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 4 * **2024**

МОСКВА

УДК 519.95; 007:159.955 ISSN 2411–4448
ББК 32.81 Издаётся с 1996 г.

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» МАТН.
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».

ОГЛАВЛЕНИЕ

Часть 1
Общие проблемы теории интеллектуальных систем

Автоматизированное построение реалистичных литофациальных карт методами комбинаторной оптимизации

А. П. Антонов[1] С. А. Афонин[2] А. С. Козицын[3] В. М. Староверов[4] А. В. Ступакова[5] А. А. Суслова[6] А. П. Завьялова[7] В. В. Чупахина[8] Р. С. Сауткин[9] С. В. Осипов[10]

[1] *Антонов Алексей Петрович* — доцент каф. математического анализа мех.-мат. ф-та МГУ, e-mail: alexey.p.antonov@gmail.com.

Antonov Alexei Petrovich — assoc. prof., Lomonosov Moscow State University, Faculty of Mechanics and Mathematics, Chair of Mathematical Analysis.

[2] *Афонин Сергей Александрович* — ведущий научный сотрудник лаб. автоматизации экспериментальных исследований НИИ механики МГУ, e-mail: serg@msu.ru

Afonin Sergey Aleksandrovich — leading researcher, Lomonosov Moscow State University, Institute of Mechanics, Experimental research automation Laboratory.

[3] *Козицын Александр Сергеевич* — ведущий научный сотрудник лаб. автоматизации экспериментальных исследований НИИ механики МГУ, e-mail: alexanderkz@mail.ru

Kozitsyn Alexander Sergeevich — leading researcher, Lomonosov Moscow State University, Institute of Mechanics, Experimental research automation Laboratory.

[4] Староверов Владимир Михайлович, — доцент каф. вычислительной математики мех.-мат. ф-та МГУ, e-mail: staroverovvl@yandex.ru

Staroverov Vladimir Mihailovich — assoc. prof., Lomonosov Moscow State University, Faculty of Mechanics and Mathematics, Chair of Numerical Mathematics.

[5] *Ступакова Антонина Васильевна* — зав. кафедры каф. геологии и геохимии горючих ископаемых геологического ф-та МГУ, e-mail: a.stoupakova@oilmsu.ru

Stupakova Antonina Vasilievna — head of the department, Lomonosov Moscow State University, Faculty of Geology, Chair of Geology and Geochemistry of Oil and Gas.

[6] *Суслова Анна Анатольевна* — ведущий научный сотрудник каф. геологии и геохимии горючих ископаемых геологического ф-та МГУ, e-mail: a.suslova@oilmsu.ru

Suslova Anna Anatolievna — leading researcher, Lomonosov Moscow State University, Faculty of Geology, Chair of Geology and Geochemistry of Oil and Gas.

[7] *Завьялова Анна Петровна* — научный сотрудник каф. геологии и геохимии горючих ископаемых геологического ф-та МГУ, e-mail: a.zavyalova@oilmsu.ru

Zavyalova Anna Petrovna — researcher, Lomonosov Moscow State University, Faculty of Geology, Chair of Geology and Geochemistry of Oil and Gas.

[8] *Чупахина Виталия Валерьевна* — инженер каф. геологии и геохимии горючих ископаемых геологического ф-та МГУ, e-mail: v.chupakhina@oilmsu.ru

Chupakhina Vitalia Valerievna — engeneer, Lomonosov Moscow State University, Faculty of Geology, Chair of Geology and Geochemistry of Oil and Gas.

[9] *Сауткин Роман Сергеевич* — старший научный сотрудник каф. геологии и геохимии горючих ископаемых геологического ф-та МГУ, e-mail: r.sautkin@oilmsu.ru

Sautkin Roman Sergeevich — senior researcher, Lomonosov Moscow State University, Faculty of Geology, Chair of Geology and Geochemistry of Oil and Gas.

[10] *Осипов Сергей Владимирович* — менеджер ДНТРиИ, ПАО «НК «Роснефть» s_osipov@rosneft.ru

Литофациальные карты отображают пространственное изменение литологического состава пород и направление характера их замещения для определенного геологического времени в зависимости от физико-географических условий их седиментации. Построение литофациальных карт проводится на основании анализа комплексов генетически связанных отложений, обладающих характерными вещественными признаками литофаций. Исходными данным могут быть как *только* точки на плоскости — координаты скважин, в которых встречаются определенные литофации, — так и *дополнительно* карта скоростей накопления осадков. Предполагается, что одинаковые фации должны иметь близкие скорости осадконакопления. Задача построения литофациальной карты заключается в определении границ областей распространения фаций. В работе предлагается сведение задачи построения карты к задаче раскраски графов, которая, в свою очередь, решается методами целочисленного линейного программирования. Такой подход позволяет автоматически строить реалистичные карты — то есть карты, удовлетворяющей экспертным ограничениям и правилам.

Ключевые слова: картопостроение; лито-фациальный анализ; палереконструкция; оптимизация; целочисленное программирование

1. Введение

В настоящее время литолого-фациальный анализ является одним из наиболее широко распространенных и важных методов геологического анализа в нефтегазовой геологии. Практическое и теоретическое значение литофациальных карт очень велико, с их помощью можно спрогнозировать закономерности распределения фаций в межскважинном пространстве, определить области отсутствия осадконакопления, восстановить историю геологического развития исследуемого региона, а понимание распределения литофаций по площади позволяет выделить районы, благоприятные для накопления нефтегазоматеринских толщ, коллекторов и флюидоупоров [1, 2]. Использование литофациальных карт совместно с другими данными позволяет осуществлять прогноз и поиски залежей нефти и газа. Автоматизация рабочего процесса создания карт позволит сэкономить время на обработке большого количества данных, заменив «ручной труд» на машинный, и исключить системные ошибки.

Под термином «литофация» понимается комплекс генетически связанных отложений, обладающих характерными вещественными признаками [3]. По сути данный термин включает в себя два понятия — «лито-» и «фация-», то есть литофация характеризует, во-первых, вещественный

Osipov Sergey Vladimirovich — manager, Rosneft, Department of Scientific and Technical Development and Innovation.

состав ассоциации горных пород, которая обладает определенными признаками, отличающими их от соседних или ассоциирующих с ними пород, и, во-вторых, условия их образования.

Построение литофациальных карт — это способ отображения современного распространения пород, со сходными вещественными и структурно-текстурными особенностями, и условия их формирования в пределах определенного стратиграфического уровня. В его основе лежит знание о взаимосвязи литологических особенностей отложений, характера их распространения, взаимопереходов и обстановок осадконакопления [4].

Исходными данными для картопостроения являются точки на плоскости с уже определенными литофациями для каждого возрастного интервала в разрезе скважин. Определение литофаций осуществляется геологом по литологическому описанию кернового материала и выявлению в нем диагностических признаков (состав, структура, текстура, включения, цвет, границы). Дополнительными данными могут выступать карты с изолиниям — мощностей и/или скоростей осадконакопления. Предлагаемые алгоритмы построения реалистичных литофациальных карт методами комбинированной оптимизации позволяют автоматически, без ручного труда геолога, получать реалистичные и достаточно подробные карты, с учетом таких особенностей, как направление углубления бассейна, сохранение последовательности смены литофаций, ограничение распространений некоторых литофаций и особенности конфигурации осадочных тел.

Современные методы автоматического построения литологических и литофациальных карт основаны на применении алгоритмов машинного обучения для определения литологии либо в скважине по данным каротажа и изображений керна, либо в заданной точке пространства по известным значениям геофизических или визуальных параметров. Например, в [5] литология определяется по 16 геофизическим параметрам и данным зондирования поверхности методом случайного леса. В [6] рассматривается задача сегментирования аэрофотоснимков поверхности. Практическая важность таких работ связана с относительной простотой получения исходных данных, в том числе, для труднодоступных районов. Обзор методов анализа данных зондирования можно найти в [7]. Методы классического машинного обучения не тестовых данных показывают точность порядка 75%, однако в большинстве работ речь идёт об определении литологии на поверхности.

В данной работе мы предполагаем, что литофации уже были определены каким-либо методом и требуется определить области распространения этих литофаций в межскважном пространстве. Отличительной особенностей предлагаемого подхода является использование допустимых последовательностей смены литофаций.

2. Постановка задачи

Задача состоит в построении карты распространения литофаций в заданной области. Исходными данными являются:

- множество возможный фаций F;
- множество возможных литологий L;
- область (полигон) распространения $\overline{P} \subset \mathbb{R} \times \mathbb{R}$ конечной площади;
- координаты скважин W с определенными в них фацией и литологией, то есть $W = \{\langle x_i, y_i \rangle\}_{i=1}^{N}$, $f : W \to F$, $l : W \to L$;
- карта скоростей осадконакопления $z : \mathbb{R} \times \mathbb{R} \to \mathbb{R}$, определенная в области распространения $\overline{P}$;
- допустимые последовательности фаций $\mathcal{F} = \{\langle f_1^k, \dots f_{n_k}^k \rangle\}_{k=1}^{ncf}$, $f_i^k \in F$;
- допустимые последовательности литологий $\mathcal{L} = \{\langle l_1^k, \dots l_{n_k}^k \rangle\}_{k=1}^{ncl}$, $l_i^k \in L$;
- допустимые литологии для фаций $VL \subseteq F \times L$.

Последовательности фаций определяют, какие фации могут соседствовать на карте. Последовательности литологий, соответственно, определяют возможные границы между областями распространения типов пород.

При построении карты используются следующие принципы:

- скважины считаются подтвержденными данными;
- границы областей распространения фаций выравниваются по линям уровня карты скоростей z;
- запрещается наличие общей границы у областей распространения фаций f_1 и f_2, если пара f_1, f_2 не встречается хотя бы в одной последовательности из $\mathcal{F}$.

Особенностью задачи является возможное отсутствие в исходных данных W скважин с фацией, которая должна быть на результирующей карте. Например, если в $\mathcal{F}$ есть последовательность f_1, f_2, f_3 (и нет последовательностей, содержащих f_1, f_3), а в W есть только скважины с фациями f_1 и f_3, то между областями распространения f_1 и f_3 должна появиться область фации f_2.

Требование выполнения корректности попарных границ является обязательным. При построении карты предпочтения отдаются таким картам, в которых возникают длинные последовательности из $\mathcal{F}$.

Результатом работы алгоритма являются фациальная и литофациальные карты (здесь $Poly$ обозначает множество всех полигонов, а 2^{Poly} — множество всех наборов полигонов)

$$\mathcal{M}_F : F \to 2^{Poly} \tag{1}$$

$$\mathcal{M} : F \times L \to 2^{Poly}, \tag{2}$$

удовлетворяющие следующим условиям:

- полигоны фациальной карты $\mathcal{M}_F$ не пересекаются:

$$\forall f_1, f_2 \in F \; f_1 \neq f_2 \to \mathcal{M}_F(f_1) \cap \mathcal{M}(f_2) = \varnothing; \tag{3}$$

- полигоны фациальной карты $\mathcal{M}_F$ покрывают всю область распространения $\overline{P}$:

$$\bigcup_{f \in F} \mathcal{M}_F(f) = \overline{P}; \tag{4}$$

- литофациальная карта $\mathcal{M}$ является разбиением фациальной карты $\mathcal{M}_F$:

$$\forall f \in F \quad \mathcal{M}_F(f) = \bigcup_{l \in L} \mathcal{M}(f, l); \tag{5}$$

- соблюдается корректность границ фациальной карты $\mathcal{M}_F$:

$$\forall f_1 \in F, f_2 \in F \quad \operatorname{dist}(\mathcal{M}_F(f_1), \mathcal{M}_F(f_2)) = 0 \to \exists ch \in \mathcal{F} \; \langle f_1, f_2 \rangle \in ch; \tag{6}$$

- согласованность карт с исходными данными — скважина w принадлежит полигону фации $f(w)$ и полигону литофациальной карты для пары $f(w), l(w)$:

$$\forall w \in W \quad w \in \mathcal{M}_F(f(w)) \tag{7}$$

$$\forall w \in W \quad w \in \mathcal{M}(f(w), l(w)). \tag{8}$$

Здесь объединение, пересечение и расстояние dist между полигонами естественным образом расширяется на наборы полигонов (мультиполигоны). Выражение $\langle f_1, f_2 \rangle \in ch$ означает, что цепочке $ch = \langle x_1, x_2, \ldots, x_k \rangle$ найдется индекс i, такой что либо $f_1 = x_i$ и $f_2 = x_{i+1}$, либо $f_2 = x_i$ и $f_1 = x_{i+1}$.

3. Алгоритм построения $\mathcal{M}_F$

На верхнем уровне алгоритма построения фациальной карты разделяется на следующих этапы.

Сначала для каждой скважины $w \in W$ находится полигон P_w, ограниченный линиями уровня карты скоростей z, в который не попадают скважины с другими фациями. Если полигон P_w содержит скважину w' той же фации $f(w) = f(w')$, то полигоны P_w и $P_{w'}$ объединяются, и одна из скважин w и w' выбирается в качестве базовой (скважина-представитель) для этого полигона. Полигон P_w может быть ограничен двумя линиями уровня z_{min}, z_{max} («полоса»), либо одной линией («остров»). Во втором случае считаем, что одно из значений бесконечно («холм» имеет граничные значения $z_{min}, +\infty$). Линия уровня, разделяющая полигоны скважин w_1 и w_2 разных фаций, выбирается из условия $z_{div} = (z(w_1) + z(w_2))/2$.

Далее полигоны P_w разбиваются на две или три части: «ядро» P_w^c и области неопределённости P_w^{uh} и P_w^{ul}. Ядро определяет ту часть P_w, для которой фация $f(w)$ считается достоверной. Области неопределённости — эти дополнение ядра. Если P_w является полосой, то возникает область неопределённости P_w^{uh} у верхней границы z_{max}, и область P_w^u у нижней границы.

Пример полигонов приведен на рис. 1. Видно, что со стороны бóльших значений z область ограничена скважиной в правом верхнем углу, со стороны меньших — скважинами в нижней части рисунка.

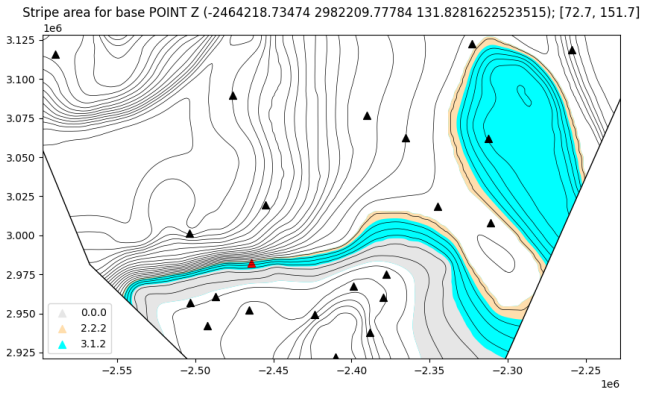


Рис. 1. Пример P_w, P_w^c. Базовая скважина выделена красным. Ядро области отмечено цветом 3.1.2, верхняя область неопределенности P_w^{uh} — цветом 2.2.2, нижняя область P_w^{ul} — цветом 0.0.0. Изолинии поля скоростей z нарисованы в границах полигона распространения $\overline{P}$.

Из полигонов P_w^c, P_w^{uh}, P_w^{ul}, соответствующих всем скважинам-представителям w, строится граф G. Вершина — полигон, ребро — наличие общей границы. Некоторым вершинам графа приписан цвет (название фации). Это вершины, соответствующие ядрам P_w^c. Остальные вершины цвета не имеют. Задача построения карты сводится к выбору цвета для всех вершин графа.

Наличие области неопределенности позволяет поставить на границу межу двумя областями до двух дополнительных фаций, которые могли отсутствовать в исходных данных.

Раскраска графа делается путем решения задачи смешанного целочисленного программирования [9]: найти корректную раскраску (выполняется условие (6)), для которой функция качества окажется максимальной. Функция качества старается минимизировать количество смен цветов, находить длинные последовательности фаций, принадлежащих одной цепочке $\mathcal{F}$, окрашивать вершины в один цвет при условии близости значений z.

3.1. Построение полигонов P_w, P_w^c

Считаем, что по карте скоростей z, скважине w в точке (x, y) и значениям z_{min}, z_{max}, таким что $z_{min} \leqslant z(x, y) \leqslant z_{max}$, можно найти область (полигон), ограниченную линиями уровней z_{min}, z_{max}, и содержащую точку (x, y). Обозначим эту область $S_w(z_{min}, z_{max})$, или просто S_w.

Пусть выбрана опорная скважина w, для которой требуется построить полигон P_w и его разбиение на P_w^c, P_w^{uh} и P_w^{ul}. Все скважины других фаций разбиваются на два множества (с меньшими и бóльшими значениями z):

$$W^- = \{w' \in W : f(w') \neq f(w) \wedge z(w') < z(w)\} \tag{9}$$

$$W^+ = \{w' \in W : f(w') \neq f(w) \wedge z(w') > z(w)\} \tag{10}$$

По множеству W^- строится множество значений z, из которых выбирается z_{min}, для которого выполняются следующие условия:

- полоса $S_w(z_{min}, z(w))$ не содержит скважин других фаций;
- для меньших значений предыдущее условие нарушается.

$$Z^- = \left\{ \frac{z(w) + z(q)}{2} \right\}_{q \in W^-} \cup \{-\infty\} \tag{11}$$

$$z_{min} = \min \left\{ z \in Z^- : \forall q \in W \; f(q) \neq f(w) \to q \notin S_w(z, z(w)) \right\}. \tag{12}$$

Аналогично выбирается значение z_{max}.

$$Z^+ = \left\{ \frac{z(w) + z(q)}{2} \right\}_{q \in W^+} \cup \{+\infty\} \tag{13}$$

$$z_{max} = \max \left\{ z \in Z^+ : \forall q \in W \ f(q) \neq f(w) \to q \notin S_w(z(w), z) \right\}. \tag{14}$$

Таким образом, полигон $P_w = S_w(z_{min}, z_{max})$ имеет максимально возможную ширину по z, при условии, что уровни выбираются из построенных множеств. Заметим, что S_w строится с учётом структуры карты скоростей, поэтому на карте могут найтись скважины q_i других фаций, для которых $z_{min} < z(q_i) < z_{max}$, но которые геометрически находятся в другой области.

В случае, когда в область S_w попадают другие скважины фации $f(w)$, вычисления (11)-(14) повторяются с заменой значения $z(w)$ в формулах (11) и (13) на минимальное и максимальное значение скорости по этим скважинам. Пусть $Q = \{q \in W : f(q) = f(w) \wedge q \in S_w(z_{min}, z_{max})\}$. Тогда в (11) значение $z(w)$ заменяется на $\min_{q \in Q} z(q)$, а в (13) — на $\max_{q \in Q} z(q)$.

Ядро P_w^c ограничено z_{min}^c, z_{max}^c, где $z_{min} \leqslant z_{min}^c = (z_{min} + z(w))/2$, а $z_{max} \geqslant z_{max}^c = (z_{max} + z(w))/2$. Пропорция, в которой диапазон скоростей разделяется на область ядра и области неопределенности, а также количество областей неопределенности с каждой стороны, могут быть произвольными.

3.2. Оптимизационная задача

Задачу о раскраске графа можно сформулировать в терминах смешанного целочисленного программирования (MILP). Пусть требуется раскрасить граф $G = \langle V, E \rangle$ цветами K, то есть найти отображение $C : V \to K$, с соблюдением ограничений корректности. Рассмотрим для наглядности случай, когда заданы только пары некорректных цветов $I \subseteq K \times K$. В графе не может быть смежных вершин u и v, для которых $(C(u), C(v)) \in I$. Требуется найти корректную раскраску с минимальным количеством используемых цветов.

Введем следующие бинарные переменные, определенные для $u, v \in V$, $k \in K$:

- x_{uk}: принимает значение 1 если вершина u покрашена в цвет k, и 0 иначе;
- z_{uvk}: принимает значение 1 тогда и только тогда, когда $(u, v) \in E$, вершина u покрашена в цвет k, а v покрашена в другой цвет;
- w_k: равная 1, если цвет k используется, и 0 иначе.

Тогда искомая раскраска может быть найдена как решение следующей оптимизационной задачи, где минимум ищется по всем значениями переменных x_{uk}, z_{uvk} и w_k.

$$\text{minimize} \sum_{k \in K} w_k + \sum_{u,v \in V, k \in K} z_{uvk} \tag{15}$$

$$\text{при усл.} \sum_{k \in K} x_{vk} = 1 \qquad \forall v \in V \tag{16}$$

$$x_{vi} + x_{uj} \leqslant 1 \qquad \forall (u,v) \in E, (i,j) \in I \tag{17}$$

$$z_{uvk} \geqslant \sum_{k' \in K, k' \neq k} x_{vk'} - \sum_{k' \in K, k' \neq k} x_{uk'} \qquad \forall (u,v) \in E, k \in K \tag{18}$$

$$x_{vk} \leqslant w_k \qquad \forall v \in V, k \in K \tag{19}$$

$$x_{vk}, z_{uvk} \in \{0,1\} \qquad \forall v \in V, k \in K, u, v \in E \tag{20}$$

Условие (16) означает, что каждая вершина должна быть покрашена в какой-то один цвет. Корректность раскраски обеспечивается условием (17): если смежные вершины будут покрашены в несовместимые цвета, то сумма переменных в левой части соотношения будет равна двум.

Условия (18) задаются для каждого ребра. Переменная в левой части должна быть равна 1, если вершина u покрашена в цвет k. Первая сумма в правой части обращается в 1 только в том случае, когда v покрашена в цвет, отличный от k. Вторая сумма равна нулю только при условии, что u покрашена в цвет k. Таким образом, выражение справа имеет значение 1 тогда и только тогда, когда u покрашена в k, а v — в любой другой цвет. В этом случае z_{uvk} обязательно будет равно единице. Избежать тожественного равенства единице всех переменных z — в этом случае ограничения (18) выполняются — позволяет второе слагаемое целевой функции (15).

Условие (19) требует, чтобы переменная w_k была равна единице, если цвет присутствует в раскраске. Случай тождественного равенства единице всех переменных w исключается первым слагаемым целевой функции.

Фиксированные цвета вершин (вершин, соответствующих полигонам P_w^c) могут быть заданы ограничениями вида

$$x_{vk} = 1 \qquad \forall (v,k) \in Fix \subseteq V \times K, \tag{21}$$

где Fix — начальная (частичная) раскраска графа.

Направления углубления бассейна. Некоторые вершины могут быть объединены в последовательности, соответствующие направлениям углубления бассейна. Вычисление этих направлений описано в разделе 3.3.

Критерии качества. Задача (15)-(20) позволяет находить корректные раскраски с минимальным количеством цветов. Множества K и I могут быть построены из цепочек $\mathcal{F}$:

$$K \text{ — все фации}, I = \{(f_1, f_2) \in K \times K \mid \forall ch \in \mathcal{F} \ \langle f_1, f_2 \rangle \notin ch\},$$

то есть I состоит из пар фаций, которые не встречаются в цепочках подряд. Среди множества корректных раскрасок требуется выбрать «хорошую». Используются следующие критерии качества.

- Минимизация количества смен цветов.
- Минимизация количества цепочек, цвета которых используются для раскраски вдоль направления углубления бассейна.
- Сохранение площади, занятой некоторым цветами (фациями-исключениями).
- Выбор цветов вершины из некоторого множества («подсказок»).

Следует отметить, что формального определения «хорошей» раскраски нет. Оценка качества определяется экспертом, и возможный сценарий использования алгоритма построения карты предполагает её дальнейшую модификацию пользователем. Перечисленные формальные критерии для набора тестовых примеров приводят к построению карт, требующих минимальной коррекции.

Структура целевой функции. Для оптимизации с упорядоченным по важности набором критериев $g_1, \ldots, g_k$ целевая функция может быть задана в виде $c_1 g_1 + c_2 g_2 + \ldots c_k g_k$, где весовые коэффициенты выбраны таким образом, что оптимальное решение будет иметь минимальное значение g_1. Если с таким значением существует несколько корректных раскрасок, то из них будет выбрана раскраска с минимальным значением g_2, и так далее.

3.3. Направления углубления бассейна

Последовательности скважин, свидетельствующие о предполагаемом направлении углубления бассейна, строятся следующим образом. По исходным скважинам W строится граф G_w, вершинами которого служат

скважины, и каждая вершина соединена с $h = 3$ ближайшими. Весом ребра является геометрическое расстояние между вершинами. Из множества скважин выделяются фациальные метки с наименьшими и наибольшими значениями скорости осадконакопления (в реализации используется фиксированная последовательность углубления, от меток с меньшими номерами к меткам с большими номерами). Далее в графе G_w выбираются пары вершин (u, v), для которых выполняются следующие условия:

- вершина v имеет метку наибольшей скорости осадконакопления;
- геометрическое расстояние $\mathrm{dist}(w(u), w(v))$ «мало отличается» от длины кратчайшего пути $len(sp(u, v))$;
- геометрическое расстояние $\mathrm{dist}(w(u), w(v))$ «сравнимо» с диаметром графа.

То есть направление углубление представляет собой достаточно длинную и прямую цепочку скважин.

Условия на расстояния определяются пороговым значением F_β меры ($\beta = 2$) между отклонением длины пути от расстояния и отношения длины пути к максимальному расстоянию между вершинами графа:

$$st = \frac{len(sp(u, v))}{\mathrm{dist}(u, v)} \tag{22}$$

$$d = \frac{\mathrm{dist}(u, v)}{\max_{u',v' \in V} dist(u', v')} \tag{23}$$

$$f_2 = \frac{(1 + \beta^2) st \cdot d}{\beta^2 st + d} \tag{24}$$

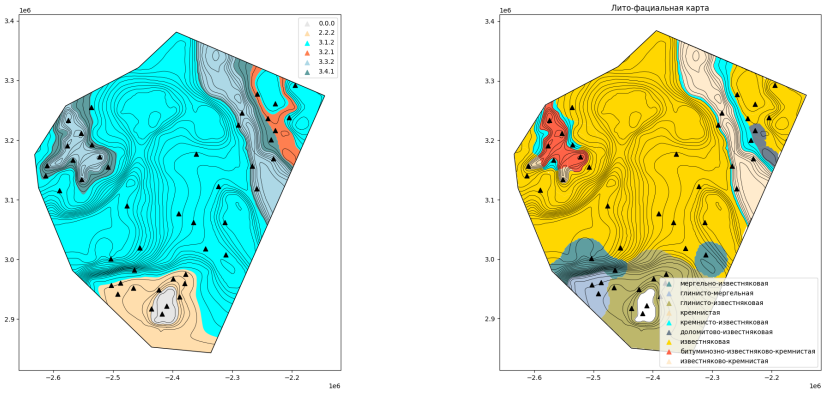
На карту добавляется $k = 5$ лучших (по значению f_2) путей, которые образуют потенциальные направления углубления бассейна.

4. Построение литофациальных карт $\mathcal{M}$

Карта $\mathcal{M}$ получается разбиением карты $\mathcal{M}_F$. Пример литофациальной карты приведен на рис. 2.

