующим образом

$$E_2^{n-1} \supset R \xrightarrow{\phi_{pol}^{-1}} E_2^k \xrightarrow{F^{-1}} E_n \xrightarrow{p} E_n \xrightarrow{F} E_2^k \xrightarrow{\phi_{pol}} R \subset E_2^{n-1}.$$

Из определения отображения ϕ_{pol} следует, что $\forall (q_0, \dots, q_{n-2}) \in Im(L_F)$, $\phi_{pol}^{-1}(q_0, \dots, q_{n-2}) = (q'_0, \dots, q'_{k-1})$, где $(q'_0, \dots, q'_{k-1}) \in E_2^k$. Следовательно,

$$\phi_{pol}(F(p(F^{-1}(\phi_{pol}^{-1}(q_0, \dots, q_{n-2})))))) = \phi_{pol}(F(p(F^{-1}(q'_0, \dots, q'_{k-1}))))).$$

Можно заметить, что булев оператор ϕ_p^F , построенный по отображению p с помощью кодирования F , задается соотношением $\phi_p^F(q'_0, \dots, q'_{k-1}) = F(p(F^{-1}(q'_0, \dots, q'_{k-1})))$. Следовательно,

$$\phi_{pol}(F(p(F^{-1}(q'_0, \dots, q'_{k-1})))) = \phi_{pol}(\phi_p^F(q'_0, \dots, q'_{k-1}))).$$

Согласно определению 7 оператор ϕ_p^F можно рассматривать как набор k булевых функций, зависящих от k переменных. Обозначим этот набор через $\mathcal{F}_p(F)$. Наше соотношение может быть переписано как

$$\phi_{pol}(\phi_p^F(q'_0, \dots, q'_{k-1}))) = \phi_{pol}(f_0(q'_0, \dots, q'_{k-1}), \dots, f_{k-1}(q'_0, \dots, q'_{k-1}))),$$

где $f_i \in \mathcal{F}_p(F)$. Согласно определению отображения ϕ_{pol} i -ая координата результата равна

$$\begin{aligned} & \phi_{pol_i}(f_0(q'_0, \dots, q'_{k-1}), \dots, f_{k-1}(q'_0, \dots, q'_{k-1}))) = \\ & f_{j_1}(q'_0, \dots, q'_{k-1}) \cdot f_{j_2}(q'_0, \dots, q'_{k-1}) \cdot \dots \cdot f_{j_i}(q'_0, \dots, q'_{k-1}), \end{aligned}$$

где $\{j_1, j_2, \dots, j_i\} = \mathbb{I}_{F_0(i+1)}$, $F_0 : E_n \rightarrow E_k^2$. Произведение булевых функций $f_{j_1}(q'_0, \dots, q'_{k-1}) \cdot f_{j_2}(q'_0, \dots, q'_{k-1}) \cdot \dots \cdot f_{j_i}(q'_0, \dots, q'_{k-1})$ есть булева функция. Обозначим ее $g(q'_0, \dots, q'_{k-1})$. Данная функция может быть представлена в виде полинома Жегалкина.

$$g(q'_0, \dots, q'_{k-1}) = \sum_{\{q'_{j_0} q'_{j_1} \dots q'_{j_i}\} \in 2^{E_n}} a_{\{q'_{j_0} q'_{j_1} \dots q'_{j_i}\}} \cdot q'_{j_0} \cdot q'_{j_1} \cdot \dots \cdot q'_{j_i},$$

где $a_R \in E_2$, $R \subset E_n$. Заметим, что $q'_{j_0} \cdot q'_{j_1} \cdot \dots \cdot q'_{j_i}$ есть t -ая компонента оператора ϕ_{pol} или в наших обозначениях q_t , где $\{q'_{j_0} q'_{j_1} \dots q'_{j_i}\} = \mathbb{I}_{F_0(t+1)}$. Обозначим коэффициент $a_{\{q'_{j_0} q'_{j_1} \dots q'_{j_i}\}}$ через a_t . Следовательно, i -ая координата оператора ϕ_p^{LF} задается как

$$\phi_p^{LF_i}(q_0, \dots, q_{n-2}) = \sum_t a_t \cdot q_t,$$

где $(q_0, \dots, q_{n-2}) \in R$. Доопределим данный оператор до оператора $\widehat{\phi}_p^{LF}$, i -ая координата которого задается формулой

$$\widehat{\phi}_p^{LF_i}(q_0, \dots, q_{n-2}) = \sum_t a_t \cdot q_t,$$

для любого вектора $(q_0, \dots, q_{n-2}) \in E_2^{n-2}$. Можно видеть, что $\mathcal{F}_{\widehat{\phi}_p^{LF}}$ состоит из линейных функций. Следовательно, отображение p линейно реализуемо посредством кодирования L_F . \square

В заключении автор выражает благодарность Алёшину Станиславу Владимировичу и Носову Михаилу Васильевичу за многочисленные обсуждения и советы, которые позволили получить результаты, изложенные в данной работе.