По скважинам с литологией методом радиальных базисных функций строится карта распространения литологий $\mathcal{M}_L : L \to 2^{Poly}$. Эта карта дает разбиение полигона распространения $\overline{P}$. Далее производится измельчение карты $\mathcal{M}_F$, то есть вычисляются все полигоны, для которых однозначно задана фация и литология:

$$\mathcal{P}_{lf} = \{P \in Poly \mid \exists f \in F, l \in L, P_f \in \mathcal{M}_F(f), P_l \in \mathcal{M}_L(l) \quad P = P_f \cap P_l\}, \tag{25}$$



(a) Карты распространения фаций. (b) Карта распространения литологий.

Рис. 2. Пример литофациальной карты (отражены области распространения литологии).

то есть все полигоны литологий пересекаются с всем полигонами фаций. По построению, в каждом полигоне этого множества могут встречаться скважины только с одной комбинацией фация-литология, поэтому P будем называть lf-полигоном.

Каждому литофациальному lf-полигону $P \in \mathcal{P}_{lf}$ требуется сопоставить литологическую метку — скорректировать границы распространения литологий. Используются следующие критерии:

- если в P есть скважина с меткой l, то эта метка приписывается полигону;
- если в P нет скважин, но он граничит с полигоном, которому приписана метка l, и литология l является допустимой для фации f, то метка l «продлевается» на P;
- в остальных случаях полигону приписывается значение литологии из ближайшей скважины фациального полигона P_f, содержащего P (в обозначениях (25)).

Назначение меток производится в три просмотра полигонов из $\mathcal{P}_{lf}$.

5. Техническая реализация

Для решения задач смешанного целочисленного программирования существует целый ряд программных систем, как коммерческих (например,

Gurobi и CPLEX), так и свободно распространяемых. Для реализации использовался язык Python и следующее программное обеспечение:

- contourpy: библиотека построения изолиний;
- shaply: библиотека для работы с полигонами и другими геометрическими объектами;
- pyomo: среда алгебраического моделирования, позволяющая формулировать задачи целочисленного программирования без привязки к конкретному пакету их решения;
- highs: пакет для решения задач смешанного целочисленного программирования.

Тестирование проводилось на данных нескольких реальных объектов, для которых известны литофациальные карты, подготовленные экспертами-геологами. Каждый объект содержал несколько десятков скважин, что, в свою очередь, приводит к графам с несколькими десятками вершин. При решении оптимизационной задачи использовалось ограничение по времени равное 60 секундам — по истечении этого времени пакет Highs возвращает лучшее решение, которое было найдено. В рассматриваемых примерах за отведённый интервал времени на персональном компьютере пользовательского уровня удавалось найти оптимальное решение задачи. Ограничение в 60 секунд считается допустимым с прикладной точки зрения.

6. Заключение

Задача построения литофациальных карт сводится к задаче раскраски графа, которая в данной работе решается методами целочисленного программирования. При построении карт по реальным данным этот подход показал высокую эффективность. В случае, когда особенность карты скоростей приводит к большому числу полигонов P_w и сложной топологической структуре графа, более эффективными могут оказаться методы раскраски на основе отложенной генерации столбцов [8].

Следует отметить, что предложенный подход имеет ряд ограничений, среди которых можно отдельно выделить полноту и качество исходных данных, масштаб картопостроения и реконструкции в складчато-надвиговых областях. Данное ограничения связано с заложенным в алгоритм принципом распределения литофаций вдоль направления углубления бассейна. В складчато-надвиговых зонах происходит деформационные процессы, которые изменяют первичное положение пород в пространстве и нарушают их взаимное расположение, заложенное в процессе осадконакопления.

Полученные результаты могут быть использованы для палеогеографических реконструкций, прогнозирования распространения различных элементов УВ-систем, в качестве входных параметром для бассейнного моделирования, а также для определения поисковых признаков.

Список литературы

[1] Кузнецов, В.Г., *Фации и фациальный анализ в нефтегазовой геологии*, РГУ нефти и газа им. И.М. Губкина, 2012.

[2] Наливкин Д.В., *Учение о фациях*, **1-2**, 1955–56.

[3] Алексеев В.П., *Литолого-фациальный анализ*, 2002.

[4] Тимофеев П.П., Рединг Х., *Обстановки осадконакопления и фации*, 1990.

[5] Kuhn, S., Cracknell, M. J., Reading, A. M., "Lithologic mapping using Random Forests applied to geophysical and remote-sensing data: A demonstration study from the Eastern Goldfields of Australia", *Geophysics*, **83**:4 (2018), B183–B193.

[6] Vasuki, Y., Holden, E. J., Kovesi, P., Micklethwaite, S., "An interactive image segmentation method for lithological boundary detection: A rapid mapping tool for geologists", *Computers & Geosciences*, 2017, № 100, 27–40.

[7] Peyghambari, S., Zhang, Y., "Hyperspectral remote sensing in lithological mapping, mineral exploration, and environmental geology: an updated review", *Journal of Applied Remote Sensing*, **15**:3 (2021), 031501.

[8] Gualandi S., Malucelli F., "Exact solution of graph coloring problems via constraint programming and column generation", *INFORMS Journal on Computing*, **24**:1 (2012), 81–100.

[9] Jabrayilov A., Mutzel P., "New integer linear programming models for the vertex coloring problem", LATIN 2018: Theoretical Informatics: 13th Latin American Symposium (Buenos Aires, Argentina, April 16-19, 2018, Proceedings 13), 2018, 640–652.

Automated construction of realistic lithofacies maps using combinatorial optimization methods

Antonov A.P., Afonin S.A., Kozitsyn A.S., Staroverov V.M., Stupakova A.V., Suslova A.A., Zavyalova A.P., Chupakhina V.V., Sautkin R.S., Osipov S.V.

Lithofacies maps display spatial changes in the lithological composition of rocks and the direction of their replacement characteristics for a certain geological time depending on the physical and geographical conditions of their sedimentation. Lithofacies maps are constructed based on the analysis of complexes of genetically related deposits with characteristic material features of lithofacies. The initial data can be either *only* points on a plane — well coordinates where certain lithofacies occur — or *additionally* a map of sediment accumulation rates. It is assumed that identical facies should have similar sedimentation rates. The task of constructing a lithofacies map is to determine the boundaries of facies distribution areas. The paper proposes to reduce the map construction problem to the graph coloring problem, which in turn is solved using integer linear programming methods. This approach allows us to automatically construct realistic maps – that is, maps that satisfy expert constraints and rules.

Keywords: paleo-reconstruction; lithofacial mapping; optimization; integer programming

References

[1] Kuznetsov V.G., *Facies and facies analysis in petroleum geology*, Gubkin Russian State University of Oil and Gas, 2012 (In Russian).

[2] Nalivkin D.V., *The doctrine of facies*, 1955–56 (In Russian).

[3] Alekseev V.P., *Lithofacies analysis*, 2002 (In Russian).

[4] Timofeev P.P., Reading H., *Depositional environments and facies*, 1990 (In Russian).

[5] Kuhn, S., Cracknell, M. J., Reading, A. M., "Lithologic mapping using Random Forests applied to geophysical and remote-sensing data: A demonstration study from the Eastern Goldfields of Australia", *Geophysics*, **83**:4 (2018), B183–B193.

[6] Vasuki, Y., Holden, E. J., Kovesi, P., Micklethwaite, S., "An interactive image segmentation method for lithological boundary detection: A rapid mapping tool for geologists", *Computers & Geosciences*, 2017, № 100, 27–40.

[7] Peyghambari, S., Zhang, Y., "Hyperspectral remote sensing in lithological mapping, mineral exploration, and environmental geology: an updated review", *Journal of Applied Remote Sensing*, **15**:3 (2021), 031501.

[8] Gualandi S., Malucelli F., "Exact solution of graph coloring problems via constraint programming and column generation", *INFORMS Journal on Computing*, **24**:1 (2012), 81–100.

[9] Jabrayilov A., Mutzel P., "New integer linear programming models for the vertex coloring problem", LATIN 2018: Theoretical Informatics: 13th Latin American Symposium (Buenos Aires, Argentina, April 16-19, 2018, Proceedings 13), 2018, 640–652.

Методы и алгоритмы автоматического извлечения информации из научных текстов для создания тезауруса научной терминологии

Е. В. Вопилова[1] Е. Н. Крючкова[2]

В статье предлагается метод автоматического построения тезауруса научной терминологии, основанный на алгоритмах извлечения многословных терминов из специальных энциклопедий и научных публикаций.

Представлены результаты работы алгоритмов создания и пополнения тезауруса на примере обработки математических текстов.

Предложен алгоритм сравнительного семантического анализа научных публикаций, а также способы количественной оценки их семантического сходства.

Ключевые слова: аспектно-ориентированный анализ, научный лексикон, семантический граф, классификация научных текстов, автоматическая обработка неструктурированных текстов.

1. Введение

Разработка инструментов для автоматической обработки текстов научной тематики является одним из актуальных направлений исследований в области NLP и включает задачи реферирования и аннотирования [1], классификации и аспектного анализа [2], информационного поиска [3]. При проектировании алгоритмов обработки текстов в компьютерной лингвистике используются вероятностно-статистические методы [4], нейросетевые модели [5], в последнее время широкое распространение получили большие языковые модели (LLM) [6].

Наличие адекватной семантической модели предметной области является необходимым условием для эффективного решения задачи тематического анализа. В работе [7] предложена формальная модель лингвистической онтологии, содержащая универсальные для различных предметных

[1] *Вопилова Елена Владимировна* — аспирант каф. прикладной математики ф-та инф. технологий АлтГТУ, e-mail: vopilova.elena@gmail.com.

Vopilova Elena Vladimirovna — graduate student, Polzunov Altai State Technical University, Faculty of Information Technology, Chair of Applied Mathematics.

[2] *Крючкова Елена Николаевна* — к.ф.м.н., профессор каф. прикладной математики ф-та инф. технологий АлтГТУ, e-mail: kruchkova_elena@mail.ru.

Kryuchkova Elena Nikolaevna — PhD in Physics and Mathematics, Professor, Polzunov Altai State Technical University, Faculty of Information Technology, Chair of Applied Mathematics.

областей типы семантических отношений. Широко распространены графовые модели, аккумулирующие семантические связи между словами и статистику частотности слов [10].

В данной работе рассматриваются методы и алгоритмы автоматического построения модели тезауруса предметной терминологии, позволяющей выполнять семантический анализ научного текста на основе объединения семантики отношений между терминами со статистикой использования терминов в научной публикации. Предложен алгоритм пополнения предлагаемой модели представления знаний информацией из научных публикаций. Построенная модель может быть использована для аспектного анализа узкоспециализированных текстов, в том числе количественной оценки сходства тематики публикаций в текстовой коллекции. Адекватность предлагаемой модели продемонстрирована на примере сравнения математических публикаций. Автоматически созданный тезаурус математических терминов, источником данных которого является математическая энциклопедия [11], размещен в открытом доступе по адресу evvopilova.pythonanywhere.com. Сервисные функции просмотра терминологии позволяют переходить по семантическим связям построенного тезауруса.

2. Многословные термины как основа научного лексикона

Тексты научных публикаций отличаются от остальных особой морфологией и лексикой, а также определёнными синтаксическими и семантическими структурами. Научная терминология обладает рядом особенностей, из которых наибольшее влияние на построение доменной семантической модели оказывает многословность научных терминов. В данной работе используется модель научного лексикона, построенная в результате автоматической обработки математической энциклопедии и представляющая собой ориентированный семантический граф G_{domain}, связывающий специфические доменные термины семантическими отношениями [12]. Анализ алгоритма построения этого графа выходит за рамки данной статьи, в данной работе мы будем использовать уже построенный семантический граф G_{domain}, вершины которого соответствуют терминам домена, а взвешенные дуги представляют семантические отношения различного типа между терминами. Вес дуги соответствует значимости семантического отношения между терминами. В процессе разработки модели тезауруса основное внимание будет уделено ассоциативным связям между терминами, которые отражают наиболее существенные взаимосвязи между сущностями [7].

3. Сравнительный семантический анализ публикаций на основе научного лексикона

Из активно развивающихся ветвей автоматического контент-анализа в современной математической лингвистике можно выделить два направления: категоризация текстов или отдельных фрагментов – определение принадлежности текста к некоторому классу внутри заданной предметной области [8] и аспектно-ориентированный анализ [9], предполагающий работу с текстом на уровне отдельных аспектов/функций целевого объекта. Основная цель аспектного анализа состоит в извлечении аспектов – сущностей, позволяющих объединить однотипные по функциональности элементы. В научной терминологии такими сущностями являются термины, объединяющие в единую группу связанные с выделенным аспектным термином другие термины. В данной работе эта задача решается методом кластеризации терминологии, связанной с текстом научной публикации, в результате задача извлечения аспектных терминов публикации рассматривается как задача выделения центров построенных кластеров.

Структурно научный текст может быть представлен как последовательность общеупотребительных слов и специальных терминов, причем семантическую нагрузку несут только научные термины. Сравнение тематики двух публикаций в простейшем варианте можно провести на уровне сравнения частотностей используемых в текстах терминов, но такой подход не учитывает контекстное окружение использованной терминологии и, следовательно, непригоден для оценки семантического сходства текстов. Построим семантический граф $G_{text}(T)$ текста T, выделяя из G_{domain} не только вершины, соответствующие используемым в T терминам, но и связанные с ними вершины из некоторой окрестности, сохраняя при этом для каждой связывающей дуги не только ее вес, но и тип связи. В процессе построения по взвешенным дугам графа на термины из окрестности распространим частотность явно используемого термина с учетом пространственного затухания.

Пусть Y – выделенный из текста анализируемой публикации термин, для которого в базовом графе G_{domain} имеется одноименная вершина, Y_i – вершина G_{domain} из некоторой окрестности $\omega(Y)$ термина Y, $f_{init}(Y_i)$ – частотность термина Y_i в анализируемом тексте, γ – коэффициент пространственного затухания. Тогда в результате работы алгоритма распространения весов вершина графа Y_i, связанная с вершиной Y с частотностью $f(Y)$, будет иметь итоговую характеристику частотности

$$f(Y_i) = f_{init}(Y_i) + f(Y) \cdot m(Y, Y_i) \cdot \gamma^{l(Y,Y_i)} \tag{1}$$

где $l(Y, Y_i)$ – число дуг на этом пути,

$m(Y, Y_i)$ – произведение весов дуг на ориентированном пути от Y к Y_i с наименьшим количеством дуг.

При каждом последующем обновлении значения $f(Y_i)$ по формуле (1) в качестве $f_{init}(Y_i)$ будет использоваться текущее значение $f(Y_i)$. В результате контекст домена, представленный графом G_{domain}, будет объединен со статистикой использования терминологии в статье. Кластеризация построенного графа $G_{text}(T)$ позволит выделить тематические аспекты научной публикации, а центры кластеров можно рассматривать как главные аспектные термины статьи. Для кластеризации был использован алгоритм *k-medoids*, который пригоден для кластеризации вершин графа так как предназначен для решения тех задач, в которых центром кластера может быть только точка из набора обозначенных. Если для каждого кластера C_i вычислить его вес $p(C_i)$ как сумму весов принадлежащих кластеру вершин, то относительный вес кластера можно рассматривать как относительную значимость соответствующего кластеру аспектного термина. Здесь и далее в качестве относительного веса кластера будем использовать отношение веса кластера к сумме весов всех кластеров.

Рассмотрим сравнительный семантический анализ публикаций. Пусть T_1 и T_2 – сравниваемые тексты, T_{12} – конкатенация T_1 и T_2. Построим графы $G_{text}(T_1)$, $G_{text}(T_2)$ и $G_{text}(T_{12})$, представляющие статистику контекстного окружения в соответствующих текстах. Построим множество кластеров $C = \{C_1, C_2, \ldots, C_k\}$ для $G_{text}(T_{12})$, в каждом кластере C_i выделим медоид – центральный термин M_i. Граф семантического сходства $G_{cmpr}(T_1, T_2)$ определяет поток из T_1 в T_2 и содержит вершины источника и стока T_1 и T_2, а также промежуточные вершины C_1, C_2, $\ldots$, C_n. Вершины C_1, C_2, $\ldots$, C_n имеют пропускную способность, равную относительному весу кластера в $_{Gtext}(_{T12})$, ребра между $_1$ (или $_2$) и C_i имеют пропускную способность, равную относительному суммарному весу вершин $G_{text}(T_1)$ (соответственно $G_{text}(T_2)$), присутствующих в кластере C_i графа $G_{text}(T_{12})$. Коэффициент семантического сходства текстов T_1 и T_2 равен потоку из истока T_1 в сток T_2 в транспортной сети $G_{cmpr}(T_1, T_2)$.

При проведении экспериментов текст публикации T_1 сравнивался с текстом T_1': изначально $T_1 = T_1'$, затем в качестве «белого шума» в T_1' постепенно добавлялся текст из математической научной публикации T_2 другой тематики. На рисунке 1 представлены результаты сравнения текстов T_1 и T_1'. При непосредственном сравнении текстов T_1 и T_2 их семантическое сходство текстов равно 0.26.

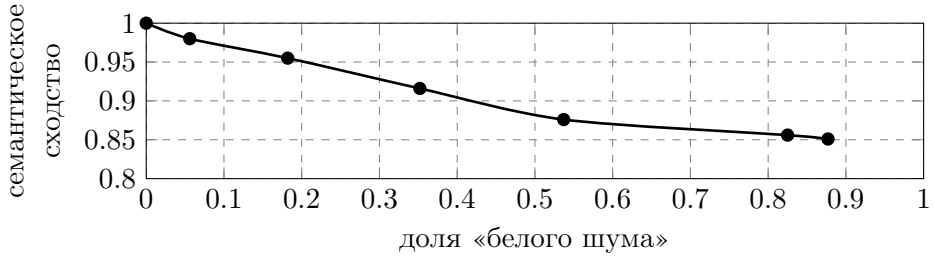


Рис. 1. Результаты сравнения текстов T_1 и T_1'

4. Алгоритмы обновления терминологии

В процессе эволюционного развития науки появляются новые разделы и научные направления, что приводит к появлению не только новой терминологии, но и к частичному изменению семантики существующей устоявшейся терминологии: возникают новые семантические связи между терминами, трансформируются некоторые существующие связи. Это приводит к необходимости пополнения базы знаний научной области, а, следовательно, и к выбору надежных источников обучающих данных. Источником надежной информации для систем автоматического обновления баз знаний могут служить публикации из рецензируемых изданий в ранжируемых научных издательствах.

Рассмотрим сначала задачу выявления новых предметных терминов. Несмотря на то, что с точки зрения семантической разметки текст научной статьи является неструктурированным, наличие в тексте статьи раздела «Ключевые слова» можно рассматривать как наличие некоторой слабой структурированности этого текста в целом. Определим специфику пополнения графа научной области G_{domain} при семантической обработке достаточно обширного текста научной публикации, некоторая часть которого связана с описанием нового понятия, в качестве которых мы будем рассматривать перечисленные в статье ключевые слова.

Пусть Y – абсолютно новый термин, тогда в семантический граф G_{domain} добавляется термин Y, а также термины, составленные из подмножества слов термина Y, связанные с Y семантическими связями типа «часть – целое» при условии, что новый термин имеет корректную синтаксическую структуру. Формирование связи типа «часть - целое» соответствует принципу формирования многословных терминов, составленных из уточняющих слов. Как правило, любое удаление хотя бы одного слова из термина приводит к определению более общего понятия. Например, «Маркова цепь сложная» является частью всех «цепей Маркова» или всех «сложных цепей». Для создания ассоциативных связей термина Y, во-первых, надо выделить в тексте статьи фрагменты, име-

ющих непосредственное отношение к новому термину, во-вторых, надо вычислить значимость связей нового термина Y с другими терминами, которые определяют главную тематику статьи. В силу того, что ключевое слово достаточно редко целиком используется в тексте статьи, попытка выделить физический фрагмент текста, имеющего непосредственное отношение к Y, в большинстве случаев заранее обречена на неудачу. Но остается возможность связать Y с тематикой статьи в целом. Семантика статьи T формируется в процессе построения графа $G_{text}(T)$. Таким образом, возникает задача выделения в построенных кластерах для графа $G_{text}(T)$ информации, контекстно связанной с новым термином. Вес ассоциативных связей между ключевым словом Y и термином M_i, который является центром кластера C_i, определяется как

$$Z(Y, M_i) = \frac{p(C_i)}{\sum_{i=1}^{n} p(C_i)} \tag{2}$$

где $p(C_i)$ – вес кластера C_i.

Исходный граф G_{domain} построен по исключительно авторитетному источнику информации – научной энциклопедии. Энциклопедия – совместный труд заслуженных авторов, аккумулирует устоявшиеся данные по большинству разделов науки. Поэтому вычисленный по формуле (2) на основании данных, извлеченных из единственного источника, следует умножить на некоторый коэффициент уровня доверия к источнику ξ ($\xi \leq 1$):

$$\tilde{Z}(Y, M_i) = \xi \cdot Z(Y, M_i) \tag{3}$$

В таблице 1 представлены результаты выделения новых терминов при обработке коллекции публикаций научных журналов [13, 14].

Таблица 1. Результаты пополнения базового семантического графа информацией из текстового корпуса

Число существующих вершин-кандидатов, переведенных в статус базового термина	36
Число новых вершин	107
Число новых связей – из них связей «часть-целое» – из них связей-ассоциаций	4345 2818 1527

5. Повышение весовых коэффициентов ассоциативных связей

Каждая новая статья может содержать информацию из новых разделов науки или новых научных направлений, данные о которых отсутствуют или плохо представлены в графеG_{domain}. Следовательно, обработка новой статьи может дополнить наши знания как новыми терминами, так и новыми семантическими связями между терминами – как новыми, так и уже имеющимися в тезаурусе. Семантика статьи может указывать на более сильные зависимости между терминами. Фактически это означает, что обработка статьи из доверенных источников может производиться в двух режимах:

1) в режиме пополнения новыми терминами и новыми связями,

2) в режиме модификации весов существующих семантических связей с поддержкой режима (1).

Выделенные из текста публикации ключевые слова могут быть как новыми терминами, так и уже существующими в графе G_{domain}. Новые термины добавляются вместе с новыми связями, веса которых рассчитаны по формуле (3). Если же семантическая связь уже существует, то возникает вопрос об изменении ее веса. Уже существующие связи построены на основе доверенных источников информации, и одна статья не может претендовать на кардинальное изменение весов, но может привести к некоторой их модификации.

Пусть $\tilde{Z}(Y_1, Y_2)$ – вес дуги в графе G_{domain}, $\tilde{Z}_1(Y_1, Y_2)$ – вычисленный по формуле (3) вес этой дуги. Если $\tilde{Z}(Y_1, Y_2) \geq \tilde{Z}_1(Y_1, Y_2)$, то новый текст не увеличивает ассоциативную связь между терминами, вес существующей связи не изменяется. В случае $\tilde{Z}(Y_1, Y_2) < \tilde{Z}_1(Y_1, Y_2)$ новый текст должен усилить эту ассоциативную связь и необходим перерасчет веса соответствующей дуги.

Вычислим относительное среднеквадратичное отклонение

$$d = \frac{|\tilde{Z}(Y_1, Y_2) - \tilde{Z}_1(Y_1, Y_2)|}{\tilde{Z}(Y_1, Y_2) + \tilde{Z}_1(Y_1, Y_2)}$$

Произведем корректировку веса дуги:

$$\tilde{Z}_{new}(Y_1, Y_2) = g(\tilde{Z}(Y_1, Y_2), \tilde{Z}_1(Y_1, Y_2))$$

где

$$g(\tilde{Z}(Y_1, Y_2), \tilde{Z}_1(Y_1, Y_2)) = \tilde{Z}(Y_1, Y_2) + \tilde{Z}_1(Y_1, Y_2) \cdot f(d) \qquad (4)$$

Выбор функции $f(d)$ и, следовательно, $g(\tilde{Z}(Y_1, Y_2), \tilde{Z}_1(Y_1, Y_2))$ определяется следующими критериями:

- при однократном появлении $\tilde{Z}_1(Y_1, Y_2)$, значительно превышающем $\tilde{Z}(Y_1, Y_2)$, не должен наблюдаться резкий рост $\tilde{Z}_{new}(Y_1, Y_2)$;
- при постоянном появлении большого значения $\tilde{Z}_1(Y_1, Y_2)$ вычисленное по формуле (4) значение $\tilde{Z}_{new}(Y_1, Y_2)$ должно за некоторое число итераций сходиться к $\tilde{Z}_1(Y_1, Y_2)$.

В данной работе используется

$$f(d) = \frac{d}{\sqrt{2}} = \frac{|\tilde{Z}(Y_1, Y_2) - \tilde{Z}_1(Y_1, Y_2)|}{\sqrt{2} \cdot (\tilde{Z}(Y_1, Y_2) + \tilde{Z}_1(Y_1, Y_2))}$$

Рассмотрим сходимость $\tilde{Z}_{new}(Y_1, Y_2)$ $\tilde{Z}_1(Y_1, Y_2)$. Вычислим количество итераций n, при котором $|\tilde{Z}(Y_1, Y_2) - \tilde{Z}_1(Y_1, Y_2)| < \varepsilon$. При $\varepsilon = 0.001$ потребуется от 10 до 14 шагов, а при $\varepsilon = 0.0001$ количество итераций лежит в диапазоне от 14 до 19. Фактически это означает, что даже при небольшом начальном весе дуги наличие 14-19 текстов с высоким уровнем ассоциативных отношений приводит к вычислению веса дуги, соответствующему данным из новых источников. С другой стороны, исходный вес дуги постепенно увеличивается только при наличии достаточного количества доверенных источников, что говорит о сбалансированном требовании к набору статей для устойчивого формирования нового веса $\tilde{Z}_{new}(Y_1, Y_2)$.

На рисунке 2 представлены результаты перерасчета веса существующей дуги при многократной обработке одной и той же научной публикации.

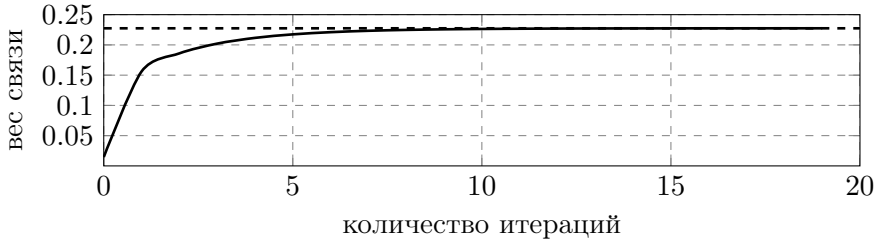


Рис. 2. Изменение веса дуги «дифференциальная игра» → «стратегия» графа G_{domain} при обработке текста научной публикации

6. Заключение

В статье рассмотрены проблемы автоматического формирования тезауруса научной области. Автоматически сформирован тезаурус математической терминологии, предложены алгоритмы пополнения построенного тезауруса. Рассматриваются вопросы выделения тематических аспектов

научной публикации, предложенные алгоритмы базируются на алгоритмах кластеризации графов.

Список литературы

[1] Altmami N., Menai M., "Automatic Summarization of Scientific Articles: A Survey", *Journal of King Saud University - Computer and Information Sciences*, **34** (2020), 1011–1026.

[2] Berna A, B., Ganiz M., "Semantic text classification: A survey of past and recent advances", *Information Processing & Management*, **54**:6 (2018), 1129–1163.

[3] Benites F., "Information Retrieval and Knowledge Extraction for Academic Writing", *Digital Writing Technologies in Higher Education*, 2023, 303–315.

[4] Hossari M., Dev S., Kelleher J. D., "TEST: A Terminology Extraction System for Technology Related Terms", *Proc. The 2019 11th International Conference on Computer and Automation Engineering*, 2019, 78-81.

[5] Danilov G., Ishankulov T., Kotik K., Orlov Yu. Shifrin M., Potapov A., "The Classification of Short Scientific Texts Using Pretrained BERT Model", *Public Health and Informatics*, **281** (2021), 83–87.

[6] Dunn A., Dagdelen J., Walker N., Lee S., Rosen A., Ceder G., Persson K., Jain A., "Structured information extraction from complex scientific text with fine-tuned large language models", 2022, 83–87.

[7] Лукашевич Н.В., Добров Б.В., "Проектирование лингвистических онтологий для информационных систем в широких предметных областях", *Онтология проектирования*, **5**:1(15) (2015), 47–69.

[8] Costa L.S., Oliveira I.A., Fileto R., "Text classification using embeddings: a survey", *Knowledge and Information Systems*, **65** (2023), 2761-2803.

[9] Marshalova A.,Bruches E., Batura T., "Automatic Aspect Extraction from Scientific Texts", *Proc. Recent Trends in Analysis of Images, Social Networks and Texts (AIST 2023), Communications in Computer and Information Science*, **1905** (2023), 67-80.

[10] Belwal R., Rai S., Gupta A., "A new graph-based extractive text summarization using keywords or topic modeling", *Journal of Ambient Intelligence and Humanized Computing*, **12** (2021), 8975–8990.

[11] *Математическая энциклопедия в 5 томах*, ред. Виноградов И.М., Советская энциклопедия, Москва, 1977.

[12] Вопилова Е.В., Крючкова Е.Н., "Методы автоматического анализа динамики изложения информации в текстах на основе адаптируемых словарей научных терминов", *Программная инженерия*, **15**:4 (2024), 206–215.

[13] *Вестник Южно-Уральского университета, серия «Математика. Механика. Физика»*, **14**:2–4 (2022).

[14] *Математический сборник*, **213**:9–12 (2022).

Methods and algorithms for automatic extraction of information from scientific texts for creating a scientific terminology thesaurus

Vopilova E.V., Kryuchkova E.N.

The paper proposes a method for automatic construction of a scientific terminology thesaurus based on algorithms for extraction of multi-word terms from special encyclopedias and scientific publications.

The results of the algorithms for thesaurus creation and replenishment are presented on the example of mathematical text processing.

We propose the algorithm for comparative semantic analysis of scientific publications and the ways of quantitative estimation of their semantic similarity.

Keywords: aspect-oriented analysis, scientific vocabulary, semantic graph, classification of scientific text, automatic processing of unstructured texts.

References

[1] Altmami N., Menai M., "Automatic Summarization of Scientific Articles: A Survey", *Journal of King Saud University - Computer and Information Sciences*, **34** (2020), 1011–1026.

[2] Berna A, B., Ganiz M., "Semantic text classification: A survey of past and recent advances", *Information Processing & Management*, **54**:6 (2018), 1129–1163.

[3] Benites F., "Information Retrieval and Knowledge Extraction for Academic Writing", *Digital Writing Technologies in Higher Education*, 2023, 303–315.

[4] Hossari M., Dev S., Kelleher J. D., "TEST: A Terminology Extraction System for Technology Related Terms", *Proc. The 2019 11th International Conference on Computer and Automation Engineering*, 2019, 78-81.

[5] Danilov G., Ishankulov T., Kotik K., Orlov Yu. Shifrin M., Potapov A., "The Classification of Short Scientific Texts Using Pretrained BERT Model", *Public Health and Informatics*, **281** (2021), 83–87.

[6] Dunn A., Dagdelen J., Walker N., Lee S., Rosen A., Ceder G., Persson K., Jain A., "Structured information extraction from complex scientific text with fine-tuned large language models", 2022, 83–87.

[7] Lukashevich N.V., Dobrov B.V., "Designing linguistic ontologies for information systems in broad subject areas", *Ontology of Designing*, **5**:1(15) (2015), 47–69 (In Russian).

[8] Costa L.S., Oliveira I.A., Fileto R., "Text classification using embeddings: a survey", *Knowledge and Information Systems*, **65** (2023), 2761-2803.

[9] Marshalova A.,Bruches E., Batura T., "Automatic Aspect Extraction from Scientific Texts", *Proc. Recent Trends in Analysis of Images, Social Networks and Texts (AIST 2023), Communications in Computer and Information Science*, **1905** (2023), 67-80.

[10] Belwal R., Rai S., Gupta A., "A new graph-based extractive text summarization using keywords or topic modeling", *Journal of Ambient Intelligence and Humanized Computing*, **12** (2021), 8975–8990.

[11] *Mathematical encyclopedia in 5 volumes*, ред. Vinogradov I.M., Soviet Encyclopedia, Moscow, 1977 (In Russian).

[12] Vopilova E. V., Kryuchkova E. N., "Automatic analysis methods of dynamics of information presentation in texts based on adaptable dictionaries of scientific terms", *Software Engineering*, **15**:4 (2024), 206–215 (In Russian).

[13] *Bulletin of the South Ural State University, Ser. Mathematics. Mechanics. Physics*, **14**:2–4 (2022) (In Russian).

[14] *Sbornik: Mathematics*, **213**:9–12 (2022) (In Russian).

Часть 2
Специальные вопросы теории интеллектуальных систем

О предельных циклах в однородных нейронных сетях

А. С. Дробышев[1]

В работе рассматривается формальная модель нейронных сетей, в которой каждой нейрон представлен в виде автомата, состоящего из пороговой булевой функции и задержки. Доказывается критерий принадлежности стартовой конфигурации предельному циклу.
Ключевые слова: нейронные сети, пороговые функции, схемы из функциональных элементов, предельные циклы.

1. Введение

Сегодня исследованию нейросетей уделяется большое внимание. Первая формальная математическая модель нейрона была представлена в 1943 году У. С. Мак-Каллоком и В. Питтсом [5], позднее в 1956 году С. К. Клини [6] показал, что каждый конечный автомат моделируется нейронной сетью с задержкой в два такта. В данной статье используется модель нейрона, впервые представленная в работе [1] и получившая дальнейшее развитие в работах [2, 3], где нейрон представляется в качестве автомата, состоящего из пороговой функции и задержки. Нейронная сеть представляет собой граф связей таких нейронов. Основным результатом работы является критерий, дающий условия, при которых стартовая конфигурация нейронной сети принадлежит ее предельному циклу.

2. Основные понятия и результаты

Отображение $f : \{0,1\}^m \to \{0,1\}$ назовём *пороговой функцией*, если существует *весовой* вектор $w = \langle w^1, \ldots, w^m \rangle \in \mathbb{Z}$ и такое $h \in \mathbb{Z}$, называемое порогом, что для всех $x = (x^1, \ldots, x^m) \in \{0,1\}^m$ имеет место равенство: $f(x) = sign(w^1 \cdot x^1 + \cdots + w^m \cdot x^m - h)$, где

$$sign(x) = \begin{cases} 1, & x > 0; \\ 0, & x \leqslant 0; \end{cases}$$

Отношение $R \subseteq \{0,1\}^m$ будем называть *пороговым*, если R — область истинности некоторой пороговой функции. Обозначим $\mathcal{N}(m) := \{\exists_c \chi_R \mid$

[1] *Дробышев Александр Сергеевич* — аспирант каф. математической теории интеллектуальных систем мех.-мат. ф-та МГУ, e-mail: drobyshev.sanya@yandex.ru.

Drobyshev Alexander Sergeevich — graduate student, Lomonosov Moscow State University, Faculty of Mechanics and Mathematics, Chair of Mathematical Theory of Intellectual Systems.

$R \subseteq \{0,1\}^m\}$, где $\chi_R(x)$ — характеристическая функция, определённая как $\chi_R(x) = 1 \Leftrightarrow x \in R$. Элементы множества $\mathcal{N} = \bigcup\limits_{m=0}^{+\infty} \mathcal{N}(m)$ будем называть *нейронами*.

Пусть $f_1, \ldots, f_n$ — пороговые функции от $n+m$ переменных. *Нейронной сетью* назовём всякую схему из функциональных элементов с задержкой $\Sigma(f_1, \ldots, f_n)$, заданную системой уравнений вида

$$\begin{cases} y_1 = \mathfrak{Z}_{c_1} f_1(y_1, \ldots, y_n), \\ \ldots \\ y_n = \mathfrak{Z}_{c_n} f_n(y_1, \ldots, y_n), \end{cases}$$

где $c_i \in \{0,1\}$ для всех $i = 1, \ldots, n$. Иногда будем писать Σ вместо $\Sigma(f_1, \ldots, f_n)$, когда функции $f_1, \ldots, f_n$ фиксированы.

В данной работе рассматриваются нейронные сети, в которых каждому нейрону приписана одна и та же пороговая функция f. Такие нейронные сети будем называть *однородными* и обозначать $\Sigma(f)$.

Каждая нейронная сеть $\Sigma(f_1, \ldots, f_n)$ определяет отображение $\Phi_\Sigma : \{0,1\}^n \to \{0,1\}^n$, заданное условием: $\Phi(y_1, \ldots, y_n) = (\mathfrak{Z}_{c_1} f_1(y_1, \ldots, y_n), \ldots, \mathfrak{Z}_{c_n} f_n(y_1, \ldots, y_n))$. Набор $\alpha(t) = (\alpha_1(t), \ldots, \alpha_n(t)) \in \{0,1\}^n$ будем называть *конфигурацией* сети в момент времени $t \in \mathbb{N} \cup \{0\}$. Отметим, что конфигурации нейронной сети Σ в моменты времени t и $t+1$ связаны соотношениями $\alpha(t+1) = \Phi_\Sigma(\alpha(t))$.

Предельным циклом нейронной сети Σ будем называть последовательность конфигураций $\alpha_1, \ldots, \alpha_k \in \{0,1\}^n$ таких, что $\alpha_{i+1} = \Phi(\alpha_i)$ для любого $i = 1, \ldots, k-1$ и $\alpha_1 = \Phi(\alpha_k)$. Будем называть нейросетевое отображение Φ *стабилизированным*, если существует момент времени $t \in \mathbb{N}$, что $(y_1(t), \ldots, y_n(t)) = (y_1(0), \ldots, y_n(0))$.

Каждой нейронной сети Σ сопоставим граф $G_\Sigma = (V, E)$, вершины которого представляют собой нейроны $v_1, \ldots, v_n$ с приписанными им значениями $\alpha_{v_i} = \mathfrak{Z}_{c_i} f_i(\alpha_1, \ldots, \alpha_n)$, где f_i — функция, соответствующая нейрону v_i. Будем говорить, что нейрон v_i равен c в момент времени t, если $\alpha_{v_i}(t) = c,\ c \in \{0,1\}$.

В графе G могут быть рёбра, ведущие из нейрона v_i в v_j, при этом функция f_j, приписанная v_j несущественно зависит от y_i. В таком случае это ребро ни на что не влияет и его можно стереть. Не ограничивая общности, будем считать, что из вершины v_i ведет ребро в вершину v_j $\Leftrightarrow f_j$ существенно зависит от y_i.

В этой работе рассматриваются однородные нейронные сети для пороговых функций конъюнкции ($f = x \wedge y$) и дизъюнкции ($f = x \vee y$).

Теорема 1. *Пусть $G_\Sigma = (V, E)$ — связный граф однородной нейронной сети $\Sigma(f)$, где $f = x \wedge y$. Тогда концигурация $\alpha(0)$ принадлежит предель-*

ному циклу сети $\Sigma(f) \Leftrightarrow$ *существуют такие подмножества вершин* $U_1, \ldots, U_l \subseteq V$, *что* $U_1 \cup \cdots \cup U_l = V$, *и выполнены условия:*

1) *Для любых* $i = 1, \ldots, l$ *и* $(u, v) \in E$ *выполнено*

   а) *если* $v \in U_i$, *то* $u \in U_{i-1}$;

   б) *если* $u \in U_i$, *то* $v \in U_{i+1}$;

   причем $U_{l+1} = U_1$;

2) *Если существуют вершины* $u, v \in U_i$ *такие, что* $\alpha_u(0) = 0, \alpha_v(0) = 1$, *то не существует пути в графе* G *из вершины* u *в вершину* v. *Для любой вершины* $v \in U_i$ *такой, что* $\alpha_v(0) = 0$, *существует непустой путь в графе* G *из* u *в вершину* $v \in U_i$ *такую, что* $\alpha_u(0) = 0$.

Из принципа двойственности и Теоремы 1 верно следующее.

Теорема 2. *Пусть* $G_\Sigma = (V, E)$ — *связный граф однородной нейронной сети* $\Sigma(f)$, *где* $f = x \vee y$. *Тогда концигурация* $\alpha(0)$ *принадлежит предельному циклу сети* $\Sigma(f) \Leftrightarrow$ *существуют такие подмножества вершин* $U_1, \ldots, U_l \subseteq V$, *что* $U_1 \cup \cdots \cup U_l = V$, *и выполнены условия:*

1) *Для любых* $i = 1, \ldots, l$ *и* $(u, v) \in E$ *выполнено*

   а) *если* $v \in U_i$, *то* $u \in U_{i-1}$;

   б) *если* $u \in U_i$, *то* $v \in U_{i+1}$;

   причем $U_{l+1} = U_1$;

2) *Если существуют вершины* $u, v \in U_i$ *такие, что* $\alpha_u(0) = 1, \alpha_v(0) = 0$, *то не существует пути в графе* G *из вершины* u *в вершину* v. *Для любой вершины* $v \in U_i$ *такой, что* $\alpha_v(0) = 1$, *существует непустой путь в графе* G *из* u *в вершину* $v \in U_i$ *такую, что* $\alpha_u(0) = 1$.

3. Доказательство Теоремы 1

Сначала докажем несколько вспомогательных лемм. Пусть $G_\Sigma = (V, E)$ — связный граф однородной нейронной сети $\Sigma(f)$, где $f = x \wedge y$. Всюду далее, если S — сильно связная компонента графа G_Σ, то через E_S, V_S будем обозначать ограничения на эту компоненту множеств рёбер и вершин соответственно.

Определим отношение ρ_S, полагая

$$\rho_0 = \{(v_1, v_2) \in V_S \times V_S \mid \exists v_3 (v_1, v_3) \in V_S : (v_2, v_3) \in E_S\} \cup \{(v, v) \in V_S\}$$

и

$$\rho_i = \{(v_1, v_2) \in V_S \times V_S \mid \exists v_3, v_4 \in V_S : (v_1, v_3), (v_2, v_4) \in E, (v_3, v_4) \in \rho_{i-1}\}$$

для всех $i \in \mathbb{N}$. Пусть $\rho_S = \bigcup\limits_{i=0}^{\infty} \rho_i$

Лемма 1. *Пусть S — сильно связная компонента графа G_Σ. Если $(a_1, b_1) \in \rho_S$, то для любых $a_2, b_2 \in V_S$, таких, что $(a_1, a_2) \in E_S, (b_1, b_2) \in E_S$, выполнено $(a_2, b_2) \in \rho_S$*

Доказательство. Пусть $(a_2, b_2) \notin \rho_S$, тогда для любых $a_3, b_3 \in V_S$ таких, что $(a_2, a_3) \in E$, $(b_2, b_3) \in E$, выполнено $(a_3, b_3) \notin \rho_S$. Так как S — сильно связная компонента, то через любые две ее вершины проходит по крайней мере один цикл. Рассмотрим два цикла:

1) Цикл S_1 минимальной длины m, проходящий через вершины a_1, a_2;

2) Цикл S_2 минимальной длины n, проходящий через вершины b_1, b_2.

Без ограничения общности считаем, что $m \leqslant n$. Тогда для этих циклов будет выполнено следующее:

$$(a_2, b_2) \notin \rho_S,\ (a_3, b_3) \notin \rho_S,\ \ldots,\ (a_m, b_m) \notin \rho_S.$$

С другой стороны, $(a_1, b_1) \in \rho_S$, следовательно, $(a_n, b_m) \in \rho_S$. А значит $(a_{n-1}, b_{m-1}) \in \rho_S, (a_{n-2}, b_{m-2}) \in \rho_S$ и т.д. Тогда, перебирая вершины этих циклов в обратном порядке не позднее, чем через nm проходов по циклам, получим, что как минимум одна из пар $(a_i, b_i), i \geqslant 2$, принадлежит ρ_S. Противоречие с изначальным предположением. □

Лемма 2. *Если S — сильно связная компонента графа G_Σ, то ρ_S — отношение эквивалентности на множесте вершин V_S.*

Доказательство.

- Рефлексивность следует из того, что включение $(v, v) \in \rho_0$ верно для всех вершин $v \in V_S$.
- Симметричность следует из построения.
- Транзитивность: Пусть $(v_1, v_2) \in \rho_S, (v_2, v_3) \in \rho_S$. Из построения следует, что существуют вершины $u_1, u_2 \in V_S$, достижимые из вершин v_1, v_2 и v_3, v_4. Пусть s_1 — длина кратчайшего пути из v_1, v_2 в u_1, а s_2 — длина кратчайшего пути из v_2, v_3 в u_2. Без ограничения общности будем считать, что $s_1 \geqslant s_2$. Обозначим эти пути следующим образом:

$$\alpha = \alpha_1\alpha_2 \ldots \alpha_{s_1} - \text{путь из } v_1 \text{ в } u_1,$$

$$\beta = \beta_1\beta_2\ldots\beta_{s_1} - \text{путь из } v_2 \text{ в } u_1,$$

$$\gamma = \gamma_1\gamma_2\ldots\gamma_{s_2} - \text{путь из } v_2 \text{ в } u_2,$$

$$\delta = \delta_1\delta_2\ldots\delta_{s_2} - \text{путь из } v_3 \text{ в } u_2,$$

где $\alpha_1 = v_1, \beta_1 = v_2, \alpha_{s_1} = \beta_{s_1} = u_1, \gamma_1 = v_2, \delta_1 = v_3, \gamma_{s_2} = \delta_{s_2} = u_2$. Тогда имеют место включения

$$(\alpha_1, \beta_1), (\beta_1, \gamma_1), (\gamma_1, \delta_1) \in \rho_S,$$

следовательно, по Лемме 1

$$(\alpha_2, \beta_2), (\beta_2, \gamma_2), (\gamma_2, \delta_2) \in \rho_S, \ldots, (\alpha_{s_2}, \beta_{s_2}), (\beta_{s_2}, \gamma_{s_2}), (\gamma_{s_2}, \delta_{s_2}) \in \rho_S.$$

Так как из $(\alpha_{s_2}, \delta_{s_2}) \in \rho_S$ следует, что $(\alpha_{s_2}, \delta_{s_2}) \in \rho_S$, то, пройдя обратно от $\alpha_{s_2}, \delta_{s_2}$ по путям α, δ, получим, что $(v_1, v_3) \in \rho_S$. Значит, $(v_1, v_2) \in \rho \Rightarrow (u_1, u_2) \in \rho_{V_1}$ — следует из построения ρ, и $(u_1, u_2) \in \rho \Rightarrow (v_1, v_2) \in \rho_{V_1}$ — следует из доказанного ранее.

□

Лемма 3. *Если S — сильно связная компонента графа G_Σ, то на вершинах этой сильно связной компоненты можно построить такое разбиение на классы эквивалентности $U_1, \ldots, U_m$, что для любого ребра $(v_1, v_2) \in E_S$ выполнено*

1) *Если $v_1 \in U_i$, то $v_2 \in U_{i+1}$;*

2) *Если $v_2 \in U_i$, то $v_1 \in U_{i-1}$.*

Доказательство. Пусть $U_1, \ldots, U_m$ — классы эквивалентности, определенные отношением ρ_S. Из Леммы 1 следует, что для любого ребра (v_1, v_2) такого, что $v_1 \in U_1$, будет выполнено $v_2 \in U_i$ для некоторого i. Без ограничения общности будем считать что это U_2. Аналогично, для любого ребра (v_1, v_2) такого, что $v_1 \in U_2$, будет выполнено $v_2 \in U_3$ и т.д. Пусть существует такое k, что для любого ребра (v_1, v_2) такого, что $v_1 \in U_k$, будет выполнено $v_2 \in U_1$ и $k \neq m$. Рассмотрим произвольные $v_1 \in U_1$ и $v_2 \in U_{k+1}$. Так как из U_k все ребра ведут в U_1, то не существует пути $\pi = \pi(v_1, v_2)$, связывающего эти две вершины — противоречие с тем, что S — сильно связная компонента. □

Лемма 4. *Пусть S — сильно связная компонента графа G_Σ, ρ — отношение эквивалентности из Леммы 2 и отображение Φ стабилизировано. Тогда для любой пары $(v_1, v_2) \in \rho$ равенство $\alpha_{v_1}(t) = \alpha_{v_2}(t)$ выполнено для любого $t \geqslant 0$.*

Доказательство. Докажем, что утверждение леммы верно для каждого ρ_i, $i \geqslant 0$. Для этого покажем сначала, что для любых v_1, v_2 таких, что $(v_1, v_2) \in \rho_0$, выполнено $\alpha_{v_1}(t) = \alpha_{v_2}(t)$ для любого $t \geqslant 0$. Рассмотрим произвольную тройку вершин $v_1, v_2, v_3 \in V_S$ таких, что $(v_1, v_3), (v_2, v_3) \in E_S$. Пусть G' — произвольный подграф графа G, тогда обозначим $0_{G'}(t) = \{v \in V_{G'} \mid v(t) = 0\}$. В силу стабилизированности Φ $0_C(t) = const$ для любого цикла C, проходящего через вершины G_Σ. Предположим, что существует t такое, что $v_1(t) = 0, v_2(t) = 1$. В таком случае, $v_3(t + 1) = 0$ и количество нулей в цикле, проходящем через v_2, v_3 увеличится. Противоречие.

Пусть для ρ_i и для любых $(v_1, v_2) \in \rho_i$ выполнено $\alpha_{v_1}(t) = \alpha_{v_2}(t)$ для любого t. Покажем, что тогда то же будет выполнено для ρ_{i+1}. Докажем, что верно для v_1, v_2, v_3, v_4 таких, что $(v_1, v_3), (v_2, v_4) \in E, (v_3, v_4) \in \rho_i$ и v_3, v_4 — различные. По предположению значения в них совпадают для любого t. Но тогда если $\alpha_{v_1}(t) \neq \alpha_{v_2}(t)$ для какого-то t, то в момент времени $t + 1$ значения в v_3 и v_4 будут отличаться. Противоречие. Таким образом, для любых $v_1, v_2 \in \rho_S$ выполнено $\alpha_{v_1}(t) = \alpha_{v_2}(t)$ для любого $t \geqslant 0$. □

Лемма 5. *Если граф G_Σ состоит из одной сильно связанной компоненты, то для него выполнены необходимые условия Теоремы 1.*

Доказательство. Пусть S — эта сильно связная компонента, а $U_1, \ldots, U_m$ — классы эквивалентности, определенные на этой сильно связной компоненте отношением эквивалентности ρ_s. Рассмотрим произвольное ребро $(u, v) \in E$, без ограничения общности будем считать, что $u \in U_i$ для некоторого i. Тогда из Леммы 3 следует, что $v \in U_{i+1}$, где $i + 1$ берется по модулю m. Из Леммы 4 для любых вершин $u_1, u_2 \in U_i$ выполнено $\alpha_{u_1}(t) = \alpha_{u_2}(t)$, а так как S — сильно связная компонента, то между любыми ее двумя вершинами существует путь. Таким образом, для этой сильно связной компоненты выполнены необходимые условия Теоремы 1. □

Будем говорить, что последовательность $U_1, \ldots, U_m$ множеств вершин из V — *охватывающий цикл* (обозначим через C), если для любых двух вершин $u, v \in V$ таких, что $(u, v) \in E$, выполнено

1) Если $u \in U_i$, то $v \in U_{i+1}$,

2) Если $v \in U_i$, то $u \in U_{i-1}$,

причем все индексы берутся по модулю m. Если $C = \langle U_1, \ldots, U_n \rangle$ — охватывающий цикл, то через $V_C = U_1 \cup \ldots \cup U_m$ будем обозначать множество вершин этого охватывающего цикла.

Лемма 6. *Граф G_Σ можно разбить на последовательность охватывающих циклов $C_1, \ldots, C_k$, удовлетворяющую следующим условиям:*

1) $V_{C_i} \cap V_{C_j} = \emptyset$ для всех $i \neq j$;

2) $V_{C_1} \cup \cdots \cup V_{C_k} = V$;

3) Для любого $C_j = \langle U_1, \ldots, U_m \rangle$ выполнено

- *Если существуют вершины $u, v \in U_i$ такие, что $\alpha_u(0) = 1$ и $\alpha_v(0) = 0$, то не существует пути в графе G из вершины u в вершину v.*
- *Для любой вершины $v \in U_i$ такой, что $\alpha_v(0) = 0$, существует путь в графе G из u в вершину $v \in U_i$ такую, что $\alpha_u(0) = 0$.*

Доказательство. Из Леммы 5 следует, что каждую сильно связную компоненту G_Σ можно разбить на классы $U_1, \ldots, U_m$ такие, что $C = \langle U_1, \ldots, U_m \rangle$ — охватывающий цикл, и для C выполнено условие 3 из утверждения леммы. Тогда сопоставим каждой сильно связной компоненте ее охватывающий цикл, получим последовательность охватывающих циклов $C_1, \ldots, C_k$. Так как множества вершин двух различных сильно связных компонент не пересекаются, то будет выполнено условие 1, а так как каждая вершина графа принадлежит хотя бы одной сильно связной компоненте, то будет выполнено условие 2. □

Введем оператор копирования охватывающего цикла $\oplus_k$. Пусть $C_1 = \langle U_1, \ldots, U_n \rangle$, тогда $C_2 = \oplus_k C_1$, если $C_2 = \langle \underbrace{U_1, \ldots, U_n}_{1}, \underbrace{U_1, \ldots, U_n}_{2}, \ldots, \underbrace{U_1, \ldots, U_n}_{k} \rangle$. При этом C_2 также будет являться охватывающим циклом по определению.

Введем оператор склейки охватывающих циклов $\odot$ и покажем, что в результате применения этого оператора также будет получаться охватывающий цикл:

1) Пусть $C_j = \langle \{v\} \rangle$, $C_i = \langle U_1, \ldots, U_m \rangle$ и в v ведет ребро из некоторого U_l, рёбер из v в U_i нет ни для какого i. Тогда $\odot(C_i, C_j) = \langle U_i^{'}, \ldots, U_m^{'} \rangle$, где $U_i^{'} = U_i \cup \{v\}$. Покажем, что для $\odot(C_i, C_j)$ будет выполнено определение охватывающего цикла. Рассмотрим произвольное ребро (u_1, u_2) из $\odot(C_i, C_j)$. Возможны следующие случаи:

   а) $u_1 \in U_i, u_2 \in U_{i+1}$ для некоторого i. В таком случае также выполнено $u_1 \in U_i^{'}, u_2 \in U_{i+1}^{'}$.