Список литературы

- [1] Яблонский С.В., *Введение в дискретную математику*, Наука, Москва.
- [2] Кудрявцев В.Б., Алёшин С.В., Подколзин А.С., *Введение в теорию автоматов*, «Наука», Москва, 1985, 320 с.
- [3] А. Клиффорд, Г. Престон, *Алгебраическая теория полугрупп*, **1**, Мир, Москва.
- [4] Р. Лидл, Г. Нидеррайтер, *Конечные поля*, Мир, Москва.
- [5] М.И. Каргаполов, Ю.И. Мерзляков, *Основы теории групп*, **3**, Наука, Москва.
- [6] Алёшин С.В., *Алгебраические системы автоматов*, МАКС Пресс, Москва, 2016.
- [7] М.А. Арбиб, *Алгебраическая теория автоматов, языков и полугрупп*, «Статистика», Москва.
- [8] Родин С.Б., “Переходные системы с максимальной вариантностью относительно кодирования состояний”, *Интеллектуальные системы*, **4:3-4**, 335–352.
- [9] Родин С.Б., “О связи линейно реализуемых автоматов и автоматов с максимальной вариативностью относительно кодирования состояний”, *Интеллектуальные системы*, **20:2**, 337–347.
- [10] Родин С.Б., “Линейно реализуемые автоматы”, *Дискретная математика*, **29:1** (2016), 59–79.
- [11] Родин С.Б., “О свойствах кодирования состояний автомата”, *Интеллектуальные системы*, **21:1**, 97–111.

On the linear realizability property of the mappings Sergey Rodin

This paper studies the property of linear realizability of mapping of the finite set into itself. This property is important from linear realizability of automata, namely linear realizability of the elements of the generating set of the automaton inner semigroup is the one of the necessary conditions for linear realizability of the automaton. Previously it was shown that every mapping of the finite set into itself is linear realizable via an encoding which code length is equal the finite

set cardinality. In this paper this result will be improved and it will be shown that every mapping of the finite set into itself is linear realizable via an encoding which code length is equal the finite set cardinality minus one.

Keywords: Automata theory, semiautomata, transition systems, assignment, state encoding, complexity, boolean operator

References

- [1] Yablonskij S.V., *Introduction to the discrete math*, Nauka, Moscow.
- [2] Kudryavtsev V.B., Alyoshin S.V., Podkolzin A.S., *Introduction to automata theory*, Nauka, Moscow, 1985, 320 c.
- [3] Clifford A.H., Preston G.B., *The algebraic theory of semigroups*, **1**, Mir, Moscow.
- [4] R. Lidl, H. Niederreiter, *Finite fields*, Mir, Moscow.
- [5] Kargapolov M.I., Merzlyakov Yu. I., *Basics of group theory*, **3**, Nauka, Moscow.
- [6] Alyoshin S.V., *Algebraic automata systems*, MAKS Press, Moscow, 2016.
- [7] M.A. Arbib, *Algebraic theory of machines, languages and semigroups*, «Statistika», Moscow.
- [8] Rodin S.B., “The most variable semiautomata with respect to the states encoding”, *Intelligent systems*, **4:3-4**, 335–352.
- [9] Rodin S.B., “On relation between the linearly realizable automata and the most variable automata with respect to the states encoding”, *Intelligent systems*, **20:2**, 337–347.
- [10] Rodin S.B., “Linearly realizable automata”, *Discrete Math*, **29:1** (2016), 59–79.
- [11] Rodin S.B., “On automata states encoding properties”, *Intelligent systems*, **21:1**, 97–111.

**К сведению авторов публикаций в журнале
«Интеллектуальные системы. Теория и приложения»**

В соответствии с требованиями ВАК РФ к изданиям, входящим в перечень ведущих рецензируемых научных журналов и изданий, в которых могут быть опубликованы основные научные результаты диссертаций на соискание ученой степени доктора и кандидата наук, статьи в журнал «Интеллектуальные системы. Теория и приложения» предоставляются авторами в следующей форме:

1. Статьи, набранные в пакете \LaTeX , предоставляются к загрузке через WEB-форму http://intsysmagazine.ru/generator_form .

2. К статье прилагаются файлы, содержащие название статьи на русском и английском языках, аннотацию на русском и английском языках (не более 50 слов), список ключевых слов на русском и английском языках (не более 20 слов), информация об авторах: Ф.И.О. полностью, место работы, должность, ученая степень и/или звание (если имеется), для аспирантов ФИО научного руководителя, контактные телефоны (с кодом города и страны), e-mail, почтовый адрес с индексом города (домашний или служебный).

3. Список литературы оформляется в едином формате, установленном системой Российского индекса научного цитирования. Список на русском языке приводится в конце файла с текстом статьи, в то время как список, переведённый на английский язык, прилагается отдельным файлом.

4. За публикацию статей в журнале «Интеллектуальные системы. Теория и приложения» с авторов (в том числе аспирантов высших учебных заведений) статей, рекомендованных к публикации, плата не взимается. Авторам бесплатно предоставляется номер журнала, в котором вышла статья. Журнал распространяется по подписке, экземпляры журнала рассылаются подписчикам наложенным платежом. Условия подписки публикуются в каталоге НТИ «Роспечать», индекс журнала 64559.

5. Доступ к электронной версии последнего вышедшего номера осуществляется через НЭБ «Российский индекс научного цитирования». Номера, вышедшие ранее, размещаются на сайте

<http://intsysmagazine.ru>,

и доступ к ним бесплатный. Там же будут размещены полные тексты всех публикуемых статей.

Подписано в печать: 20.03.2024

Дата выхода: 29.03.2024

Тираж: 200 экз.

Цена свободная

Свидетельство о регистрации СМИ: ПИ № ФС77-58444 от 25 июня 2014 г.,
выдано Федеральной службой по надзору в сфере связи, информационных
технологий и массовых коммуникаций(Роскомнадзор).