   б) $u_1 \in U_l, u_2 = v$. Тогда $u_1 \in U_l^{'}$, а $u_2 \in U_l^{'}$ по построению.

   в) $u_1 = u_2 = v$. Тогда $u_1 \in U_i^{'}$, а $u_2 \in U_{i+1}^{'}$ для любого i по построению.

Таким образом для $\odot(C_i, C_j)$ будет выполнено определение охватывающего цикла.

2) Пусть $C_i = \langle U_1^1, \ldots, U_m^1 \rangle, C_j = \langle U_1^2, \ldots, U_l^2 \rangle$. Пусть $v \in U_a^2$ для некоторого a, а $u \in U_b^1$ для некоторого b и $(u, v) \in E$. Рёбер (u', v') таких, что $u' \in U_{i_1}^2, v' \in U_{j_1}^1$ для некоторых i_1, j_1 не существует. Тогда, если $s = \text{НОК}(m, l)$, то $\odot(C_i, C_j) = \langle U_1^{'}, \ldots, U_s^{'} \rangle$, где $U_t^{'} = U_{b+t}^1 \cup U_{a+t-1}^2$, а $b+t$ и $a+t-1$ берутся по модулю m и l соответственно. Рассмотрим произвольное ребро (u_1, u_2) из $\odot(C_i, C_j)$ и покажем, что $\odot(C_i, C_j)$ будет являться охватывающим циклом.

   а) Пусть $u_1 \in U_k^1, u_2 \in U_{k+1}^1$, где k берется по модулю m. В таком случае для $\odot(C_i, C_j)$ выполнено $u_1 \in U_{k-b}^{'}, u_2 \in U_{k-b+1}^{'}$, где $k-b, k-b+1$ берутся по модулю m.

   б) Пусть $u_1 \in U_k^2, u_2 \in U_{k+1}^2$, где k берется по модулю l. В таком случае для $\odot(C_i, C_j)$ выполнено $u_1 \in U_{k-b+1}^{'}, u_2 \in U_{k-b+2}^{'}$, где $k-b+1, k-b+2$ берутся по модулю l.

   в) Пусть $u_1 \in U_b^1, u_2 \in U_a^2$, тогда $u_1 \in U_0^{'}, u_2 \in U_1^{'}$.

   Таким образом для $\odot(C_i, C_j)$ будет выполнено определение охватывающего цикла.

3) Пусть $C_k = \langle \{v\} \rangle$, $C_i = \langle U_1^1, \ldots, U_m^1 \rangle, C_j = \langle U_1^2, \ldots, U_l^2 \rangle$. Пусть $u_1 \in U_{i-1}^1, u_2 \in U_{j-1}^2$ и $(u_1, v), (u_2, v) \in E$. Тогда, если t — длина предельного цикла, то $C_i^{'} = \oplus_a C_i, C_j^{'} = \oplus_b C_j$, причем $a, b \in \mathbb{Z}$ — такие, что $am = bl = t$. В таком случае, $\odot(C_i, C_j, C_k) = \langle U_1^{'}, \ldots, U_t^{'} \rangle$, где $U_s^{'} = U_{i+s-1}^1 \cup U_{j+s-1}^2$ при $s \neq 1$ и $U_1^{'} = U_i^1 \cup U_j^2 \cup \{v\}$, а $i+s-1$ и $j+s-1$ берутся по модулю m и l соответственно, причем $U_s^1 \in C_i^{'}, U_s^2 \in C_j^{'}, s = \overline{1,t}$ и $|\odot(C_i, C_j, C_k)| = t$. Рассмотрим произвольное ребро (v_1, v_2) из $\odot(C_i, C_j, C_k)$ и покажем, что для $\odot(C_i, C_j, C_k)$ будет выполнено определение охватывающего цикла.

   а) Пусть $v_1 \in U_g^1, v_2 \in U_{g+1}^1$, в таком случае $v_1 \in U_{g-i+1}^{'}, v_2 \in U_{g-i+2}^{'}$, где $g-i+1, g-i+2$ берутся по модулю t.

   б) Пусть $v_1 \in U_g^2, v_2 \in U_{g+1}^2$, в таком случае $v_1 \in U_{g-j+1}^{'}, v_2 \in U_{g-j+2}^{'}$, где $g-j+1, g-j+2$ берутся по модулю t.

   в) Пусть $v_1 = u_1, v_2 = v$, тогда $v_1 \in U_t^{'}, v_2 \in U_1^{'}$

   г) Пусть $v_1 = u_2, v_2 = v$, тогда $v_1 \in U_t^{'}, v_2 \in U_1^{'}$

   Таким образом для $\odot(C_i, C_j, C_k)$ будет выполнено определение охватывающего цикла.

Других случаев быть не может, так как их можно будет разложить на более частные одного из этих трех типов.

Лемма 7. *Если существует последовательность охватывающих циклов графа G_Σ длины k, удовлетворяющая условиям Леммы 6, то существует последовательность охватывающих циклов графа G_Σ длины не более $k-1$, удовлетворяющая условиям Леммы 6.*

Доказательство. Введём топологический порядок сильно связных компонент графа G_Σ. Пусть W — весовая функция, удовлетворяющая условиям:

1) Существуют u_1, v_1 такие, что $u_1 \in V_{S_i}, v_1 \in V_{S_j}$, и существует путь из u_1 в v_1, Не существуют u_2, v_2 такие, что $u_2 \in V_{S_j}, v_2 \in V_{S_i}$, и существует путь из u_2 в v_2. Тогда $W(S_i) > W(S_j)$

2) Для любых различных S_i и S_j $W(S_i) \neq W(S_j)$.

Такая весовая функция будет задавать топологический порядок на сильно связных компонентах.

Введём понятие глубины сильно связной компонинеты S и обозначим её через $L(S)$:

1) Если S_i - такая сильно связная компонента, что не существует ребёр (u, v) таких, что $v \in V_{S_i}, u \notin V_{S_i}$, то $L(S_i) = 0$.

2) Если $S_{i_1}, \ldots, S_{i_k}$ — такие, что существуют ребра $(u_j, v), u_j \in S_j, v \in S_i$ для всех $j \in \{i_1, \ldots, i_k\}$, то $L(S_i) = max(L(S_{i_1}), \ldots, L(S_{i_k})) + 1$

Понятия топологического порядка и глубины для сильно связных компонент естественным образом переносятся на охватывающие циклы. Возьмём в упорядоченной последовательности $C_1, \ldots, C_k$ такой C_j, что $L(C_j) = 1$ и проведем его склейку с такими C_i, что из C_i в C_j ведёт ребро. Как было показано ранее, в таком случае получится последовательность охватывающих циклов длины не более $k-1$. Покажем, что для этой последовательности будут выполнены условия Леммы 6. Так как при склейке два или три охватывающих цикла объединяются в один, а другие не меняются, то условие 1 выполнено. Так как при объединении в новом охватывающем цикле содержатся все вершины, содержащиеся в склеиваемых циклах, а также их копии, то условие 2 будет выполнено. Покажем, что будет выполнено условие 3. Ни один из циклов, из которых есть ребра в C_j не может состоять из одной вершины, так как тогда в нее не вело бы ни одного ребра. Рассмотрим 3 случая, аналогичные случаям из Леммы 6:

1) Если $u, v \in U_i$, то условие 3 выполнено, так как оно было выполнено для C_i. Пусть v — вершина из C_j. Если $\alpha_v(0) = 0$, то $\alpha_v(t) = 0$ для любого t и тогда $u = v$ и условие 3 выполнено. Если $\alpha_v(0) = 1$ и существует путь π такой, что $\pi = \pi(u, v)$ и $|\pi| = s$, а $u, v \in U_i$. Но тогда $\alpha_v(t) = 0$ для любого $t \geqslant s$. Противоречие с тем, что стартовая конфигурация принадлежит предельному циклу. То есть условие 3 выполнено.

2) Если условие 3 было выполнено для C_i и C_j, то единственная ситуация, когда может возникнуть противоречие — если существует $v \in U_j^2, u \in U_i^1, i \neq j$ такие, что:

   а) $u, v \in U_t^{'}$

   б) $\alpha_v(0) = 1, \alpha_u(0) = 0$

   в) существует путь $\pi = \pi(u, v), |\pi| = s$

   Рассмотрим путь π и отрезок (u_1, v) такой, что $u_1 \in C_i$, а $v \in C_j$. Пусть u_2 — такая, что $u_2 \in C_j$ и $(u_2, v) \in E$, а $\pi \setminus (u_1, v) = \{\pi_1, \pi_2\}$ и $|\pi_1| = s_1$. Тогда $\alpha_{u_1}(s_1) = 0$ и если $\alpha_{u_2}(s_1) \neq 0$, то $0_{S_j}(s_1{+}1) = 0_{C_j}(s_1)$ — противоречие с тем, что $0_{C_j}(t) = const$ при $t \geqslant 0$. Таким образом $\alpha_{u_2}(s_1) = 0$, а значит, существует $u^{'} \in U_j^2$ такое, что $\alpha_{u'}(0) = 0$. Противоречие с тем, что условие 3 Леммы 6 выполнено для C_j. А значит, таких u и v быть не может.

3) Если условие 3 было выполнено для C_i и C_j, то оно также будет выполнено для $C_i^{'}$ и $C_j^{'}$. Так как ребер, ведущих из C_i в C_j нет, то если существует $\pi = \pi(u_1, u_2)$ такой, что $\alpha_{u_1}(0) = 0, \alpha_{u_2}(0) = 1, u_1, u_2 \in U_k^{'}$, то $u_2 = v$, а $u_1 \in C_i$ или $u_1 \in C_j$. Не ограничивая общности считаем, что $u_1 \in U_i^1$ и $h = |\pi|$. Если $\alpha_v(0) = 1$, то $\alpha_v(pt) = 1$ для любого $p \in \mathbb{N} \cup \{0\}$. Но $h = q * t$, где $q \in \mathbb{N} \cup \{0\}$, а значит, если $\alpha_{u_1}(0) = 0$, то $\alpha_v(h) = 0$. Противоречие.

Если $C_k = \langle U_1^3, \ldots, U_s^3 \rangle$, $C_i = \langle U_1^1, \ldots, U_m^1 \rangle, C_j = \langle U_1^2, \ldots, U_l^2 \rangle$, то проводится склейка с C_i или C_j согласно пункту 2. Таким образом получили последовательность длины на 1 меньше, удовлетворяющую условиям Леммы 6. □

Теперь мы можем доказать Теорему 1. Так как из Лемм 6 и 7 следует, что существует один охватывающий цикл $= \langle U_1, \ldots, U_m \rangle$, для которого выполнены условия Леммы 6, то необходимость условий Теоремы 1 доказана. Теперь докажем достаточность условий Теоремы 1.

Пусть существуют $U_1, \ldots, U_m$ — множества вершин графа G_Σ, удовлетворяющие условиям 1 и 2 Теоремы 1. Разобьем каждое из множеств

$U_i = U_i^0 \cup U_i^1$, где $U_i^0 = \{v \in U_i \mid \alpha_v(0) = 0\}$, а $U_i^1 = \{v \in U_i \mid \alpha_v(0) = 1\}$. Введем множество вершин $U_j^0(k)$:

$$U_j^0(0) = U_j^0 \text{ и } U_j^0(k+1) = \{v \in U_{j+k+1} | \ \exists u \in U_j^0(k) : (u,v) \in E\} \text{ для всех } k \geq 0.$$

Предположим, что $v \in U_j^0(m)$. Докажем по индукции, что тогда существует путь длины m из $u \in U_j^0(0)$ в v. Для $m = 0$ это очевидно. Рассмотрим вершину v, до которой существует путь длины $m+1$ из $U_j^0(0)$, тогда по предположению индукции, если (w, v) — последнее ребро этого пути, то $w \in U_j^0(m)$. Но тогда $v \in U_j^0(m+1)$ по определению этого класса. Так как существует путь из $u \in U_j^0(0)$ в $v \in U_j^0(m)$, то из условия теоремы следует, что $\alpha_v(0) = 0$, а значит $v \in U_j^0(0)$. Таким образом $U_j^0(m) \subseteq U_j^0(0)$.

Пусть $v \in U_j^0(0)$, то есть $\alpha_v(0) = 0$, тогда из условия теоремы существует путь из некоторой $u \in U_j^0(0)$ в v. Так как каждое ребро $(u_1, u_2) \in E$ этого пути лежит между соседними классами U_l и U_{l+1}, где $l \in 1, \ldots, m$, то для любого пути $\pi = \pi(u, v)$, где $u, v \in U_j$, длина пути $|\pi| = km$. Пусть $k \geq 2$, тогда рассмотрим $u^{'} \in U_j$ такую, что $\pi_1 = \pi_1(u, u^{'})$ и $|\pi_1| = (k-1)m$, а $\pi_2 = \pi_2(u^{'}, v)$ и $|\pi_2| = m$. Предположим, что $\alpha_{u'}(0) = 1$, тогда π_1 — путь из $u \in U_j^0$ в $u^{'} \in U_j^1$ — противоречие с условиями теоремы, а значит $\alpha_{u'} = 0$ и π_2 — путь длины m из $u^{'} \in U_j^0(0)$ в $v \in U_j^0(0)$. Докажем индукцией по k, что если $u \in U_j^0(0)$ и существует путь в G_Σ длины k из u в v, то $v \in U_j^0(k)$. Для $k = 0$ это очевидно. Рассмотрим вершину v, до которой существует путь длины $k+1$ из $U_j^0(0)$, тогда по предположению индукции, если (w, v) — последнее ребро этого пути, то $w \in U_j^0(k)$. Но тогда $v \in U_j^0(k+1)$ по определению этого класса. Таким образом, так как $u^{'} \in U_j^0(0)$ и существует путь длины m из $u^{'}$ в v, то $v \in U_j^0(m)$, то есть $U_j^0(0) \subseteq U_j^0(m)$.

Тогда $U_j^0(m) = U_j^0(0)$, а значит $\alpha_v(m) = 0 \Leftrightarrow \alpha_v(0) = 0$. Так как $U_j^1 = U_j \setminus U_j^0$, то $\alpha_v(m) = 1 \Leftrightarrow \alpha_v(0) = 1$. Следовательно, в силу произвольности j, $\alpha_v(0) = \alpha_v(m)$ для любой v, а значит $\alpha(m) = \alpha(0)$, то есть спустя m тактов попадем в начальную конфигурацию, а значит, она будет принадлежать предельному циклу.

Список литературы

[1] С.В.Моисеев, "О реализации автоматов нейронными сетями", *Журнал Интеллектуальные системы*, **12** (2008), 283-316.

[2] Дробышев А.С., Боков Г.В., "Критерий нейропорождённости автоматных функций с задержкой", *Вестник Московского университета. Серия 1: Математика. Механика*, **6** (2020), 54-55.

[3] Дробышев А.С., “Реализация схем из функциональных элементов с задержкой нейронными сетями”, *Материалы Международного молодежного научного форума «ЛОМОНОСОВ-2023»*, 2023.

[4] С.В.Яблонский, *Введение в дискретную математику*, «Наука», Москва, 1979, 272 с.

[5] Warren S. McCulloch, Walter Pitts, “A logical calculus of the ideas immanent in nervous activity”, *Bulletin of Mathematical Biophysics*, **5** (1943), 115–133

[6] Kleene S.C., “Representation of Events in Nerve Nets and Finite Automata”, *Automata Studies.Princeton University Press*, 1956, 3-42

[7] Siegelmann H.T., “Reccurent Neural Networks and Finite Automata”, *Communications of the ACM*, **12** (1996), 567-574

[8] Siegelmann, H. T., Sontag E. D., “Turing Computability with Neural Networks”, *Applied Mathematics Letters*, **6** (1991), 77-80

[9] Twining C.J., “The Limiting Behavior of Non-cylindrical Elementary Cellular Automata”, *Complex Systems*, **6** (1992), 417-432

On limit cycles of homogeneous neural networks
Drobyshev A.S.

The paper considers a formal model of neural networks in which each neuron is represented as an automaton consisting of a threshold Boolean function and a time delay. The criterion of belonging of the starting configuration to the limit cycle is proved.

Keywords: neural networks, threshold functions, boolean circuits, limit cycles.

References

[1] S. V. Moiseev, “On the implementation of automata by neural networks”, *Intelligent systems*, **12** (2008), 283-316

[2] A. S. Drobyshev, G. V. Bokov, “Criterion of Neural Generation of Automaton Functions with Time Delay”, *Bulletin of the Moscow University. Series 1: Mathematics. Mechanics*, **6** (2020), 54-55

[3] A. S. Drobyshev, “Implementation of circuits of functional elements with time delay by neural networks”, *Materials of the International Youth Scientific Forum "LOMONOSOV-2023"*, 2023

[4] C. V. Yablonsky, *Introduction to Discrete Mathematics*, «Science», Moscow, 1979, 272 pp.

[5] Warren S. McCulloch, Walter Pitts, "A logical calculus of the ideas immanent in nervous activity", *Bulletin of Mathematical Biophysics*, **5** (1943), 115–133

[6] Kleene S.C., "Representation of Events in Nerve Nets and Finite Automata", *Automata Studies.Princeton University Press*, 1956, 3-42

[7] Siegelmann H.T., "Reccurent Neural Networks and Finite Automata", *Communications of the ACM*, **12** (1996), 567-574

[8] Siegelmann, H. T., Sontag E. D., "Turing Computability with Neural Networks", *Applied Mathematics Letters*, **6** (1991), 77-80

[9] Twining C.J., "The Limiting Behavior of Non-cylindrical Elementary Cellular Automata", *Complex Systems*, **6** (1992), 417-432

О взаимосвязи криптографически важных свойств конечных квазигрупп

Р. А. Жигляев[1]

В данной работе устанавливается взаимосвязь между некоторыми свойствами конечных квазигрупп. Доказывается, что в случае квазигрупп простого порядка из бесформенности следует полиномиальная полнота. Даны примеры, показывающие, что обратное утверждение и обобщение до составных порядков не являются верными.

Ключевые слова: конечная квазигруппа, полиномиальная полнота, бесформенность.

1. Введение

Некоммутативные и неассоциативные алгебраические структуры играют важную роль в построении криптографических алгоритмов [1]. Один из примеров таких структур – квазигруппы. В алгоритмах GAGE и InGAGE [2], участвовавших в конкурсе Lightweight Cryptography от NIST, используются e-преобразования и d-преобразования, основанные на одной из квазигрупп порядка 4. В конкурсе SHA-3 были квазигрупповые кандидаты Edon-R' [3] и NaSHA [4]. Основанное на квазигруппах табличное гаммирование обладает свойством совершенной секретности [5]. В работе [6] приводится обзор применения квазигрупп в построении однонаправленных функций, A-кодов, и методов шифрования. В [7] представлен более широкий обзор применения квазигрупп в криптографии.

У каждого криптографического алгоритма могут быть свои требования к используемым квазигруппам. Среди наиболее часто встречаемых требований можно выделить следующие: полиномиальная полнота, отсутствие собственных подквазигрупп, бесформенность [8, 9].

Полиномиальная полнота гарантирует NP-полноту задачи проверки разрешимости уравнений и систем уравнений [10, 11]. Ранее упомянутый алгоритм NaSHA требует квазигруппы больших размеров. Наличие подквазигрупп может снизить стойкость таких алгоритмов. Требование к отсутствию собственных подквазигрупп также является частью более общего набора требований, называемого бесформенностью. Аккуратные определения всех этих понятий будут даны в разделе 2.

[1] *Жигляев Родион Алексеевич* — аспирант каф. математической теории интеллектуальных систем мех.-мат. ф-та МГУ, e-mail: rzhiglyaev@mail.ru.

Zhigliaev Rodion Alekseevich — graduate student, Lomonosov Moscow State University, Faculty of Mechanics and Mathematics, Chair of Mathematical Theory of Intellectual Systems.

В данной работе устанавливается взаимосвязь между понятиями бесформенности и полиномиальной полноты.

Все эксперименты в рамках данной работы проводились с использованием программы https://github.com/Gerror/Quasigroup.

Автор выражает благодарность А.В. Галатенко за постановку задачи и помощь в работе.

2. Основные определения

Введем основные определения.

Определение 1. *Конечное множество Q, на котором задана бинарная операция умножения $f\colon Q \times Q \to Q$, такая, что для любых элементов $a, b \in Q$ уравнения $f(a, x) = b$ и $f(y, a) = b$ однозначно разрешимы в Q, называется конечной квазигруппой. Операцию f будем называть квазигрупповой.*

Далее слово "конечная" будем опускать, предполагая, что речь всегда идет о конечных квазигруппах. Вместо символа f иногда для удобства будем обозначать квазигрупповую операцию символом умножения $*$.

Определение 2. *Пусть задана квазигруппа (Q, f) и $Q' \subset Q$, $1 \leq |Q'| < |Q|$. Если для любых элементов $a, b \in Q'$ верно, что $f(a, b) \in Q'$, то будем говорить, что квазигруппа (Q, f) содержит собственную подквазигруппу $(Q', f_{Q'})$, где $f_{Q'}$ — ограничение операции f на $Q' \times Q'$.*

Определение 3. *Квазигруппа (Q, f) называется аффинной, если на множестве Q можно ввести структуру абелевой группы $(Q, +)$, такую, что существуют автоморфизмы α, β группы $(Q, +)$ и элемент $c \in Q$, для которых выполнено тождество*

$$f(x, y) = \alpha(x) + \beta(y) + c.$$

Рассмотрим множество элементов квазигруппы $Q = \{q_1, ..., q_N\}$, $N \geq 2$ и некоторое разбиение α множества Q в объединение непересекающихся подмножеств $Q = A_1 \sqcup ... \sqcup A_m$. Будем называть разбиение α нетривиальным, если $m > 1$, $A_i \neq \emptyset$, $i = 1, ..., m$, и существует индекс j, $1 \leq j \leq m$, такой, что $|A_j| > 1$. В случае если $|A_1| = ... = |A_m|$, такое нетривиальное разбиение будем называть равномерным. Элементы a, b, которые принадлежат одному множеству A_i, далее назовем эквивалентными и будем использовать запись $a \sim b$.

Будем говорить, что f сохраняет разбиение α, если для любой пары наборов (a_1, b_1), $(a_2, b_2) \in Q^2$, таких, что $a_i \sim b_i$, $i = 1, 2$, выполнено $f(a_1, a_2) \sim f(b_1, b_2)$. Как можно заметить, квазигрупповые операции могут сохранять только равномерные разбиения.

Определение 4. *Квазигруппа (Q, f) называется простой, если операция f не сохраняет никакое нетривиальное разбиение Q.*

Для фиксированного (конечного) множества A обозначим через $\mathcal{O}_n(A)$ совокупность всех n-арных операций на A ($n \geq 0$). Под 0-арными операциями будем подразумевать константы. Пусть $\mathcal{O}(A) = \bigcup_{n=0}^{\infty} \mathcal{O}_n(A)$. Далее под множеством A понимается множество элементов квазигруппы, поэтому будем использовать упрощённую запись: $\mathcal{O}_n$ и $\mathcal{O}$.

Стандартным образом введем операции суперпозиции и замыкания [12]. Обозначим замыкание множества F через $[F]$.

Определение 5. *Квазигруппа Q называется полиномиально полной, если $[\{f\} \cup \mathcal{O}_0] = \mathcal{O}$.*

Известно, что полиномиальная полнота эквивалентна одновременной простоте и неаффинности [13].

Определение 6. *Квазигруппа $(Q, *)$ порядка N называется бесформенной, если:*

- *квазигруппа не идемпотентна, т.е. $\exists x \in Q$, т.ч. $x * x \neq x$;*
- *квазигруппа не коммутативна, т.е. $\exists x, y \in Q$, т.ч. $x * y \neq y * x$;*
- *квазигруппа не ассоциативна, т.е. $\exists x, y, z \in Q$, т.ч. $(x * y) * z \neq x * (y * z)$;*
- *квазигруппа не содержит ни левой, ни правой единицы, т.е. не существует элементов $e_1, e_2 \in Q$, т.ч. $\forall x \in Q\ e_1 * x = x$, $x * e_2 = x$;*
- *квазигруппа не содержит собственных подквазигрупп;*
- *не существует $k < 2N$, при котором выполняются тождества*

$$\underbrace{x * (x \ldots * (x}_{k} * y)) = y,\ y = ((y * \underbrace{x) * \ldots x) * x}_{k}\ \forall x, y \in Q.$$

Определение 7. *Автоморфизм группы G называется регулярным, если он оставляет неподвижным только тривиальный элемент из G.*

3. Сложность проверки свойств

В данном разделе при подсчете сложности будем предполагать, что квазигруппы заданы таблично, а вычисление умножения в квазигруппе – это элементарная операция.

Сложность проверки бесформенности зависит от самого сложного из проверяемых свойств. Очевидно, идемпотентность легко проверить со сложностью $O(N)$, перебрав все элементы квазигруппы. Для проверки коммутативности достаточно перебрать все пары элементов квазигруппы. Сделать это можно со сложностью $O\left(N^2\right)$. Для проверки ассоциативности можно воспользоваться процедурой, называемой тестом Лайта.

Теорема 1 ([14]). *Пусть G — множество с заданной операцией умножения $*$, и у G есть порождающее множество S. Тогда для проверки ассоциативности операции $*$ достаточно проверить тождества $x*(g*y)$ и $(x*g)*y$ для всех $x, y \in G$ и $g \in S$.*

Теорема 2 ([15]). *Пусть G — квазигруппа порядка N. Тогда в G можно выделить порождающее множество S размером не больше $\lfloor \log_2 N \rfloor + 1$.*

Из этих утверждений нетрудно установить, что ассоциативность проверяется со сложностью $O\left(N^2 \log_2 N\right)$.

Проверить что какой-то элемент квазигруппы является левой или правой единицей можно со сложностью $O(N)$. Таким образом, для квазигруппы порядка N алгоритм нахождения левой или правой единицы можно реализовать за $O\left(N^2\right)$, проверив каждый элемент квазигруппы.

Для поиска собственных подквазигрупп можно использовать следующее утверждение.

Теорема 3 ([16]). *Существует алгоритм, который устанавливает наличие собственных подквазигрупп в квазигруппе порядка N с временной сложностью $O\left(N^{7/3} \cdot (\log N)^{2/3}\right)$ и пространственной сложностью $O\left(N^2\right)$, $N \to \infty$.*

В работе [17] было анонсировано, что временную сложность поиска подквазигрупп можно понизить до $O\left(N^{7/3}\right)$.

Проверку тождеств из определения бесформенности можно произвести явно. Для вычисления одного равенства с m операциями умножения требуется m действий. Чтобы проверить одно тождество с m умножениями необходимо перебрать все пары элементов квазигруппы и для каждой пары вычислить произведение длины m. В худшем случае придется проверить все тождества, т.е. выполнить $N^2 + 2N^2 + 3N^2 + ... + 2N * N^2$ действий. Таким образом, сложность проверки тождеств из определенеия квазигруппы составляет $O\left(N^4\right)$. Поскольку этот шаг является наиболее сложным, мы получаем следующее утверждение.

Теорема 4. *Сложность процедуры проверки бесформенности $O\left(N^4\right)$, где N — порядок квазигруппы.*

Отметим, что тождества можно проверять более оптимальным образом. Если для каждого тождества с m операциями умножения хранить результаты вычисления тождества с $m-1$ операцией умножения, то временную сложность алгоритма можно понизить до $O\left(N^3\right)$. Это также повысит пространственную сложность до $O\left(N^2\right)$. Однако, в случае табличного задания квадратичная память требуется на задание операции, поэтому такая пространственная сложность алгоритма не является существенной. Таким образом, можно сформулировать следующее утверждение.

Теорема 5. *Существует процедура проверки бесформенности, имеющая временную сложность $O\left(N^3\right)$ и пространственную сложность $O\left(N^2\right)$. Здесь N — порядок квазигруппы.*

Теорема 6 ([18])**.** *Сложность процедуры проверки аффинности $O\left(N^3\right)$, где N — порядок квазигруппы.*

Отметим, что наибольшую сложность в алгоритме проверки аффинности имеет этап проверки ассоциативности. Воспользовавшись ранее упомянутым тестом Лайта можно понизить сложность проверки аффинности до $O\left(N^2 \log_2 N\right)$.

Теорема 7 ([18, 19])**.** *Сложность процедуры проверки простоты $O\left(N^3\right)$, где N — порядок квазигруппы.*

Таким образом, полиномиальную полноту можно проверить со сложностью $O\left(N^3\right)$.

4. Взаимосвязь криптографически важных свойств

Теорема 8. *Пусть $(Q,*)$ квазигруппа порядка p, где p – простое число, $p \geq 5$. Если $(Q,*)$ бесформенна, то она полиномиально полна.*

Доказательство. Предположим противное. Тогда существует биективное отображение $\varphi : Q \to \mathbb{Z}_p$, такое что $\varphi(x*y) = a\varphi(x) + b\varphi(y) + c$, где $a, b \in \mathbb{Z}_p \backslash \{0\}$, а сложение и умножение на скаляры ведутся по модулю p [20]. Несложно проверить, что

$$\varphi(\underbrace{x*(x...*(x}_{k}*y))) = a\varphi(x) + c + \textstyle\sum_{i=1}^{k-1} b^i(a\varphi(x)+c) + b^k\varphi(y).$$

Возьмем $k = p-1$. По малой теореме Ферма $b^k = 1 \pmod p$. Обозначим

$$\alpha(x) = a\varphi(x) + c + \textstyle\sum_{i=1}^{k-1} b^i(a\varphi(x)+c).$$

Перепишем $a\varphi(x)+c$ как $b^k(a\varphi(x)+c)$ и вынесем b за скобки. Тогда

$$\alpha(x) = b\alpha(x).$$

Отсюда либо $b = 1$, либо $\alpha(x) \equiv 0$. Рассмотрим оба случая.

1) Пусть $b = 1$. Поскольку $(Q, *)$ бесформенна, то она не содержит левой единицы. Значит для любого x можно найти y, такой, что $x * y \neq y$. Т.е. $\varphi(x * y) = a\varphi(x) + \varphi(y) + c \neq \varphi(y)$. Значит для любого x верно, что $a\varphi(x) + c \neq 0$. Но поскольку φ биекция, такой x всегда можно найти. Таким образом, если $b = 1$, то $(Q, *)$ содержит левую единицу и не может быть бесформенной.

2) Пусть теперь $\alpha(x) \equiv 0$. Тогда $\varphi(\underbrace{x * (x... * (x}_{p-1} * y))) = \varphi(y)$. Но это значит, что $\underbrace{x * (x... * (x}_{p-1} * y)) = y$. Длина этого произведения $p - 1 < 2p$. Следовательно, квазигруппа не будет бесформенной.

Аналогично доказывается, что либо $a = 1$ и тогда квазигруппа содержит правую единицу, либо верно тождество $y = ((y * \underbrace{x) * ...x) * x}_{k}$. Следовательно, квазигруппа не может не быть полиномиально полной. □

Замечание 1. *При $p = 2, 3$ все квазигруппы не полиномиально полные и не бесформенные. Поэтому, формально, теорема верна при любом простом порядке.*

Покажем, что в обратную сторону теорема не верна.

Рассмотрим квазигруппу $(Q, *)$ порядка 5, где $Q = \{0, 1, 2, 3, 4\}$, а умножение задано таблицей:

*	0	1	2	3	4
0	2	4	3	1	0
1	3	0	4	2	1
2	4	3	1	0	2
3	1	2	0	4	3
4	0	1	2	3	4

Очевидно, эта квазигруппа простая, поскольку её порядок простое число. Воспользуемся алгоритмом проверки аффинности из работы [18] и покажем, что эта квазигруппа неаффинна. Построим латинский квадрат L', в котором при каждом $i = 1, 2, 3, 4, 5$ строка с номером i содержит перестановку $\sigma_i \cdot \sigma_1^{-1}$, где σ_i – перестановка, соответствующая i-й строке

исходного латинского квадрата. Тогда L':

*'	0	1	2	3	4
0	0	1	2	3	4
1	1	2	3	4	0
2	2	0	4	1	3
3	3	4	1	0	2
4	4	3	0	2	1

Следующим шагом необходимо построить матрицу L'' из матрицы L' перестановкой строк, такой, что первый столбец L'' совпадает с первой строкой. Но в матрице L' уже первая строка совпадает с первым столбцом. Поэтому матрицы L' и L'' и задаваемые ими операции совпадают. Эта матрица не симметрична: $1 *'' 2 \neq 2 *'' 1$. Следовательно, квазигруппа $(Q, *)$ неаффинна. Поскольку квазигруппа простая и не аффинная, то она является полиномиально полной. При этом она не удовлетворяет сразу нескольким свойствам из определения бесформенности:

- 4 является левой и правой единицей;
- верно тождество $((((((y * x) * x) * x) * x) * x) * x) = y$;
- $4 * 4 = 4$, т.е. $\{4\}$ является подквазигруппой.

Аналогично можно привести пример полиномиально полной идемпотентной квазигруппы:

*	0	1	2	3	4
0	0	4	3	1	2
1	2	1	4	0	3
2	3	0	2	4	1
3	4	2	1	3	0
4	1	3	0	2	4

и пример полиномиально полной коммутативной квазигруппы:

*	0	1	2	3	4
0	1	3	2	0	4
1	3	4	0	1	2
2	2	0	3	4	1
3	0	1	4	2	3
4	4	2	1	3	0

При помощи ранее упомянутой программной реализации алгоритма проверки полиномильной полноты было выявлено, что знакопеременная

группа A_5 является полиномиально полной. При этом, A_5, очевидно, ассоциативна. Эксперимент осуществлялся с использованием ранее упомянутых алгоритмов проверки аффинности, простоты и бесформенности.

Покажем, что в случае составного порядка можно построить бесформенную квазигруппу, не являющуюся полиномиально полной. Рассмотрим следующую квазигруппу порядка 4:

*	0	1	2	3
0	2	0	3	1
1	1	2	0	3
2	0	3	1	2
3	3	1	2	0

Возьмем 4 вспомогательных квазигруппы порядка 4:

$*_0$	0	1	2	3
0	1	3	2	0
1	2	0	3	1
2	0	2	1	3
3	3	1	0	2

$*_1$	4	5	6	7
4	7	5	6	4
5	6	4	5	7
6	4	6	7	5
7	5	7	4	6

$*_2$	8	9	10	11
8	9	8	10	11
9	10	11	9	8
10	8	10	11	9
11	11	9	8	10

$*_3$	12	13	14	15
12	13	15	12	14
13	14	12	13	15
14	12	14	15	13
15	15	13	14	12

Несложно проверить, что все 5 квазигрупп являются бесформенными. Заменим в исходной таблице умножения элемент 0 на таблицу умножения $*_0$, элемент 1 на таблицу $*_1$, элемент 2 на таблицу $*_2$, элемент 3 на таблицу $*_3$. Получим следующую квазигруппу порядка 16:

*	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	9	8	10	11	1	3	2	0	13	15	12	14	7	5	6	4
1	10	11	9	8	2	0	3	1	14	12	13	15	6	4	5	7
2	8	10	11	9	0	2	1	3	12	14	15	13	4	6	7	5
3	11	9	8	10	3	1	0	2	15	13	14	12	5	7	4	6
4	7	5	6	4	9	8	10	11	1	3	2	0	13	15	12	14
5	6	4	5	7	10	11	9	8	2	0	3	1	14	12	13	15
6	4	6	7	5	8	10	11	9	0	2	1	3	12	14	15	13
7	5	7	4	6	11	9	8	10	3	1	0	2	15	13	14	12
8	1	3	2	0	13	15	12	14	7	5	6	4	9	8	10	11
9	2	0	3	1	14	12	13	15	6	4	5	7	10	11	9	8
10	0	2	1	3	12	14	15	13	4	6	7	5	8	10	11	9
11	3	1	0	2	15	13	14	12	5	7	4	6	11	9	8	10
12	13	15	12	14	7	5	6	4	9	8	10	11	1	3	2	0
13	14	12	13	15	6	4	5	7	10	11	9	8	2	0	3	1
14	12	14	15	13	4	6	7	5	8	10	11	9	0	2	1	3
15	15	13	14	12	5	7	4	6	11	9	8	10	3	1	0	2

Эта квазигруппа не будет простой, так как квазигрупповая операция сохраняет разбиение $\{0,1,2,3\} \cup \{4,5,6,7\} \cup \{8,9,10,11\} \cup \{12,13,14,15\}$. Однако, она состоит из бесформенных блоков, а значит некоммутативна, неассоциативна, не содержит левых и правых единиц и неидемпотентна. Несложно проверить, что она также не содержит подквазигрупп и в ней не выполняются тождества $\underbrace{x * (x... * (x}_{k} * y)) = y$, $y = (\underbrace{(y * x) * ...x) * x}_{k}$ $\forall k < 32$. Следовательно, это бесформенная, но не полиномиально полная квазигруппа.

Однако, в случае составного порядка можно сформулировать ряд утверждений, когда отсутствие полиномиальной полноты влечет за собой отсутствие бесформенности.

Утверждение 1. *Пусть $(Q, *)$ аффинная квазигруппа над абелевой группой $(Q, +)$. Тогда $(Q, *)$ содержит правую единицу тогда и только тогда, когда α тривиальный автоморфизм.*

Доказательство. Квазигруппа $(Q, *)$ содержит правую единицу тогда и только тогда, когда существует элемент e, такой, что $x * e = x$ $\forall x \in Q$. Поскольку квазигруппа аффинна, это значит, что $\alpha(x) + \beta(e) + c = x$ $\forall x \in Q$. В частности, это верно для нейтрального элемента e' группы $(Q, +)$. Следовательно, $\beta(e) = -c$ и $\alpha(x) = x$. Таким образом, из существования правой единицы, следует, что α тождественный автоморфизм. А если α тождественный автоморфизм, то элемент $\beta^{-1}(-c)$ является правой единицей. □

Утверждение 2. *Пусть $(Q, *)$ аффинная квазигруппа над абелевой группой $(Q, +)$. Тогда $(Q, *)$ содержит левую единицу тогда и только тогда, когда β тривиальный автоморфизм.*

Доказательство. Аналогично предыдущему утверждению. □

Утверждение 3. *Пусть $(Q, *)$ аффинная квазигруппа над абелевой группой $(Q, +)$. Тогда $(Q, *)$ ассоциативна тогда и только тогда, когда α и β тождественные автоморфизмы.*

Доказательство. Известно, что квазигруппа ассоциативна тогда и только тогда, когда она группа [21]. В частности, это значит, что в ассоциативной квазигруппе есть единица. По предыдущим утверждениям это возможно только в том случае, когда α и β тождественные автоморфизмы. И наоборот, если α и β, то, очевидно, элемент $-c$ является единицей в квазигруппе. □

Утверждение 4. *Пусть* $(Q,*)$ *аффинная квазигруппа над абелевой группой* $(Q,+)$. *Тогда* $(Q,*)$ *коммутативна тогда и только тогда, когда* $\alpha \equiv \beta$.

Доказательство. Аффинная квазигруппа $(Q,*)$ коммутативна тогда и только тогда, когда $\alpha(x)+\beta(y)+c=\alpha(y)+\beta(x)+c\ \forall x,y\in Q$. Отсюда следует, что $\alpha(x-y)=\beta(x-y)\ \forall x,y\in Q$. Любой элемент из Q можно представить как разность некоторых элементов x,y из Q. Следовательно, α и β это один и тот же автоморфизм. □

Утверждение 5. *Пусть* $(Q,*)$ *аффинная квазигруппа над абелевой группой* $(Q,+)$. *Тогда* $(Q,*)$ *идемпотентна тогда и только тогда, когда* $\alpha(x)=x-\beta(x)$, *а* c – *единица группы* $(Q,+)$.

Доказательство. Аффинная квазигруппа $(Q,*)$ идемпотентна тогда и только тогда, когда $\alpha(x)+\beta(x)+c=x$. Подставив в это тождество единичный элемент e группы $(Q,+)$ получим, что $c=e$. Таким образом, квазигруппа идемпотентна только в тех случаях, когда $c=e$ и $\alpha(x)+\beta(x)=x$. □

Теорема 9. *Пусть* $(Q,*)$ *аффинная квазигруппа над абелевой группой* $(Q,+)$ *порядка* N, *а* α *и* β – *регулярные автоморфизмы. Тогда* $(Q,*)$ *небесформенна.*

Доказательство. По индукции несложно показать, что

$$\underbrace{x*(x...*(x}_{k}*y))=\alpha(x)+c+\textstyle\sum_{i=1}^{k-1}(\beta^{(i)}(\alpha(x)+c))+\beta^{(k)}(y).$$

Возьмем $k=|\beta|$, где $|\beta|$ – порядок автоморфизма β. Тогда $\beta^{(k)}(y)=y$. Обозначим

$$\gamma(x)=\alpha(x)+c+\textstyle\sum_{i=1}^{k-1}(\beta^{(i)}(\alpha(x)+c)).$$

Запишем $\alpha(x)+c$ как $\beta^{(k)}(\alpha(x)+c)$ и воспользуемся тем, что β гомоморфизм. Тогда $\gamma(x)=\beta(\gamma(x))$. Поскольку β регулярный автоморфизм, то $\gamma(x)\equiv 0$. Следовательно, $\underbrace{x*(x...*(x}_{k}*y))=y$. Аналогично можно показать, что при $k=|\alpha|$ верно тождество $y=((y*\underbrace{x)*...x)*x}_{k}$. Здесь $|\alpha|$ – порядок автоморфизма α. Поскольку $|\alpha|,\ |\beta|<N-1<2N$ [22] квазигруппа небесформенна. □

С поиском бесформенных не полиномиально полных квазигрупп также был проведен эксперимент. Алгоритмом Джейкобсона-Мэтьюза [23] было

сгенерировано по 1000000 случайных квазигрупп порядков 6, 8 и 10. Все бесформенные квазигруппы среди них были полиномиально полными.

Существует всего 576 квазигрупп порядка 4. В ходе эксперимента было установлено, что среди них 384 полиномиально полных квазигруппы и 48 бесформенных квазигрупп. Более подробная классификация:

- 48 бесформенных полиномиально полных;
- 0 бесформенных не полиномиально полных;
- 336 не бесформенных полиномиально полных;
- 192 не бесформенных не полиномиально полных.

На основании результатов экспериментов можно предположить, что значительная часть бесформенных квазигрупп являются полиномиально полными. Обоснование этого вывода для квазигрупп составного порядка является направлением дальнейших исследований.

5. Заключение

В работе было установлено, что все бесформенные квазигруппы простого порядка являются полиномиально полными. Был приведен пример, показывающий, что обобщение этого утверждения на составные порядки невозможно. Однако, эксперименты показывают, что такие примеры по всей видимости являются исключительными ситуациями и значительное число бесформенных квазигрупп являются полиномиально полными. Кроме того, было показано, что существуют полиномиально полные квазигруппы, не являющиеся бесформенными. Для каждого свойства из определения бесформенности был приведен пример полиномиально полной квазигруппы, которая этим свойством не обладает.

Список литературы

[1] Markov V. T., Mikhalev A. V., Nechaev A. A., “Nonassociative algebraic structures in cryptography and coding”, *Journal of Mathematical Sciences*, **245**:2 (2020), 178–196.

[2] Gligoroski D., “On the S-box in GAGE and InGAGE”, 2019, http://gageingage.org/upload/LWC2019NISTWorkshop.pdf.

[3] Gligoroski D., Ødegård R. S., Mihova M., Knapskog S. J., Drápal A., Klima V., Amundsen J., El-Hadedy M., “Cryptographic hash function

EDON-R'", *Proceedings of the 1st International Workshop on Security and Communication Networks*, 2009, 1–9.

[4] Markovski S., Mileva A., "NaSHA — family of cryptographic hash functions", *The First SHA-3 Candidate Conference*, 2009.

[5] Shannon C., "Communication theory of secrecy systems", *Bell System Technical Journal*, **28**:4 (1949), 656–715.

[6] Глухов М. М., "О применениях квазигрупп в криптографии", *Прикладная дискретная математика*, 2008, № 2(2), 28–32.

[7] Shcherbacov V. A., "Quasigroups in cryptology", *Computer Science Journal of Moldova*, **17**:2 (2009), 193–228.

[8] Artamonov V. A., Chakrabarti S., Pal S. K., "Characterization of polynomially complete quasigroups based on Latin squares for cryptographic transformations", *Discrete Applied Mathematics*, **200** (2016), 5–17.

[9] Markovski S., "Design of crypto primitives based on quasigroups", *Quasigroups and Related Systems*, **23** (2015), 41–90.

[10] Horváth G., Nehaniv Gh. L., Szabó Cs., "An assertion concerning functionally complete algebras and NP-completeness", *Theoretical Computer Science*, **407** (2008), 591–595.

[11] Larose B., Zadori L., "Taylor terms, constraint satisfaction and the complexity of polynomial equations over finite algebras", *International Journal of Algebra and Computation*, **16** (2006), 563–581.

[12] Яблонский С. В., "Введение в дискретную математику", *Наука*, 1986.

[13] Chaplygina S. S., Galatenko A. V., "Polynomial completeness and completeness of finite n-quasigroups", *Quasigroups and Related Systems*, **32**:2 (2024), 207–223.

[14] Clifford A., Preston G., "Light's associativity test", *The Algebraic Theory of Semigroups*, **1** (1961), 7–8.

[15] Tarjan R. E., "Determining whether a groupoid is a group", *Information Processing Letters*, **1** (1972), 120–124.

[16] Галатенко А. В., Панкратьев А. Е., Староверов В. М., "Об одном алгоритме проверки существования подквазигрупп", *Чебышевский сборник*, **22**:2 (2021), 76–89.

[17] Galatenko A. V., Mazurin A. D., Pankratiev A. E., Zhigliaev R. A., "Efficient verification of some properties of finite quasigroups", *Mathematics in Armenia: advances and perspectives*, 2023, 29–30.

[18] Галатенко А. В., Панкратьев А. Е., "О сложности проверки полиномиальной полноты конечных квазигрупп", *Дискретная математика*, **30**:4 (2018), 3–11.

[19] Galatenko A. V., Pankratiev A. E., Staroverov V. M., "Efficient verifiation of polynomial completeness of quasigroups", *Lobachevskii Journal of Mathematics*, 2020, 1444–1453.

[20] Галатенко А. В., Панкратьев А. Е., Родин С. Б., "О полиномиально полных квазигруппах простого порядка", *Алгебра и логика*, **57**:5 (2018), 509–521.

[21] Prasad V. B. V. N., Venkateswara Rao J., "Characterization of Quasigroups and Loop", *International Journal of Scientific and Innovative Mathematical Research*, **1**:2 (2013), 95–102.

[22] Хорошевский М. В., "Об автоморфизмах конечных групп", *Математический сборник*, **135**:4 (1974), 576–587.

[23] Jacobson M. T., Matthews P., "Generating uniformly distributed random Latin squares", *Journal of Combinatorial Designs*, **4**:6 (1996), 405–437.

On the relationship between cryptographically important properties of finite quasigroups
Zhigliaev R.A.

In this paper we establish a relationship between some properties of finite quasigroups. It is proved that in the case of quasigroups of prime order, all shapeless quasigroups are polynomially complete. Examples are given to show that the converse statement and the generalization to composite orders are not true.

Keywords: finite quasigroup, polynomial completeness, shapeless quasigroup.

References

[1] Markov V. T., Mikhalev A. V., Nechaev A. A., "Nonassociative algebraic structures in cryptography and coding", *Journal of Mathematical Sciences*, **245**:2 (2020), 178–196.

[2] Gligoroski D., "On the S-box in GAGE and InGAGE", 2019, http://gageingage.org/upload/LWC2019NISTWorkshop.pdf.

[3] Gligoroski D., Ødegård R. S., Mihova M., Knapskog S. J., Drápal A., Klima V., Amundsen J., El-Hadedy M., "Cryptographic hash function EDON-R'", *Proceedings of the 1st International Workshop on Security and Communication Networks*, 2009, 1–9.

[4] Markovski S., Mileva A., "NaSHA — family of cryptographic hash functions", *The First SHA-3 Candidate Conference*, 2009.

[5] Shannon C., "Communication theory of secrecy systems", *Bell System Technical Journal*, **28**:4 (1949), 656–715.

[6] Glukhov M. M., "Some applications of quasigroups in cryptography", *Prikl. Diskr. Mat.*, 2008, №2(2), 28–32 (In Russian).

[7] Shcherbacov V. A., "Quasigroups in cryptology", *Computer Science Journal of Moldova*, **17**:2 (2009), 193–228.

[8] Artamonov V. A., Chakrabarti S., Pal S. K., "Characterization of polynomially complete quasigroups based on Latin squares for cryptographic transformations", *Discrete Applied Mathematics*, **200** (2016), 5–17.

[9] Markovski S., "Design of crypto primitives based on quasigroups", *Quasigroups and Related Systems*, **23** (2015), 41–90.

[10] Horváth G., Nehaniv Gh. L., Szabó Cs., "An assertion concerning functionally complete algebras and NP-completeness", *Theoretical Computer Science*, **407** (2008), 591–595.

[11] Larose B., Zadori L., "Taylor terms, constraint satisfaction and the complexity of polynomial equations over finite algebras", *International Journal of Algebra and Computation*, **16** (2006), 563–581.

[12] Yablonskii S. V., "Introduction to discrete mathematics", *Nauka*, 1986 (In Russian).

[13] Chaplygina S. S., Galatenko A. V., "Polynomial completeness and completeness of finite n-quasigroups", *Quasigroups and Related Systems*, **32**:2 (2024), 207–223.

[14] Clifford A., Preston G., "Light's associativity test", *The Algebraic Theory of Semigroups*, **1** (1961), 7–8.

[15] Tarjan R. E., "Determining whether a groupoid is a group", *Information Processing Letters*, **1** (1972), 120–124.

[16] Galatenko A. V., Pankratiev A. E., Staroverov V. M., "An algorithm for checking the existence of subquasigroups", *Chebyshevskii Sbornik*, **22**:2 (2021), 76–89 (In Russian).

[17] Galatenko A. V., Mazurin A. D., Pankratiev A. E., Zhigliaev R. A., "Efficient verification of some properties of finite quasigroups", *Mathematics in Armenia: advances and perspectives*, 2023, 29–30.

[18] Galatenko A. V., Pankratiev A. E., "The complexity of checking the polynomial completeness of finite quasigroups", *Discrete Mathematics and Applications*, **30**:3 (2020), 169–175.

[19] Galatenko A. V., Pankratiev A. E., Staroverov V. M., "Efficient verifiation of polynomial completeness of quasigroups", *Lobachevskii Journal of Mathematics*, 2020, 1444–1453.

[20] Galatenko A. V., Pankratiev A. E., Rodin S. B., "Polynomially Complete Quasigroups of Prime Order", *Algebra and Logic*, **57** (2018), 327–335.

[21] Prasad V. B. V. N., Venkateswara Rao J., "Characterization of Quasigroups and Loop", *International Journal of Scientific and Innovative Mathematical Research*, **1**:2 (2013), 95–102.

[22] Khoroshevskii M. V., "On automorphisms of finite groups", *Mathematics of the USSR-Sbornik*, **22**:4 (1974), 584–594.

[23] Jacobson M. T., Matthews P., "Generating uniformly distributed random Latin squares", *Journal of Combinatorial Designs*, **4**:6 (1996), 405–437.

Часть 3
Математические модели

О функциональной системе, полученной из алгебры множеств добавлением индикаторов мощности

Ю. С. Капустин[1]

В данной работе исследуются свойства функциональной системы C_n с носителем $2^{\mathbb{Z}}$, порождённой теоретико-множественными функциями и индикаторами мощности $\mathbf{card}_0(x) \dots \mathbf{card}_n(x)$.

Ключевые слова: функциональная система, предполный класс, алгебра множеств, критерий полноты.

1. Введение

В работе [1] изучалась алгебраическая система с носителем $\mathbb{Z} \bigcup 2^{\mathbb{Z}}$, образованная отношениями и операциями, выразимыми с помощью логических связок, описателей и кванторов через отношение принадлежности. Было установлено, что любая кванторно определимая функция над множествами в этой системе может быть выражена через обычные теоретико-множественные функции и индикаторы мощности $\mathbf{card}_i(x)$, принимающие значение $\mathbb{Z}$, если множество x содержит ровно i элементов, и значение $\varnothing$ иначе.

Это привлекло внимание к изучению функциональной системы C_n с носителем $2^{\mathbb{Z}}$, порождённой теоретико-множественными функциями и индикаторами мощности $\mathbf{card}_0(x) \dots \mathbf{card}_n(x)$.

Получен ряд важных свойств этой функциональной системы. Найдено число функций в системе, предложен алгоритм решения уравнений в системе. Интересно, что число функций от заданного числа переменных в C_n имеет существенно больший порядок роста, чем для функций конечнозначных логик P_2 и P_k, в связи с чем её изучение не сводится к изучению указанных функциональных систем.

2. Основные понятия

Пусть $\mathbb{Z}$ — множество целых чисел. В качестве универсума возьмём $2^{\mathbb{Z}}$ — множество его подмножеств.

[1] *Капустин Юрий Сергеевич* — аспирант каф. математической теории интеллектуальных систем мех.-мат. ф-та МГУ, e-mail: kapustin.iu@yandex.ru

Kapustin Iurii Sergeevich — graduate student, Lomonosov Moscow State University, Faculty of Mechanics and Mathematics, Chair of Mathematical Theory of Intellectual Systems.

На множестве $2^{\mathbb{Z}}$ естественным образом определены двухместные функции $a \bigcap b, a \bigcup b, a \backslash b$ и нульместная функция-константа $\mathbb{Z}$.

Запись $|x|$ обозначает число элементов в x.

Определим на этом множестве также счётное число функций $\mathbf{card}_k(a)$ (k — целый неотрицательный параметр) следующим образом:

$$\mathbf{card}_k(a) = \begin{cases} \mathbb{Z}, \text{ если } |a| = k \\ \varnothing, \text{ если } |a| \neq k. \end{cases}$$

Обозначим S_n — множество функций

$\{a \bigcap b, a \bigcup b, a \backslash b, \mathbb{Z}, \mathbf{card}_0(a), \ldots, \mathbf{card}_n(a)\}$, определённых на множестве $2^{\mathbb{Z}}$.

Обозначим через C_n функциональную систему с носителем $2^{\mathbb{Z}}$, порождённую функциями из S_n, то есть содержащую все функции, выразимые над S_n при помощи суперпозиции, и только их. Определение суперпозиции можно посмотреть в книге [3].

Обозначим S_C — множество функций $\{a \bigcap b, a \bigcup b, a \backslash b, \mathbb{Z}\}$. Функциональную систему с носителем $2^{\mathbb{Z}}$, порождённую функциями из S_C, обозначим через C. Как будет доказано далее, она изоморфна P_2.

Будем называть два терма равносильными, если они выражают одну и ту же функцию.

Через x^σ будем обозначать терм x, если σ — булева константа 1, и $(\mathbb{Z} \backslash x)$, если σ — булева константа 0. Обозначение $x_1^{\sigma_1} \ldots x_m^{\sigma_m}$ будет использоваться для терма $x_1^{\sigma_1} \bigcap \cdots \bigcap x_m^{\sigma_m}$.

Пусть рассматриваются функции из C_n от m переменных $x_1 \ldots x_m$, где n, m — фиксированные натуральные числа. Атомарным индикатором называется терм вида

$$\mathbf{card}_k(x_1^{\sigma_1} \ldots x_m^{\sigma_m})$$

для $0 \leq k \leq n$ или

$$\mathbb{Z} \backslash (\mathbf{card}_0(x_1^{\sigma_1} \ldots x_m^{\sigma_m}) \bigcup \cdots \bigcup \mathbf{card}_n(x_1^{\sigma_1} \ldots x_m^{\sigma_m})).$$

Для простоты будем обозначать $\mathbb{Z} \backslash (\mathbf{card}_0(x) \bigcup \cdots \bigcup \mathbf{card}_n(x))$ как $\mathbf{card}_{>n}(x)$. Значение этого выражения равно $\mathbb{Z}$, если множество x содержит более n элементов, и $\varnothing$ иначе.

Здесь и далее, когда упоминается функция $\mathbf{card}_l(x)$, где $l > n$, под ней подразумевается функция $\mathbf{card}_{>n}(x) \bigcap \mathbf{card}_{>n}(\mathbb{Z} \backslash x)$.

Например, если рассматриваются функции от переменных x_1, x_2 в C_2, терм

$$\mathbf{card}_1(x_1 \bigcap (\mathbb{Z} \backslash x_2))$$

— атомарный индикатор, а терм

$$\mathbf{card}_0(x_2)$$

— нет, так как не содержит переменной x_1.

Составным индикатором называется терм

$$\bigcap_{\sigma\in\{0,1\}^m} \mathbf{card}_{k_\sigma}(x_1^{\sigma_1}\dots x_m^{\sigma_m}),$$

где $\mathbf{card}_{k_\sigma}$ может означать $\mathbf{card}_0, \mathbf{card}_1, \dots, \mathbf{card}_n$ или $\mathbf{card}_{>n}$.

Например, если рассматриваются функции от переменной x_1 в C_2, терм

$$\mathbf{card}_1(x_1)\bigcap\mathbf{card}_2(\mathbb{Z}\backslash x_1)$$

— составной индикатор, а терм

$$\mathbf{card}_0(x_1)$$

— нет, так как не содержит атомарного индикатора для $\mathbb{Z}\backslash(x_1)$ (то есть для x_1^0).

Далее будет доказано, что если составной индикатор A_i не содержит $\mathbf{card}_{>n}$, то его значение — константа $\varnothing$ (поскольку объединение конечного число конечных множеств не может быть бесконечным множеством $\mathbb{Z}$).

Стандартной формой функции из C от переменных $x_1, \dots, x_m$ назовём терм вида $B_1\bigcup\dots\bigcup B_j$, где каждое B_i имеет вид $x_1^{\sigma_1}\dots x_m^{\sigma_m}$. Эта форма является аналогом ДНФ. Для функции-константы $\varnothing$ стандартной формой назовём терм $\mathbb{Z}\backslash\mathbb{Z}$. В дальнейшем этот терм будет обозначаться как $\varnothing$.

Нормальной формой m-местной функции из C_n называется терм вида $\bigcup_i(A_i\bigcap D_i)$, где A_i — составной индикатор, в терме участвуют все возможные составные индикаторы от m переменных, D_i — терм, выраженный в C. Если все термы D_i являются стандартной формой, назовём такую нормальную форму стандартной нормальной формой.

Пример: Пусть рассматривается функция от одной переменной x в C_0. Тогда терм

$$(\mathbf{card}_0(x)\bigcap\mathbf{card}_0(\mathbb{Z}\backslash(x))\bigcap\mathbb{Z})\bigcup$$
$$(\mathbf{card}_0(x)\bigcap(\mathbb{Z}\backslash(\mathbf{card}_0(\mathbb{Z}\backslash(x))\bigcap x))\bigcup$$
$$((\mathbb{Z}\backslash(\mathbf{card}_0(x)))\bigcap(\mathbb{Z}\backslash(\mathbf{card}_0(\mathbb{Z}\backslash(x))\bigcap\mathbb{Z}))\bigcup$$
$$((\mathbb{Z}\backslash(\mathbf{card}_0(x)))\bigcap\mathbf{card}_0(\mathbb{Z}\backslash(x))\bigcap\varnothing)$$

является нормальной формой, а терм

$$(\mathbf{card}_0(x)\bigcap\mathbf{card}_0(\mathbb{Z}\backslash(x))\bigcap\mathbb{Z})\bigcup$$
$$(\mathbf{card}_0(x)\bigcap(\mathbb{Z}\backslash(\mathbf{card}_0(\mathbb{Z}\backslash(x))\bigcap x))\bigcup$$
$$((\mathbb{Z}\backslash(\mathbf{card}_0(x)))\bigcap(\mathbb{Z}\backslash(\mathbf{card}_0(\mathbb{Z}\backslash(x))\bigcap\mathbb{Z}))$$

— нет, так как содержит не все составные индикаторы.

3. Основные результаты

В данной статье доказываются следующие теоремы:

Теорема 1. *Любую функцию из C_n можно выразить термом в стандартной нормальной форме.*

Теорема 2. *Две стандартные нормальные формы $\bigcup(A_i \bigcap D_i)$ и $\bigcup(A_i \bigcap D'_i)$ задают одну и ту же функцию тогда и только тогда, когда для каждого i (где индекс i параметризует всё множество составных индикаторов) верно одно из следующих утверждений:*

1) A_i не содержит $\mathbf{card}_{>n}$.

2) $D_i \equiv D'_i$

3) Все термы вида $x_1^{\sigma_1} \dots x_m^{\sigma_m}$, где присутствуют все x_m, которые содержатся в только одном из термов D_i и D'_i, присутствуют в D_i в атомарном индикаторе $\mathbf{card}_0(x_1^{\sigma_1} \dots x_m^{\sigma_m})$.

Теорема 3. *Число функций от m переменных $x_1, ..., x_m$ в C_n равно $2^{(n+1)\cdot 2^m \cdot (n+2)^{2^m-1} - n\cdot 2^m \cdot (n+1)^{2^m-1}}$*

Пусть $O_1(x, y_1, \dots y_m), O_2(x, y_1, \dots y_m)$ — термы в C_n. Выражение $O_1(x, \overline{y}) = O_2(x, \overline{y})$ назовём уравнением относительно выбранной переменной x с параметрами $\overline{y}$. Пусть SP — некоторое множество предикатов. Будем говорить, что уравнение $O_1(x, \overline{y}) = O_2(x, \overline{y})$ имеет решение в множестве SP относительно переменной x, если предикат $O_1(x, \overline{y}) = O_2(x, \overline{y})$ выразим некоторой формулой над предикатами из SP. Эту формулу назовём решением уравнения $O_1(x, \overline{y}) = O_2(x, \overline{y})$ в множестве SP относительно переменной x.

Теорема 4. *Любое уравнение в C_n относительно переменной x с параметрами $x_1, \dots x_m$ имеет решение в множестве предикатов вида*

$$\mathbf{card}_{n_j}(x \bigcap F_j(x_1 \dots x_m)) = \mathbb{Z}$$

и

$$\mathbf{card}_{n_j}(F_j(x_1 \dots x_m) \backslash x) = \mathbb{Z},$$

где F_j — функция из C. При этом существует алгоритм, позволяющий найти это решение.

4. Число функций от m переменных в C_n

Чтобы определить число функций от m переменных, найдём стандартую форму, в которой выражается каждая функция из C, и нормальную форму, в которой выражается любая функция из C_n. Также найдём

необходимое и достаточное условие, при котором две стандартные формы задают одну и ту же функцию.

Во-первых, заметим, что проскольку

$$a\backslash b = a\bigcap(\mathbb{Z}\backslash b),$$

функции системы C порождаются также системой функций

$\{a\bigcap b, a\bigcup b, \mathbb{Z}\backslash b, \mathbb{Z}\}$. Рассмотрим формулы алгебры логики над набором функций:

$a\&b$, $a|b$, $\neg b$, 1

Рассмотрим оператор $G(f)$, сопоставляющий по индукции формуле алгебры логики над набором функций $\{a\&b,\ a|b,\ \neg b,\ 1\}$ терм из C_n над системой операций $\{a\bigcap b, a\bigcup b, \mathbb{Z}\backslash b, \mathbb{Z}\}$. Определим его (и обратное отображение) по индукции по длине формулы:

$G(x_i) = x_i, G^{-1}(x_i) = x_i$, если x — переменная.

$G(1) = \mathbb{Z}, G^{-1}(\mathbb{Z}) = 1$.

Если a, b — формулы алгебры логики над $\{a\&b,\ a|b,\ \neg b,\ 1\}$, c, d — термы над $\{a\bigcap b, a\bigcup b, \mathbb{Z}\backslash b, \mathbb{Z}\}$, то

$G(a|b) = G(a)\bigcup G(b), G^{-1}(c\bigcup d) = G^{-1}(c)|G^{-1}(d)$,

$G(a\&b) = G(a)\bigcap G(b), G^{-1}(c\bigcap d) = G^{-1}(c)\&G^{-1}(d)$,

$G(\neg a) = \mathbb{Z}\backslash G(a), G^{-1}(\mathbb{Z}\backslash c) = \neg G^{-1}(c)$.

Лемма 1. *Отображение G множества функций алгебры логики на множество функций в системе C, корректно определено и обратное отображение также корректно определено. То есть если две формулы f_1 и f_2 задают одну и ту же функцию алгебры логики, то термы $G(f_1)$ и $G(f_2)$ задают одну и ту же функцию. И наоборот, если два терма g_1 и g_2 в C_n задают одну и ту же функцию, то термы $G_1^{(-1)}(g_1)$ и $G_1^{(-1)}(g_2)$ задают одну и ту же функцию алгебры логики.*

Доказательство леммы.

Рассмотрим произвольную формулу алгебры логики $f_1(x_1 \dots x_n)$ над $\{\&, |, 1, \}$. Докажем индукцией по длине терма, что $e \in \mathbb{Z}$ принадлежит результату функции, выражаемой термом $G(f_1(x_1 \dots x_n))$ при тех и только тех значениях набора $x_1 \dots x_n$, при которых истинен результат формулы $f_1(y_1 \dots y_n)$, где $y_i = (e \in x_1)$.

База индукции — $(e \in x_i) \iff e \in (x_i)$ — очевидно выполняется.

Шаг индукции. Пусть a, b — два терма над $\{\bigcap, \bigcup, \backslash, \mathbb{Z}\}$. Тогда:

$e \in (a\bigcup b) \iff (e \in a)|(e \in b)$;

$e \in (a\bigcap b) \iff (e \in a)\&(e \in b)$;

$e \in \mathbb{Z} \iff 1$;

$e \in (\mathbb{Z}\backslash b) \iff \neg(e \in b)$ — эти утверждения также выполнены.

Следовательно, e принадлежит результату функции, выражаемой термом $G(f_1(x_1 \dots x_n))$ при тех и только тех значениях набора $x_1 \dots x_n$,

при которых истинен результат формулы $f_1(y_1 \dots y_n)$, где $y_i = (e \in x_1)$. Перейдём к доказательству самой леммы.

$\rightarrow$) Допустим, что две формулы f_1 и f_2 задают одну и ту же функцию. Результат функции, задаваемой термом $G(f_1)$ по доказанному ранее равен множеству тех элементов $e \in \mathbb{Z}$, для которых истинно значение формулы $f_1(y_1 \dots y_n)$, при $y_i = (e \in x_1)$, что полностью определает эту функцию. Результат функции, задаваемой термом $G(f_2)$ равен множеству тех e из $\mathbb{Z}$, для которых истинно значение формулы $f_2(y_1 \dots y_n)$ при $y_i = (e \in x_1)$. Поскольку формулы f_1 и f_2 задают одну и ту же функцию, то множества e из $\mathbb{Z}$, для которых значения $f_2(y_1 \dots y_n)$ и $f_1(y_1 \dots y_n)$ истинны, совпадают. Следовательно, термы $G(f_1)$ и $G(f_2)$ задают одну и ту же функцию.

$\leftarrow$) Допустим, что два терма g_1 и g_2 в C_n задают одну и ту же функцию, e — элемент $\mathbb{Z}$. Для любого набора $(x_1, \dots, x_n) \in \{0,1\}^n$ можно рассмотреть набор $(y_1, \dots, y_n) \in (2^{\mathbb{Z}})^n$:

$$y_i = \begin{cases} \{e\}, \text{если } x_1 = 1 \\ \varnothing, \text{если } x_1 = 0. \end{cases}$$

Тогда $G^{-1}(g_1(x_1 \dots x_n)) = e \in g_1(y_1 \dots y_n) = e \in g_2(y_1 \dots y_n) = G^{-2}(g_1(x_1 \dots x_n))$.

Отсюда из равенства функций, задаваемых термами g_1 и g_2 следует равенство значений функций, задаваемых формулами $G^{-1}(g_1)$ и $G^{-1}(g_2)$, на любом наборе $(x_1, \dots, x_n) \in \{0,1\}^n$. Следовательно, эти функции равны, ч.т.д.

Следствие 1. *В системе C 2^{2^m} различных m-местных функций.*

В моей статье [1] при доказательстве леммы 4 из теорем было доказано следующее утверждение:

Лемма 2. *(О разложении) Для любого терма T от переменных $x_1, \dots x_m$ в C можно найти такой набор $N(T)$ термов вида $x_1^{\sigma_1} \dots x_m^{\sigma_m}$, что при любом значении набора переменных $x_1, \dots, x_m$ значения термов из $N(T)$ — непересекающиеся множества и значение их объединения равно значению T.*

1) Для любых $a_1, \dots a_m \in 2^m athbbZ$ и двух различных термов $T = x_1^{\sigma_1} \dots x_m^{\sigma_m}$ и $T' = x_1^{\sigma'_1} \dots x_m^{\sigma'_m}$ значения этих двух термов - непересекающиеся множества.

Действительно, пусть, без ограничения общности, $\sigma'_i \neq \sigma_i$, $\sigma'_i = 1, \sigma_i = 0$. Тогда $T(a_1, \dots, a_n) = a_1^{\sigma_1} \dots a_m^{\sigma_m} \in \mathbb{Z} \backslash a_i$ $T'(a_1, \dots, a_n) = a_1^{\sigma'_1} \dots a_m^{\sigma'_m} \in a_i$. Так как a_i и $\mathbb{Z} \backslash a_i$ не пересекаются, $T'(a_1, \dots, a_n)$ и $T(a_1, \dots, a_n)$ не пересекаются.

2) Пусть T – терм в над S_C. К нему можно последовательно применить следующие преобразования:

– заменить все подтермы вида $a\backslash b$, где a, b - термы, $a \neq \mathbb{Z}$ на подтермы $a \bigcap (Z\backslash b)$.

– заменить все подтермы вида $Z\backslash(a \bigcup b)$ на $(Z\backslash a) \bigcap (Z\backslash b)$ и все подтермы вида $Z\backslash(a \bigcap b)$ на $(Z\backslash a) \bigcup (Z\backslash b)$. Повторять процедуру, пока не останется операций $Z\backslash$, внешних по отношению к $\bigcap, \bigcup$.

– заменить все подтермы вида $Z \bigcap (a \bigcup b)$ на $(Z \bigcup a) \bigcap (Z \bigcup b)$. Повторять процедуру, пока не останется операций $\bigcap$, внешних по отношению к $\bigcup$. Получим терм вида $B_1 \bigcup B_n$ (или B_1), где B_i имеет вид пересечения термов $x_i^{sigma_{ij}}$ и $Z\backslash Z$.

– Если пересечение B_i содержит $\mathbb{Z}\backslash\mathbb{Z}$, то он равносилен $\varnothing$ и его можно убрать из пересечения. Если при этом B_i единственный, то исходный терм тождественно равен пустому объединению.

– Если пересечение B_i не содержит $x_i^{\sigma_i}$, заменить его на
$(B_i \bigcap x_i) \bigcup (B_i \bigcap (\mathbb{Z}\backslash x_i))$.

В результате получим терм-объединение термов вида $x_1^{\sigma_1} \dots x_m^{\sigma_m}$. Значение его на любом наборе значений переменных равно объединению их значений. Лемма доказана.

Лемма доказана.

Теперь найдём такую форму, что для каждой функции в C_n, найдётся терм в этой форме.

Обозначение x^σ будет использоваться для терма:
x, если $\sigma = 1$;
$(\mathbb{Z}\backslash x$, если $\sigma = 0$.

Обозначение $x_1^{\sigma_1} \dots x_m^{\sigma_m}$ будет использоваться для терма

$$x_1^{\sigma_1} \bigcap x_2^{\sigma_2} \cdots \bigcap x_m^{\sigma_m}$$

.

Лемма 3. *Для каждой функции из C_n от переменных $x_1 \dots x_m$, найдётся выражающий её терм над S_n, в котором операция* **card** *применяется только к термам вида $x_1^{\sigma_1} \dots x_m^{\sigma_m}$.*

Доказательство.

Допустим, некоторая функция выражается термом g в C_n. Покажем, что её можно выразить термом, в котором нет вложенных **card**.

Действительно, если в терм g входит терм $\mathbf{card}_l(g')$, то терм g задаёт ту же функцию, что и терм

$$(\mathbf{card}_l(g') \bigcap g|_{\mathbf{card}_l(g')=\mathbb{Z}}) \bigcup (g|_{\mathbf{card}_l(g')=\varnothing} \backslash \mathbf{card}_l(g')),$$

где через

$$g|_{\mathbf{card}_l(g')=\mathbb{Z}}$$

обозначен терм, получающийся из g путём замены $\mathbf{card}_l(g')$ на $\mathbb{Z}$; константа $\varnothing$ выражается как $\mathbb{Z}\backslash\mathbb{Z}$. Назовём замену этого типа заменой вынесения $\mathbf{card}_l(g')$. Эта замена не меняет выражаемую термом функцию, поскольку для тех значений переменных, для которых значение терма g равно $\mathbb{Z}$, значение обоих термов равно значению терма $g|_{\mathbf{card}_l(g')=\mathbb{Z}}$, а для тех значений переменных, для которых значение терма g равно $\varnothing$, значение обоих термов равно значению терма $g|_{\mathbf{card}_l(g')=\varnothing}$.

Пусть максимальная вложенность операций **card** в терме T равна k, $k > 1$. Тогда если последовательно применить к терму T замену вынесения $\mathbf{card}_l(g')$ для каждого подтерма вида $\mathbf{card}_l(g')$, где g не содержит операций **card**, получим терм с максимальной вложенностью операций **card**, равной $k-1$. Повторяя эту процедуру, получим терм (обозначим его g''), в котором операция **card** будет применяться только к термам над S_C.

Например,

$$\mathbf{card}_0(x_1 \bigcup (\mathbf{card}_1(x_2))) = (\mathbf{card}_1(x_2) \bigcap \mathbf{card}_0(x_1 \bigcup \mathbb{Z})) \bigcup$$

$$(\mathbf{card}_0(x_1 \bigcup \varnothing) \backslash \mathbf{card}_1(x_2)).$$

Пусть терм g'' содержит подтерм $\mathbf{card}_l(g''')$. По лемме о разложении, существует множество $N(g''')$ термов вида $x_1^{\sigma_1} \dots x_m^{\sigma_m}$, значения которых при любом значении переменных — непересекающиеся множества и в объединении дают значение g'''. Пусть $N(g''')$ содержит k термов g_j. Пусть $(l_{i1}, \dots, l_{ik})$ — все возможные наборы целых неотрицательных чисел, сумма которых равна l.

Тогда терм $\mathbf{card}_l(g''')$ равносилен терму

$$\bigcap_i (\mathbf{card}_{l_{i1}}(g_1) \bigcup \dots \bigcup \mathbf{card}_{l_{ik}}(g_k)).$$

Таким образом терм $\mathbf{card}_l(g''')$ равносилен терму, выразимому через термы $\mathbf{card}_{n_i}(g_i)$, где все g_i имеют вид $x_1^{\sigma_1} \dots x_m^{\sigma_m}$ и принадлежат $N(g''')$, а все n_i не больше l.

Например,

$$\mathbf{card}_2(x_2) = (\mathbf{card}_0(x_2 \bigcap x_1) \bigcap \mathbf{card}_2(x_2 \backslash x_1)) \bigcup (\mathbf{card}_1(x_2 \bigcap x_1)$$

$$\bigcap \mathbf{card}_1(x_2 \backslash x_1)) \bigcup (\mathbf{card}_2(x_2 \bigcap x_1) \bigcap \mathbf{card}_0(x_2 \backslash x_1)).$$

Таким образом каждую функцию в C_n можно выразить термом над S_n, в котором операция **card** применяется только к термам вида $x_1^{\sigma_1} \dots x_m^{\sigma_m}$. Лемма доказана.

Чтобы найти точное число функций в C_n от m переменных, найдём стандартную форму для таких термов.

Назовём терм $\mathbf{card}_k(x_1^{\sigma_1} \dots x_m^{\sigma_m})$ или $\mathbb{Z} \backslash (\mathbf{card}_0(x_1^{\sigma_1} \dots x_m^{\sigma_m}) \bigcup \dots \bigcup \mathbf{card}_n(x_1^{\sigma_1} \dots x_m^{\sigma_m}))$ атомарным индикатором, если $x_1 \dots x_m$ — все переменные. Для простоты будем обозначать $\mathbb{Z} \backslash (\mathbf{card}_0(x_1^{\sigma_1} \dots x_m^{\sigma_m}) \bigcup \dots \bigcup \mathbf{card}_n(x_1^{\sigma_1} \dots x_m^{\sigma_m}))$ как $\mathbf{card}_{>n}(x_1^{\sigma_1} \dots x_m^{\sigma_m})$. Поскольку существует 2^m возможных значений для $\sigma_1 \dots \sigma_m$ и $n+2$ различных операций $\mathbf{card_0}, \dots, \mathbf{card_n}, \mathbf{card}_{>\mathbf{n}}$, всего существует $(n+2) \cdot (2^m)$ атомарных индикаторов от m данных переменных в C_n.

Составным индикатором назовём терм $\bigcap_{\sigma \in \{0,1\}^m} \mathbf{card}_{k_\sigma}(x_1^{\sigma_1} \dots x_m^{\sigma_m})$, в котором функцией $\bigcap$ соединены атомарные индикаторы для всех значений параметров $\sigma \in \{0,1\}^m$; $\{0,1\}^m$ — булев куб; $\mathbf{card}_{k_\sigma}$ может означать $\mathbf{card}_0, \dots, \mathbf{card}_n$ или $\mathbf{card}_{>n}$. При этом будем считать равными составные индикаторы,которые различаются лишь порядком множителей во внешней операции пересечения. Всего существует $(n+2)^{(2^m)}$ различных составных индикаторов, поскольку каждый индикатор определяется 2^m параметрами k_σ, каждый из которых может принимать одно из $n+2$ значений — либо число от 0 до n, либо "$> n$".

Также заметим, что атомарные и составные индикаторы могут принимать только значения $\mathbb{Z}$ или $\varnothing$.

Пусть составные индикаторы параметризуются индексом i. Нормальной формой функции из C_n называется терм вида $\bigcup_i (A_i \bigcap D_i)$, где A_i — составной индикатор, соответствующий параметру i, D_i — формула, выраженная в C, в терме используются все возможные составные индикаторы. Если все D_i выражены в стандартной форме, назовём такую нормальную форму стандартной нормальной формой.

Лемма 4. *Любую функцию из C_n можно выразить термом в нормальной форме.*

Доказательство. Для каждого набора значений переменных $x'_1 \dots x'_m$ каждый из термов $x_1^{'\sigma_1} \dots x_m^{'\sigma_m}$ имеет одно значение — множество, имеющее определённое конечное или счётное число элементов $|x_1^{'\sigma_1} \dots x_m^{'\sigma_m}|$. Каждому набору значений переменных $x'_1 \dots x'_m$ таким образом можно сопоставить ровно один составной индикатор $\bigcap_{\sigma \in \{0,1\}^m} \mathbf{card}_{|x_1^{'\sigma_1} \dots x_m^{'\sigma_m}|}(x_1^{\sigma_1} \dots x_m^{\sigma_m})$, значение которого на этом наборе равно $\mathbb{Z}$ (здесь $\mathbf{card}_{|x_1^{'\sigma_1} \dots x_m^{'\sigma_m}|}$ считается принимающим значение или до n или значение $> n$)). Значение остальных составных индикаторов на этом наборе равно $\varnothing$. Следовательно, два различных составных индикатора

не могут принимать значение, не равное $\varnothing$, на одном наборе значений переменных.

Пример. Набору значений переменных $x'_1 = \{1\}, x'_2 = \{1,2\}$ в C_1 соответствует составной индикатор

$$\mathbf{card}_0(x_1 \bigcap (\mathbb{Z} \backslash x_2)) \bigcap \mathbf{card}_1(x_1 \bigcap x_2) \bigcap \mathbf{card}_{>1}((\mathbb{Z} \backslash x_1)$$

$$\bigcap (\mathbb{Z} \backslash x_2)) \bigcap \mathbf{card}_1((\mathbb{Z} \backslash x_1) \bigcap x_2)$$

Составной индикатор — пересечение атомарных индикаторов, которые (поскольку для них **card** — внешняя функция) могут принимать только значения $\mathbb{Z}$ или $\varnothing$. Если атомарный индикатор $\mathbf{card}_l(x_1^{\sigma_1} \dots x_m^{\sigma_m})$ входит в составной индикатор I, то на наборе значений переменных $(x'_1, \dots, x'_m)$, на котором составной индикатор принимает значение $\mathbb{Z}$, атомарный индикатор принимает значение $\mathbb{Z}$.

Если же этот атомарный индикатор не входит в I, то в него входит другой индикатор $\mathbf{card}_{l_i}(x_1^{\sigma_1} \dots x_m^{\sigma_m})$, $l_i \neq l$. Поскольку $x_1^{\sigma_1} \dots x_m^{\sigma_m}$ не может иметь различное число элементов при одном и том же значении $x_1 \dots x_m$, то на наборе значений переменных, на котором составной индикатор принимает значение $\mathbb{Z}$, атомарный индикатор значение $\varnothing$.

Следовательно, для каждого составного индикатора и каждого атомарного индикатора на всех наборах значений переменных, на которых составной индикатор принимает значение $\mathbb{Z}$, атомарный индикатор принимает одно и то же значение. Это значение — $\mathbb{Z}$, если атомарный индикатор входит в составной, и $\varnothing$ иначе.

Рассмотрим функцию $O(x_1, \dots, x_m)$. По предыдущей лемме без ограничения общности можно считать, что она выражена термом $O'(x_1, \dots, x_m)$, в котором под **card** находятся только термы $x_1^{\sigma_1} \dots x_m^{\sigma_m}$. То есть все вхождения **card** в этот терм — атомарные индикаторы. Для каждого составного индикатора A_i обозначим за D_i терм, который получается из $O'(x_1, \dots, x_m)$ путём замены содержащихся в A_i атомарных индикаторов на $\mathbb{Z}$, а не содержащихся — на $\varnothing$. В этом случае каждое D_i выражено только через функции из S_C, и $\bigcup(A_i \bigcap D_i)$ — нормальная форма. Покажем, что $\bigcup(A_i \bigcap D_i)$ — нормальная форма для O, то есть выражает O.

Действительно, рассмотрим набор значений N переменых $x_1, \dots, x_m$. Пусть A_k — тот единственный составной индикатор, который на данном наборе принимает значение $\mathbb{Z}$. Тогда значение $(A_i \bigcap D_i)(N)$ равно $\varnothing$ при i не равном k, а значение $\bigcup(A_k \bigcap D_k)$ равно $D_k(N)$. Из определения D_k, $D_k(N) = O(N)$. Следовательно, на любом наборе N $O(N) = \bigcup(A_i \bigcap D_i)(N)$. То есть нормальная форма $\bigcup(A_i \bigcap D_i)$ задаёт функцию O, ч.т.д.

Будем называть две нормальные формы $\bigcup(A_i \bigcap D_i)$ и $\bigcup(A_i \bigcap D'_i)$ равными, если термы D_i и D'_i выражают одну и ту же функцию для каждого i. В противном случае будем называть их различными. Стандартной нормальной формой назовём такую форму, где каждое D_i выражено в виде, аналогичном ДНФ, то есть в виде $\bigcup(x_1^{\sigma_1}, \ldots, x_1^{\sigma_n})$. Несложно убедиться, что для любой нормальной формы существует равная ей стандартная нормальная форма.

Заметим, что две различные нормальные формы могут задавать одну и ту же функцию. Например, в C_0

$$\bigcup_i (A_i \bigcup \varnothing)$$

,

$$(\mathbf{card}_0(x) \bigcap \mathbf{card}_{>0}(\mathbb{Z} \backslash x) \bigcap x) \bigcup (\bigcup_j (A_j \bigcup \varnothing))$$

и

$$(\mathbf{card}_0(x) \bigcap \mathbf{card}_0(\mathbb{Z} \backslash x) \bigcap \mathbb{Z}) \bigcup (\bigcup_k (A_k \bigcup \varnothing))$$

задают одну и ту же функцию. Но можно показать, что подобные пары форм — единственные различные формы, выражающие одну и ту же функцию.

Лемма 5. *$\bigcup_{(\sigma_1 \ldots \sigma_m) \in \{0,1\}^m} (x_1^{\sigma_1} \ldots x_m^{\sigma_m}) = \mathbb{Z}$ для любого значения переменных $x_1 \ldots x_m$*

Докажем индукцией по числу m переменных. Если $m = 1$, $(\mathbb{Z} \backslash x_1) \bigcup x_1 = \mathbb{Z}$ — утверждение леммы верно.

Если утверждение леммы верно для m, то для $m' = m + 1$
$\bigcup_{(\sigma_1 \ldots \sigma_{m'}) \in \{0,1\}^{m'}} (x_1^{\sigma_1} \ldots x_{m'}^{\sigma_{m'}}) = \mathbb{Z} =$
$((\bigcup_{(\sigma_1 \ldots \sigma_m) \in \{0,1\}^m} (x_1^{\sigma_1} \ldots x_m^{\sigma_m})) \bigcap (x_{m'})) \bigcup$
$((\bigcup_{(\sigma_1 \ldots \sigma_m) \in \{0,1\}^m} (x_1^{\sigma_1} \ldots x_m^{\sigma_m})) \bigcap (\mathbb{Z} \backslash x_{m'})) =$
$((\bigcup_{(\sigma_1 \ldots \sigma_m) \in \{0,1\}^m} (x_1^{\sigma_1} \ldots x_m^{\sigma_m})) \bigcap (x_{m'} \bigcup (\mathbb{Z} \backslash x_{m'})) =$
$= ((\bigcup_{(\sigma_1 \ldots \sigma_m) \in \{0,1\}^m} (x_1^{\sigma_1} \ldots x_m^{\sigma_m})) = \mathbb{Z}$. Лемма доказана.

Лемма 6. *Если составной индикатор A_i не содержит $\mathbf{card}_{>n}$, то $A_i \equiv \varnothing$*

Доказательство. Поскольку по предыдущей лемме объединение конечного числа множеств $x_1^{\sigma_1} \ldots x_m^{\sigma_m}$ равно бесконечному множеству $\mathbb{Z}$. Следовательно, хотя бы одно из них — пусть это будет $x_1^{\sigma'_1} \ldots x_m^{\sigma'_m}$ бесконечно. Тогда соответствующий атомарный индикатор $\mathbf{card}_l(x_1^{\sigma'_1} \ldots x_m^{\sigma'_m})$ принимает значение $\varnothing$, и весь составной индикатор A_i принимает значение $\varnothing$.

Лемма 7. *Две стандартные нормальные формы* $\bigcup(A_i \bigcap D_i)$ *и* $\bigcup(A_i \bigcap D'_i)$ *задают одну и ту же функцию тогда и только тогда, когда для каждого* i *(где индекс* i *параметризует всё множество составных индикаторов) верно одно из следующих утверждений:*

1) A_i *не содержит* $\mathbf{card}_{>n}$.

2) $D_i \equiv D'_i$

3) Все термы вида $x_1^{\sigma_1} \dots x_m^{\sigma_m}$*, где присутствуют все* x_m*, которые содержатся в только одном из термов* D_i *и* D'_i*, присутствуют в* D_i *в атомарном индикаторе* $\mathbf{card}_0(x_1^{\sigma_1} \dots x_m^{\sigma_m})$.

Доказательство. $\leftarrow$) Пусть две формы $\bigcup(A_i \bigcap D_i)$ и $\bigcup(A_i \bigcap D'_i)$ таковы, что для каждого i одно из утверждений (1) – (3) верно. Рассмотрим набор значений переменных N и соответствующий ему составной индикатор A_i, который принимает на нём значение $\mathbb{Z}$.

Для этого набора согласно предыдущей лемме не может выполняться 1).

Если для него выполняется 2), то, поскольку $D_i \equiv D'_i$, верно равенство $A_i(N) \bigcap D_i(N) = A_i(N) \bigcap D'_i(N)$.

Если для него выполняется 3), то $(A_i \bigcap D_i)(N) \equiv (A_i \bigcap D'_i)(N)$, поскольку обе части являются объединением одних и тех же непустых множеств и некоторого количества пустых.

Следовательно, $\bigcup(A_i \bigcap D_i)$ и $\bigcup(A_i \bigcap D'_i)$ принимают одно и то же значение на любом значении переменных N, и выражаемые этими термами функции совпадают.

$\rightarrow$) От противного. Пусть две стандартные нормальные формы $\bigcup(A_i \bigcap D_i)$ и $\bigcup(A_i \bigcap D'_i)$ задают одну и ту же функцию. Пусть существует i, для которого A_i содержит $\mathbf{card}_{>n}$, и, без ограничения общности, в терме D_i содержится терм $x_1^{\sigma_1} \dots x_m^{\sigma_m}$, который не содержится в D'_i и присутствует в A_i в атомарном индикаторе $\mathbf{card}_k(x_1^{\sigma_1} \dots x_m^{\sigma_m})$, где k не равно 0.

Пусть $x_1 \dots x_m$ — набор значений переменных, соответствующий A_i (то есть тот, на котором $A_i = \mathbb{Z}$). Такой набор существует, поскольку можно найти 2^m непересекающихся множеств с заданным числом элементов у каждого (хотя бы одно из которых бесконечно), объединение которых равно $\mathbf{Z}$. $(A_i \bigcap D_i)(N)$ содержит элемент из множества $x_1^{\sigma_1} \dots x_m^{\sigma_m}$, $(A_i \bigcap D'_i)(N)$ не содержит элемент из $x_1^{\sigma_1} \dots x_m^{\sigma_m}$. При этом никакие другие A_i не содержат элемент из $x_1^{\sigma_1} \dots x_m^{\sigma_m}$. Следовательно, $\bigcup(A_i \bigcap D_i)$ и $\bigcup(A_i \bigcap D'_i)$ задают разные функции. Лемма доказана.

С учётом этой леммы найдём число функций в C_n от m переменных.

Теорема 1. *Число функций от* m *переменных* $x_1, ..., x_m$ *в* C_n *равно* $2^{(n+1)\cdot 2^m \cdot (n+2)^{2^m-1} - n \cdot 2^m \cdot (n+1)^{2^m-1}}$

Доказательство.

Как было указано ранее, всего существует $(n+2)^{(2^m)}$ различных составных индикаторов.

Зафиксируем A_i и найдём количество функций типа $(A_i \bigcap D)$. Обозначим его $F(i)$. Если A_i не содержит $\mathbf{card}_{>n}$, то значение A_i всегда равно $\varnothing$, $(A_i \bigcap D)$ — константа $\varnothing$, $F(i) = 1$. Если A_i содержит $\mathbf{card}_{>n}$, и j — число $\mathbf{card}_0$ в A_i, то $F(i) = 2^{k-j}$ (поскольку наличие или отсутствие в D подтерма $x_1^{\sigma_1} \dots x_m^{\sigma_m}$, для которого A_i имеет подтерм $\mathbf{card}_0(x_1^{\sigma_1} \dots x_m^{\sigma_m})$, не влияет на значение функции), и $k - j = log_2(F(i))$.

Количество же всех m-местных функций $N(n,m)$ равно $\Pi_i F(i) = 2^{\Sigma_i log_2(F(i))}$. (*)

Обозначим $l = n + 2$ — число различных индексов для $\mathbf{card}$, включая $> n$.

Поскольку в составном индикаторе A_i при фиксированных j и k есть:

— C_k^j различных вариантов, где расположены j различных нулевых $\mathbf{card}$,

— $(l-1)^{k-j}$ варианта для значения ненулевых параметров $\mathbf{card}$,

— $(l-2)^{k-j}$ варианта для значения ненулевых параметров $\mathbf{card}$, в которых нет $\mathbf{card}_{>n}$,

— $(l-1)^{k-j} - (l-2)^{k-j}$ варианта для значения ненулевых параметров $\mathbf{card}$, среди которых есть хотя бы одно $\mathbf{card}_{>n}$,

получим:

$N(n,m) = 2^{\Sigma_i log_2(F(i))} = 2^{\Sigma_{j=0}^k C_k^j \cdot ((l-1)^{k-j} - (l-2)^{k-j}) \cdot (k-j)} =$

$2^{\Sigma_{j=0}^k C_k^{k-j} \cdot ((l-1)^{k-j} - (l-2)^{k-j}) \cdot (k-j)} = 2^{\Sigma_{j'=0}^k C_k^{j'} \cdot ((l-1)^{j'} - (l-2)^{j'}) \cdot (j')}$ (*)

Чтобы найти эту сумму, найдём значение суммы $\Sigma_{i=0}^n (C_n^i \cdot x^i \cdot i)$ для произвольного натурального i и вещественного x. Для этого воспользуемся фактом из математического анализа, что производная суммы дифференцируемых функций равна сумме их производных:

$\Sigma_{i=0}^n (C_n^i \cdot x^i \cdot i) = x \cdot \Sigma_{i=0}^n (C_n^i \cdot x^{i-1} \cdot i) = x \cdot \Sigma_{i=0}^n (C_n^i \cdot (x^i)'_x) = x \cdot (\Sigma_{i=0}^n C_n^i \cdot (x^i))'_x = x \cdot ((x+1)^n)'_x = x \cdot n \cdot (x+1)^{n-1}$ (**)

Таким образом, значение выражения (*) равно

$2^{(l-1)*k*l^{k-1} - (l-2)*k*(l-1)^{k-1}} = 2^{(n+1)\cdot 2^m \cdot (n+2)^{2^m-1} - n \cdot 2^m \cdot (n+1)^{2^m-1}}$

Теорема доказана.

5. Уравнения в C_n

Пусть $O_1(x, y_1, \dots y_m), O_2(x, y_1, \dots y_m)$ — термы в C_n. Выражение $O_1(x, \overline{y}) = O_2(x, \overline{y})$ назовём уравнением относительно выбранной переменной x с параметрами $\overline{y}$. Пусть SP — некоторое множество предикатов. Будем говорить, что уравнение $O_1(x, \overline{y}) = O_2(x, \overline{y})$ имеет решение в множестве SP относительно переменной x, если предикат $O_1(x, \overline{y}) = O_2(x, \overline{y})$

выразим некоторой формулой над предикатами из SP. Эту формулу назовём решением уравнения $O_1(x, \overline{y}) = O_2(x, \overline{y})$ в множестве SP относительно переменной x.

Теорема 2. *Любое уравнение в C_n относительно переменной x с параметрами $x_1, \dots x_m$ имеет решение в множестве предикатов вида*

$$\mathbf{card}_{n_j}(x \bigcap F_j(x_1 \dots x_m)) = \mathbb{Z}$$

и

$$\mathbf{card}_{n_j}(F_j(x_1 \dots x_m) \backslash x) = \mathbb{Z},$$

где F_j — функция из C. При этом существует алгоритм, позволяющий найти это решение.

Доказательство. Сначала докажем лемму

Лемма 8. *Существует алгоритм, с помощью которого любой терм $T(x, x_1, \dots, x_n)$ можно привести к стандартной форме (то есть найти терм в стандартной форме, выражающий ту же функцию).*

Один из возможных алгоритмов выглядит следующим образом:

1) Рассмотреть все составные индикаторы A_i от переменных $x, x_1, \dots, x_n$. Записать терм $\bigcup_i (A_i \bigcap T)$.

2) Преобразовать каждый терм $A_i \bigcap T_i$ следующим образом (терм T_i может меняться между шагами алгоритма):

— Пока в рассматриваемый терм T_i входит **card**:

—— Найти в нём вхождение вида $\mathbf{card}_k(T')$, где в T' не входит никакой другой **card** (то есть T' выражается над $\{\mathbb{Z} \backslash, \bigcap, \bigcup\}$)

—— Найти, объединением каких пересечений вида $x_1^{\sigma_1} \bigcap \dots \bigcap x_m^{\sigma_m}$ является T' (из изоморфизма C и P_2 это делается аналогично приведению формулы из P_2 к СКНФ), просуммировать по j индексы k_{ij} из входящих в A_i термов $\mathbf{card}_{k_{ij}}(a_i)$.

—— Если результат равен индексу k (или $> n$, если k — индекс "$> n$"), заменить в рассматриваемом терме $\mathbf{card}_k(T')$ на $\mathbb{Z}$. Иначе заменить его на $\varnothing$.

В результате получим равносильный T терм $\bigcup_i (A_i \bigcap T_i')$ в нормальной форме.

3) Аналогично алгоритму приведения функции алгебры логики к СКНФ, привести T_i к виду, аналогичному СКНФ.

В результате получится терм, равносильный T и имеющий стандартную нормальную форму, ч.т.д. Лемма доказана.

Как следует из леммы, без ограничения общности можно считать, что в выражении $O_1(x, x_1, \dots x_m) = O_2(x, x_1, \dots x_m)$ оба терма O_1 и O_2

записаны в стандартной форме. То есть достаточно решать уравнения вида $\bigcup(A_i \bigcap D_i) = \bigcup(A_i \bigcap D'_i)$.

Как было показано ранее, чтобы набор $x_1, \ldots, x_n$, на котором выполнено $A_j(x_1, \ldots, x_n) = \mathbb{Z}$, удовлетворял равенству $\bigcup(A_i \bigcap D_i) = \bigcup(A_i \bigcap D'_i)$, необходимо и достаточно, чтобы любой атомарный терм $x_1^{\sigma_1}, \ldots, x_n^{\sigma_n}$, который входит в D_i, но не в D'_i, или наоборот, входил в A_i в виде $\mathbf{card}_0(x_1^{\sigma_1}, \ldots, x_n^{\sigma_n})$. Рассмотрим B — множество всех A_i, для которых любой атомарный терм $x_1^{\sigma_1}, \ldots, x_n^{\sigma_n}$, который входит в D_i, но не в D'_i, или наоборот, входит в A_i в виде $\mathbf{card}_0(x_1^{\sigma_1}, \ldots, x_n^{\sigma_n})$.

Если на наборе $(x'_1, \ldots, x'_n)$ принимает значение $\mathbb{Z}$ такой A_i, то $\bigcup(A_j \bigcap D_j)(x'_1, \ldots, x'_n) = \varnothing \bigcup \varnothing \cdots \bigcup \varnothing \bigcup (A_i \bigcap D_i)(x'_1, \ldots, x'_n)) = D_i(x'_1, \ldots, x'_n) = D'_i(x'_1, \ldots, x'_n) = \bigcup(A'_j \bigcap D'_j)(x'_1, \ldots, x'_n)$.

Если же на наборе $(x'_1, \ldots, x'_n)$ принимает значение $\mathbb{Z}$ A_i, не удовлетворяющий этому свойству, то $\bigcup(A_j \bigcap D_j)(x'_1, \ldots, x'_n) = \varnothing \bigcup \varnothing \cdots \bigcup \varnothing \bigcup (A_i \bigcap D_i)(x'_1, \ldots, x'_n)) = D_i(x'_1, \ldots, x'_n) \neq D'_i(x'_1, \ldots, x'_n) = \bigcup(A'_j \bigcap D'_j)(x'_1, \ldots, x'_n)$.

Следовательно, равенство истинно на наборе $(x'_1, \ldots, x'_n)$ если и только если $(x'_1, \ldots, x'_n) \in A_i$ и $A_i \in B$. Решение равносильно формуле $\bigvee_{A_i \in B}(A_i = \mathbb{Z})$.

Заменив $\bigvee(\bigcap(I_{ij}) = \mathbb{Z})$ на $\bigvee(\&(I_{ij} = \mathbb{Z}))$, где I_{ij} — атомарные индинаторы, получим решение уравнения.

6. Заключение

В следующих статьях будут описаны свойства функциональной системы C_n, представлена шефферова функции в ней. Будет представлена серия предполных классов, позволяющая получить критерий относительной полноты.

Автор выражает благодарность профессору А.С. Подколзину за постановку задачи и помощь в работе.

Список литературы

[1] Капустин Ю. С., “Об элементарной выразимости в логике предикатов”, *Интеллектуальные системы. Теория и приложения*, **23**:2 (2019), 135–158.

[2] Яблонский С.В., *Введение в дискретную математику*, «Высшая школа», М, 2003, 384 с.

[3] Яблонский С.В, Грврилов Г.П., Кудрявцев В. Б., *Функции алгебры логики и классы Поста*, «Наука», М, 1966, 120 с.

On algebraic system created from set algebra by adding the set power indicator

Kapustin I.S.

This paper concerns the properties of the functional system C_n. This system has the domain $2^{\mathbb{Z}}$, and is generated by functions $2^{\mathbb{Z}} \backslash x, x \bigcup y, x \bigcap x$ and power indicators $\mathbf{card}_0(x) \dots \mathbf{card}_n(x)$.

Keywords: functional system, precomplete class, set algebra, completion criteria.

References

[1] Kapustin I. S., "On the elementary expressibility in predicate logic", *Intelligent Systems. Theory and Applications*, **23**:2 (2019), 135–158

[2] Yablonsky S.V., *Introduction to discrete mathematics*, «Vysshaya shkola», M, 2003, 384 c.

[3] Yablonsky S.V., Gavrilov G.P., Kudryavtsev V. B., *Functions of the Algebra of Logic and the Post Classes*, «Nauka», M, 1966, 120 c.

Классы двунаправленного движения на луче, реализуемые автоматами с 4 состояниями

Е. В. Кузнецова[1]

В работе [1] показано, что существует универсальный экран с 5 состояниями для класса всех законов движения со скоростью движения вперёд не более, чем 1/2, при этом не существует универсального экрана с 4 состояниями для этого класса законов движения. В данной работе приведены 3 класса закоконов двунаправленного движения на луче, которые можно реализовать клеточным автоматом с 4 состояниями.

Ключевые слова: клеточный автомат, число состояний, бесконечный экран, двунаправленное движение, конструирование изображений.

1. Введение

В работе исследуется конструирование движущихся изображений клеточными автоматами. Рассматривается конечный автомат и бесконечная справа полоса высотой в одну клетку. К каждой клетке полосы прикреплен свой экземпляр конечного автомата. В дальнейшем состояние автомата, прикрепленного к клетке, будем называть состоянием клетки. Множество автоматов, прикреплённых к клеткам полубесконечной прямой, является рассматриваемым в данной работе клеточным автоматом.

Состояние автомата, прикрепленного к клетке, зависит от состояния этого автомата и двух его входов (левого и правого) в предыдущий момент времени. Под входами будем понимать состояния автоматов, прикрепленных к соседним клеткам (имеются в виду две клетки: ближайшая слева и ближайшая справа). Состоянием покоя считается нулевое значение состояния автомата, и автомат в состоянии покоя остаётся таковым, если его соседи тоже находятся в состоянии покоя.

Левый вход самой левой клетки полубесконечной полосы будем называть управляющим входом и будем подавать на него произвольные, но определённые управляющие сигналы.

[1] *Кузнецова Екатерина Викторовна* — м.н.с. каф. математической теории интеллектуальных систем мех.-мат. ф-та МГУ, e-mail: kuz.net.sova@mail.ru.

Kuznetsova Ekaterina Viktorovna — junior researcher of Lomonosov Moscow State University, Faculty of Mechanics and Mathematics, Chair of Mathematical Theory of Intellectual Systems.

Все клетки полубесконечной полосы будем называть экраном. Конфигурацию из состояний клеточных автоматов в данный момент времени будем называть изображением на экране.

Отметим, что состояния клетки интерпретируются не одинаково. Так некоторое заранее фиксированное подмножество состояний клетки, называемых метками, интерпретируются как клетки чёрного цвета, а все остальные состояния, включая состояние покоя, интерпретируются как клетки белого цвета. В результате на экране получается чёрно-белое изображение.

В работе накладывается ограничение на возможные изображения. Так, двух меток на экране быть не может, т.е. две разные клетки не могут принимать состояния, содержащиеся в подмножестве состояний, считающихся метками. Единственная метка на экране интерпретируется нами как точка.

Законом движения точки на экране назовем последовательность, состоящую из символов f, s, b ($f - forward, s - stop, b - back$), кодирующих перемещение точки по экрану в каждый момент времени. Так, если в момент времени t точка сместилась на одну клетку вправо, то t-ый член последовательности закона движения примет значение f, если сместилась влево, то t-ый член последовательности примет значение b, если никуда не сместилась, то t-ый член последовательности примет значение s. Здесь время t отсчитывается от того момента, когда в самой левой клетке появляется метка.

Тема конструирования стационарных изображений клеточными автоматами рассматривалась Е. Е. Титовой в ее работах [2, 3, 4]. Так, в работе [2] исследовалась задача конструирования изображений клеточными автоматами на прямоугольном экране. В работе было показано, что для конструирования любого изображения необходимо и достаточно, чтобы клеточный автомат имел 3 состояния.

В работе [3] продолжалось рассмотрение конструирования изображений клеточным автоматом на прямоугольном экране. В работе были даны оценки времени формирования клеточным автоматом изображений для разного числа состояний данного автомата.

В работе [4] было подробно рассмотрено исследование движения точки на бесконечном экране. В работе описан алгоритм реализации на экране широкого класса законов движения и исследована мера Бернулли множества реализуемых законов движения. Показано, что почти все законы движения являются реализуемыми. Также показано, что относительно Тихоновской топологии множество реализуемых законов движения относится к первой категории Бэра, т.е. очень мало.

В работах Э. Э. Гасанова [5, 6, 7] вводится модель клеточного автомата с локаторами. Он получается добавлением к клеточным автомату

новой возможности — посылать сигналы в “эфир” и получать из “эфира” суммарный сигнал всех элементарных автоматов. Приводится решение некоторых классических и новых задач с помощью стандартных клеточных автоматов, а затем показывается, что эти же задачи с помощью клеточных автоматов с локаторами решаются значительно легче. В частности, описан клеточный автомат с локаторами, который решает задачу однонаправленного движения точки на луче. С помощью клеточных автоматов с локаторами можно решать и другие задачи. Например, в работе [8] показано, что с помощью двумерных клеточных автоматов с локаторами можно решить задачу умножения и деления n-разрядных чисел за время порядка n.

В работе [9] Е. Е. Титовой был рассмотрен класс законов движения, в которых нет двух символов f подряд, а также нет движения назад (то есть нет символов b), такие классы будем называть классами законов движения со скоростью движения вперёд на более, чем $1/2$. Был получен результат, что такие классы движения невозможно реализовать с помощью клеточного автомата с тремя состояниями, но можно реализовать с помощью клеточного автомата с четырьмя состояниями.

В работе [1] рассматривался определенный класс S законов движения, в которых нет двух символов f подряд, движение назад возможно. Основной целью работы являлось определение наименьшего числа состояний клеточных автоматов, при котором можно реализовать все законы движения из класса S, при этом выбор управляющих сигналов через управляющий вход экрана зависел от закона движения. В работе было показано, что минимальное количество состояний клеточного автомата, при котором можно реализовать любое движение из данного класса, равно пяти.

Однако некоторые законы движения из класса S с помощью автомата с четырьмя состояниями реализовать возможно. В данной работе рассматриваются подклассы класса S: законы движения со скоростью движения вперёд $1/4$ и скоростью движения назад $1/2$, законы движения со скоростью движения вперёд $1/4$ и скоростью движения назад 1, а также законы движения со скоростью движения вперёд $1/3$, скоростью движения назад 1 и чётным количеством остановок. Показано, что эти подклассы можно реализовать с четырьмя состояниями. Насколько полно это множество классов, которые возможно реализовать с помощью клеточного автомата с четырьмя состояниями, является темой дальнейших исследований.

Автор выражает благодарность профессору Э.Э. Гасанову за научное руководство и постановку задачи.

2. Основные понятия и формулировка результата

Определим основные понятия, используемые в данной работе.

Пусть S — множество конечных и бесконечных последовательностей, состоящих из элементов $\alpha_n \in \{sf, s, b\}$, в префиксе любой длины которых количество символов b не превышает количества символов f, это правило также верно для всех подмножеств S, рассматриваемых в данной работе. Элементы множества S будем называть *законами движения*. Символ f подразумевает движение на одну клетку вправо, s — остаться на месте, b — на одну клетку влево.

Экраном будем называть следующую конструкцию.

Пусть имеется бесконечная в правую сторону полоса высотой в одну клетку. В каждую клетку полосы поместим по одному экземпляру одного и того же конечного автомата. К входам этого автомата присоединим выходы автоматов, стоящих в двух соседних с ним клетках, то есть у автомата имеется *левый* вход, *правый* вход и текущее состояние автомата. Выходом автомата в заданный момент времени является его состояние в этот момент времени. Для автомата, стоящего в самой левой клетке полосы левый вход не определён. Будем называть его *управляющим входом* и подавать на него управляющие сигналы.

Метками будем называть значения состояний клеточного автомата, при которых считается, что клетка, находящаяся в данном состоянии, видима (чёрная).

Будем говорить, что *на экране реализуется движение по закону* $A \in S$, если выполняются следующие условия:

1) в некоторый момент времени в самой левой клетке экрана появляется метка (до этого на экране нет меток) — этот момент будем называть *моментом начала движения* или *началом движения*;

2) изменение позиции метки на экране в i-й момент от начала движения соответствует i-й букве в слове или сверхслове A, а именно, если $A(i) = s$, то в $(i+1)$-й момент метка остается в той же клетке, где была в текущий момент, если $A(i) = f$, то в $(i+1)$-й момент метка сдвинется на одну ячейку вправо, если $A(i) = b$, то в $(i+1)$-й момент метка сдвинется на одну клетку влево, по сравнению со своим текущим положением;

3) в каждый момент времени после начала движения на экране есть ровно одна метка.

Экран будем называть *универсальным* для множества законов движения S, если для любого закона движения из S существует такая последовательность управляющих сигналов, что на экране формируется

такое изображение, что метка движется по закону S. Через $Q(S)$ обозначим минимальное число состояний, достаточное, чтобы реализовать универсальный экран для множества S.

Изначально в данном клеточном автомате одни нули. Затем управляющее устройство начинает подавать ему на вход управляющие сигналы (управляющую последовательность). В какой-то момент в самой левой клетке экрана появится метка, которая интерпретируется как точка, движение которой мы и изучаем.

Таким образом, под появлением точки на экране будем подразумевать переключение клетки автомата, соответствующей самой левой клетке экрана, в состояние, соответствующее состоянию метки.

После того, как точка (метка) появилась на экране, то она никуда не исчезает и двух точек (меток) на экране быть не может (поэтому, если метка движется, например, вправо, то клетка, в которой была метка, должна перейти в состояние, не соответствующее метке).

В данной статье рассматривается вопрос о том, какие законы движения из S возможно реализовать с 4 состояниями клеточного автомата. Получены следующие результаты.

Теорема 1. *Пусть S^1 — множество законов движения, состоящих из элементов множества $\{ssfs, s, sb\}$. Тогда $Q\left(S^1\right) = 4$.*

Теорема 2. *Пусть S^2 — множество законов движения, состоящих из элементов множества $\{sfss, s, b\}$ и начинающихся с элемента s. Тогда $Q\left(S^2\right) = 4$.*

Теорема 3. *Пусть S^3 — множество законов движения, состоящих из элементов множества $\{ssf, ss, b\}$. Тогда $Q\left(S^3\right) = 4$.*

3. Вспомогательные определения

Перед доказательством основных утверждений введем ряд вспомогательных определений и обозначений.

Пусть $M = \{f, s, b\}$ — алфавит базовых букв движения. Если a — слово в алфавите M, то через $|a|$ обозначим число букв в слове a.

Обозначим $FW = \{ssfs, sfss, ssf\}$ — множество слов движения вперёд, $BC = \{sb, b\}$ — множество слов движения назад, $ST = \{s, ss\}$ — множество слов остановки на месте.

Пусть $\overline{\alpha_n} = \alpha_1\alpha_2\ldots\alpha_n \in S$ — закон движения, где $\alpha_i \in Z = \{fw, st, bc\}$, $fw \in FW, st \in ST, bc \in BC$, $i = 1, 2, \ldots, n$. Из каждого множества слов FW, ST, BC выбираем по одному, всего получаем три слова, образующих множество Z, из них и будет состоять закон движения.

Пусть $a \in FW \cup BC \cup ST$. Определим функцию, идентифицирующую слово в i-ой позиции закона движения:

$$I(\alpha_i = a) = \begin{cases} 1 & \text{если } \alpha_i = a, \\ 0 & \text{если } \alpha_i \neq a. \end{cases}$$

Теперь используем её для подсчёта количества слов $\alpha_i \in Z = \{fw, st, bc\}$ в префиксе закона движения длины n (подразумевается, что в префиксе n слов, а не букв):

$$\begin{aligned} I_{FW}\left(\overline{\alpha_n}\right) &= \sum_{i=1}^{n} I\left(\alpha_i = fw\right), \\ I_{BC}\left(\overline{\alpha_n}\right) &= \sum_{i=1}^{n} I\left(\alpha_i = bc\right), \\ I_{ST}\left(\overline{\alpha_n}\right) &= \sum_{i=1}^{n} I\left(\alpha_i = st\right). \end{aligned}$$

Отсюда можем определить позицию метки на экране

$$d(\overline{\alpha_n}) = I_{FW}\left(\overline{\alpha_n}\right) - I_{BC}\left(\overline{\alpha_n}\right) \geq 0$$

и количество тактов с начала движения

$$t(\overline{\alpha_n}) = I_{ST}\left(\overline{\alpha_n}\right) \cdot |st| + I_{BC}\left(\overline{\alpha_n}\right) \cdot |bc| + I_{FW}\left(\overline{\alpha_n}\right) \cdot |fw|$$

для префикса закона движения длины n. Здесь считается, что позиция самой левой клетки равна 0.

4. Нижняя оценка

Лемма 1. *Пусть S^1 — множество законов движения, состоящих из элементов множества $\{ssfs, s, sb\}$, S^2 — множество законов движения, состоящих из элементов множества $\{sfss, s, b\}$ и S^3 — множество законов движения, состоящих из элементов множества $\{ssf, ss, b\}$.*

Мощность множества состояний клеточного автомата, реализующего законы движения S^1, S^2 и S^3 $|Q| > 3$.

Доказательство. Предположим, что $|Q| = 3$, $Q = \{0, 1, 2\}$.

$0 \in Q \backslash L$, $1 \in L$, для 2 возможны два варианта:

1) $2 \in L$ ($L = \{1, 2\}$).

То есть у нас есть две метки и ни одного сигнала. В любой момент после начала и до конца движения на экране находится ровно одна метка.

Рассмотрим произвольный момент времени. Пусть на экране находится метка 1, в следующий момент времени она может: остаться на месте, переместиться вправо или влево, в то же время она может остаться меткой 1, или стать меткой 2. Таким образом, для метки 1 есть 6 вариантов развития событий в следующий момент.

То же самое верно и для метки 2. То есть, имея две метки и ни одного сигнала, возможно реализовать не более 6^2 законов движения, а нам требуется реализовать континуум, поскольку в S^1, S^2 и S^3 континуум законов движения.

2) $2 \in Q \backslash L$ ($L = \{1\}$).

Пусть $F^1 \subset S^1$ — множество законов движения, состоящих из элементов множества $\{ssfs, s\}$, $F^2 \subset S^2$ — множество законов движения, состоящих из элементов множества $\{sfss, s\}$, $F^3 \subset S^3$ — множество законов движения, состоящих из элементов множества $\{ssf, ss\}$. То есть движение назад нами исключено. Попробуем реализовать F^1, F^2 и F^3.

Метка всего одна. Все движения вперед можно осуществить только из $\varphi(1, a, b) = 1$, $a, b \in \{0, 2\}$, т.е. из предобработки. Под предобработкой здесь понимается конечное множество сигналов, поданных на вход до появления метки на экране.

К моменту появления метки на экране там уже находится какая-то конфигурация, состоящая из элементов множества $\{0, 2\}$. Таких конфигураций счётное число, а законов движения континуум.

□

5. Законы движения со скоростью движения вперёд 1/4 и скоростью движения назад 1/2

Рассмотрим клеточный автомат K_1. Множество состояний автомата имеет вид $Q = \{0, 1, 2, 3\}$, множество меток — $L = \{1, 2\}$. Функция переходов

будет следующая:

$$\begin{aligned}
\varphi(0,0,a) &= 0, \text{ где } a \in Q,\\
\varphi(0,1,a) &= 1, \text{ где } a \in Q\backslash L,\\
\varphi(0,2,0) &= 2,\\
\varphi(0,2,3) &= 0,\\
\varphi(0,3,a) &= 0, \text{ где } a \in Q\backslash L,\\
\varphi(0,3,1) &= 0,\\
\varphi(0,3,2) &= 1,\\
\varphi(1,a,b) &= 0, \text{ где } a \in Q\backslash L, b \in Q\backslash L,\\
\varphi(2,0,a) &= 3, \text{ где } a \in Q\backslash L,\\
\varphi(2,3,a) &= 1, \text{ где } a \in Q\backslash L,\\
\varphi(3,0,a) &= 3, \text{ где } a \in Q,\\
\varphi(3,1,a) &= 2, \text{ где } a \in Q\backslash L,\\
\varphi(3,2,0) &= 0,\\
\varphi(3,2,3) &= 3,\\
\varphi(3,3,a) &= 3, \text{ где } a \in Q,\\
\varphi(a,b,c) &= 0, \text{ где } a,b, \text{ или } b,c, \text{ или } a,c \in L.
\end{aligned}$$

Движение точки (1 и 2, выделенные жирным шрифтом) на экране вперёд изображено на рисунке 1*а*, движение назад — на рисунке 1*б*. Строки — моменты времени. Справа от вертикальной черты — картинка, которая получается на экране, слева — символ, который подаётся на управляющий вход; соответственно, символы слева от вертикальной черты, прочитанные сверху вниз — управляющая последовательность.

Эта функция переходов позволяет осуществлять движение точки на экране вперёд со скоростью $1/4$, назад — со скоростью $1/2$ с 4 состояниями клеточного автомата. Докажем это строго.

Лемма 2. *Пусть S^1 — множество законов движения, состоящих из элементов множества $\{ssfs, s, sb\}$. И пусть дано слово $\alpha = \alpha_1...\alpha_n... \in S^1$, сопоставим ему управляющую последовательность — слово $\beta = 22\beta_1...\beta_n...$, где*

$$\beta_i = \begin{cases} 0, & \text{если } \alpha_i = s,\\ 330, & \text{если } \alpha_i = sb,\\ 300, & \text{если } \alpha_i = ssfs. \end{cases}$$

Тогда, если на управляющий вход клеточного автомата K_1 подавать последовательность β, то справедливы следующие утверждения:

3	**1**	0	0	0
0	**2**	0	0	0
0	**2**	3	0	0
3	0	**1**	3	0
0	3	**1**	0	3
0	0	**2**	0	0
3	0	**2**	3	0
0	3	0	**1**	3
0	0	3	**1**	0

а

3	0	0	0	**1**
3	3	0	0	**1**
0	3	3	0	**1**
3	0	3	3	**1**
3	3	0	3	**2**
0	3	3	**1**	0
3	0	3	**2**	0
3	3	**1**	0	3
0	3	**2**	0	0
3	**1**	0	3	0
3	**2**	0	0	3

б

Рис. 1.

1) в момент времени $t(\overline{\alpha_{n-1}})$ *в позиции* $d(\overline{\alpha_{n-1}})-1$ *будет находиться символ* $\beta_n^\star$*, где*

$$\beta_n^\star = \begin{cases} 0, & \text{если } \alpha_n = s, \\ 3, & \text{если } \alpha_n = sb, \\ 3, & \text{если } \alpha_n = ssfs, \end{cases}$$

2) в момент времени $t(\overline{\alpha_n})$ *метка на экране будет в позиции* $d(\overline{\alpha_n})$.

Доказательство. Доказательство будем вести индукцией по длине n префикса закона движения.

Базис индукции: $n = 1$ и $n = 2$. Возможны следующие случаи.

1. $n = 1$. $\overline{\alpha_1} = \alpha_1 = sb$ — данный закон движения не из класса S^1, $n = 2$ рассматривать не нужно.

2. $n = 1$. $\overline{\alpha_1} = \alpha_1 = s$, $t(\overline{\alpha_1}) = 1$, $d(\overline{\alpha_1}) = 0$.

Подадим на вход последовательность 220. Согласно функции переходов будем наблюдать картину, изображённую на рисунке 2*а*. В дальнейшем символом * будем обозначать произвольный символ из множества $Q\backslash L$.

В момент времени $t = 0$ (момент появления метки на экране) в позиции $d = -1$, то есть на входе, будет находиться $\beta_1^\star = 0$. В момент времени $t(\overline{\alpha_1}) = 1$ метка на экране будет находиться в позиции $d(\overline{\alpha_1}) = 0$. Верно.

2.1. $n = 2$. $\overline{\alpha_2} = \alpha_1\alpha_2 = ssb$ — данный закон движения не из класса S^1.

2.2. $n = 2$. $\overline{\alpha_2} = \alpha_1\alpha_2 = ss$, $t(\overline{\alpha_2}) = 2$, $d(\overline{\alpha_2}) = 0$.

Подадим на вход последовательность 2200. Согласно функции переходов будем наблюдать картину, изображённую на рисунке 2*б*.

2	0	0	0
2	3	0	0
0	**1**	3	0
*	**1**	0	3

а

2	0	0	0
2	3	0	0
0	**1**	3	0
0	**1**	0	3
*	**1**	0	0

б

2	0	0	0
2	3	0	0
0	**1**	3	0
3	**1**	0	3
0	**2**	0	0
0	**2**	3	0
*	0	**1**	0
*	*	**1**	0

в

Рис. 2.

2	0	0	0
2	3	0	0
3	**1**	3	0
0	**2**	0	3
0	**2**	3	0
0	0	**1**	3
*	0	**1**	0
*	*	**1**	0

а

2	0	0	0
2	3	0	0
3	**1**	3	0
0	**2**	0	3
0	**2**	3	0
3	0	**1**	3
3	3	**1**	0
0	3	**2**	0
*	**1**	0	3

б

2	0	0	0
2	3	0	0
3	**1**	3	0
0	**2**	0	3
0	**2**	3	0
0	0	**1**	3
*	0	**1**	0
*	*	**1**	0

в

2	0	0	0	0
2	3	0	0	0
3	**1**	3	0	0
0	**2**	0	3	0
0	**2**	3	0	3
3	0	**1**	3	0
0	3	**1**	0	3
0	0	**2**	0	0
*	0	**2**	3	0
*	*	0	**1**	3
*	*	*	**1**	0

г

Рис. 3.

В момент времени $t(\overline{\alpha_1}) = 1$ в позиции $d(\overline{\alpha_1}) - 1 = -1$, то есть на входе, будет находиться $\beta_2^\star = 0$. В момент времени $t(\overline{\alpha_2}) = 2$ метка на экране будет находиться в позиции $d(\overline{\alpha_2}) = 0$. Верно.

2.3. $n = 2$. $\overline{\alpha_2} = \alpha_1\alpha_2 = sssfs$, $t(\overline{\alpha_2}) = 5$, $d(\overline{\alpha_2}) = 1$.

Подадим на вход последовательность 220300. Согласно функции переходов будем наблюдать картину, изображённую на рисунке 2*в*.

В момент времени $t(\overline{\alpha_1}) = 1$ в позиции $d(\overline{\alpha_1}) - 1 = -1$, то есть на входе, будет находиться $\beta_2^\star = 3$. В момент времени $t(\overline{\alpha_2}) = 5$ метка на экране будет находиться в позиции $d(\overline{\alpha_2}) = 1$. Верно.

3. $n = 1$. $\overline{\alpha_1} = \alpha_1 = ssfs$, $t(\overline{\alpha_1}) = 4$, $d(\overline{\alpha_1}) = 1$.

Подадим на вход последовательность 22300. Согласно функции переходов будем наблюдать картину, изображённую на рисунке 3*а*.

В момент времени $t = 0$ (момент появления метки на экране) в позиции $d = -1$, то есть на входе, будет находиться $\beta_1^\star = 3$. В момент времени $t(\overline{\alpha_1}) = 4$ метка на экране будет находиться в позиции $d(\overline{\alpha_1}) = 1$. Верно.

3.1. $n = 2$. $\overline{\alpha_2} = \alpha_1\alpha_2 = ssfssb$, $t(\overline{\alpha_2}) = 6$, $d(\overline{\alpha_2}) = 0$.

Подадим на вход последовательность 22300330. Согласно функции переходов будем наблюдать картину, изображённую на рисунке 3*б*.

В момент времени $t(\overline{\alpha_1}) = 4$ в позиции $d(\overline{\alpha_1}) - 1 = 0$, будет находиться $\beta_2^\star = 3$. В момент времени $t(\overline{\alpha_2}) = 6$ метка на экране будет находиться в позиции $d(\overline{\alpha_2}) = 0$. Верно.

3.2. $n = 2$. $\overline{\alpha_2} = \alpha_1\alpha_2 = ssfss$, $t(\overline{\alpha_2}) = 5$, $d(\overline{\alpha_2}) = 1$.

Подадим на вход последовательность 223000. Согласно функции переходов будем наблюдать картину, изображённую на рисунке 3*в*.

В момент времени $t(\overline{\alpha_1}) = 4$ в позиции $d(\overline{\alpha_1}) - 1 = 0$, будет находиться $\beta_2^\star = 0$. В момент времени $t(\overline{\alpha_2}) = 5$ метка на экране будет находиться в позиции $d(\overline{\alpha_2}) = 1$. Верно.

3.3. $n = 2$. $\overline{\alpha_2} = \alpha_1\alpha_2 = ssfsssfs$, $t(\overline{\alpha_2}) = 8$, $d(\overline{\alpha_2}) = 2$.

Подадим на вход последовательность 2300300. Согласно функции переходов будем наблюдать картину, изображённую на рисунке 3*г*.

В момент времени $t(\overline{\alpha_1}) = 4$ в позиции $d(\overline{\alpha_1}) - 1 = 0$, будет находиться $\beta_2^\star = 3$. В момент времени $t(\overline{\alpha_2}) = 8$ метка на экране будет находиться в позиции $d(\overline{\alpha_2}) = 2$. Верно.

Отметим, что в конце реализации слов движения вперёд, назад и остановки на экране находится метка 1.

Индуктивный переход. Пусть при $n = k - 1$ утверждение выполнено.

Тогда в момент времени $t(\overline{\alpha_{k-2}})$ в позиции $d(\overline{\alpha_{k-2}}) - 1$ будет находиться символ $\beta_{k-1}^\star$. А в момент времени времени $t(\overline{\alpha_{k-1}})$ метка 1 будет находиться в позиции $d(\overline{\alpha_{k-1}})$.

Докажем, что при $n = k$ утверждение также выполняется.

1) Покажем, что в момент времени $t(\overline{\alpha_{k-1}})$ в позиции $d(\overline{\alpha_{k-1}}) - 1$ будет находиться символ $\beta_k^\star$.

Сигнал (0 или 3) до встречи с меткой движется со скоростью 1 — это следует из рассматриваемой функции переходов. То есть, зная положение какого-то определённого сигнала в какой-то момент времени, можно однозначно определить, где он был несколько тактов назад, или где будет несколько тактов вперёд.

Рассмотрим момент времени $t(\overline{\alpha_{k-1}})$. Единица находится в позиции $d(\overline{\alpha_{k-1}})$ (предположение индукции).

1. Если последним в законе движения было движение вправо ($ssfs$), значит, в момент времени $t(\overline{\alpha_{k-1}}) - 4$ тройка стояла в позиции $d(\overline{\alpha_{k-1}}) - 2$, нули — в позициях $d(\overline{\alpha_{k-1}}) - 3$ и $d(\overline{\alpha_{k-1}}) - 4$, а $\beta_k^\star$ — в $d(\overline{\alpha_{k-1}}) - 5$. То есть в $t(\overline{\alpha_{k-1}})$ $\beta_k^\star$ будет в $d(\overline{\alpha_{k-1}}) - 1$ (рисунок 4*а*).

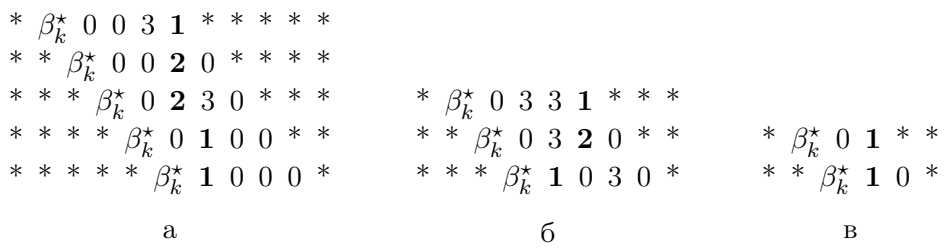


Рис. 4.

2. Если последним в законе движения было движение влево (sb), значит, в момент времени $t(\overline{\alpha_{k-1}}) - 2$ тройки стояли в позициях $d(\overline{\alpha_{k-1}})$ и $d(\overline{\alpha_{k-1}}) - 1$, 0 — в позиции $d(\overline{\alpha_{k-1}}) - 2$, а $\beta_k^\star$ — в $d(\overline{\alpha_{k-1}}) - 3$. То есть в $t(\overline{\alpha_{k-1}})$ $\beta_k^\star$ будет в $d(\overline{\alpha_{k-1}}) - 1$ (рисунок 4*б*).

3. Если последней в законе движения была остановка (s), значит, в момент времени $t(\overline{\alpha_{k-1}}) - 1$ ноль стоял в позиции $d(\overline{\alpha_{k-1}}) - 1$, а $\beta_k^\star$ — в $d(\overline{\alpha_{k-1}}) - 2$. То есть в $t(\overline{\alpha_{k-1}})$ $\beta_k^\star$ будет в $d(\overline{\alpha_{k-1}}) - 1$ (рисунок 4*в*).

2) Покажем, что в момент времени $t(\overline{\alpha_k})$ метка 1 будет находиться в позиции $d(\overline{\alpha_k})$.

В момент времени времени $t(\overline{\alpha_{k-1}})$ метка 1 находится в позиции $d(\overline{\alpha_{k-1}})$ (предположение индукции).

В позиции $d(\overline{\alpha_{k-1}}) - 1$ будет находиться символ $\beta_k^\star$ (согласно утверждению пункта 1 леммы).

Отдельно рассмотрим три случая для каждого возможного значения α_k.

1. $\alpha_k = s$. В этом случае выполнены соотношения:

$$\begin{aligned} I_{ST}(\overline{\alpha_k}) &= I_{ST}(\overline{\alpha_{k-1}}) + 1, \\ I_{FW}(\overline{\alpha_k}) &= I_{FW}(\overline{\alpha_{k-1}}), \\ I_{BC}(\overline{\alpha_k}) &= I_{BC}(\overline{\alpha_{k-1}}), \\ t(\overline{\alpha_k}) &= t(\overline{\alpha_{k-1}}) + 1, \\ d(\overline{\alpha_k}) &= d(\overline{\alpha_{k-1}}). \end{aligned}$$

Согласно функции переходов получим картину, изображённую на рисунке 5*а*.

В момент времени $t(\overline{\alpha_{k-1}}) = t(\overline{\alpha_k}) - 1$ в позиции $d(\overline{\alpha_k}) = d(\overline{\alpha_{k-1}})$ находится метка 1, в позиции $d(\overline{\alpha_{k-1}}) - 1 = d(\overline{\alpha_k}) - 1$ находится символ $\beta_k^\star = 0$.

Обозначим $q(d)$ символ в позиции d. В момент времени $t(\overline{\alpha_k})$ символ в позиции $d(\overline{\alpha_k})$ будет равен

$$q(d(\overline{\alpha_k})) = \varphi(\beta_k^\star, 1, a) = \varphi(0, 1, a) = 1, \text{ где } a \in Q \backslash L.$$

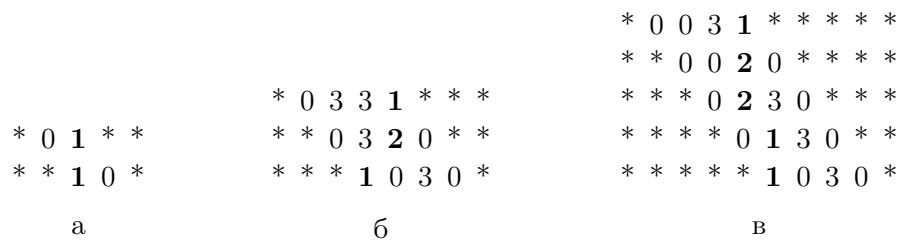


Рис. 5.

2. $\alpha_k = sb$. В этом случае выполнены соотношения:

$$\begin{aligned} I_{BC}(\overline{\alpha_k}) &= I_{BC}(\overline{\alpha_{k-1}}) + 1, \\ I_{FW}(\overline{\alpha_k}) &= I_{FW}(\overline{\alpha_{k-1}}), \\ I_{ST}(\overline{\alpha_k}) &= I_{ST}(\overline{\alpha_{k-1}}), \\ t(\overline{\alpha_k}) &= t(\overline{\alpha_{k-1}}) + 2, \\ d(\overline{\alpha_k}) &= d(\overline{\alpha_{k-1}}) - 1. \end{aligned}$$

Согласно функции переходов получим картину, изображённую на рисунке 5*б*.

В момент времени $t(\overline{\alpha_{k-1}}) = t(\overline{\alpha_k}) - 2$ в позиции $d(\overline{\alpha_{k-1}}) = d(\overline{\alpha_k}) + 1$ находится метка 1, в позиции $d(\overline{\alpha_{k-1}}) - 1 = d(\overline{\alpha_k})$ находится символ $\beta_k^\star = 3$. В момент времени $t(\overline{\alpha_{k-1}}) + 1 = t(\overline{\alpha_k}) - 1$ в позиции $d(\overline{\alpha_{k-1}}) = d(\overline{\alpha_k}) + 1$ находится метка 2:

$$q(d(\overline{\alpha_k}) + 1) = \varphi\,(3, 1, a) = 2, \text{ где } a \in Q \backslash L.$$

В момент времени $t(\overline{\alpha_k})$ символ в позиции $d(\overline{\alpha_k})$ будет равен

$$q(d(\overline{\alpha_k})) = \varphi\,(0, 3, 2) = 1.$$

3. $\alpha_k = ssfs$. В этом случае выполнены соотношения:

$$\begin{aligned} I_{FW}(\overline{\alpha_k}) &= I_{FW}(\overline{\alpha_{k-1}}) + 1, \\ I_{BC}(\overline{\alpha_k}) &= I_{BC}(\overline{\alpha_{k-1}}), \\ I_{ST}(\overline{\alpha_k}) &= I_{ST}(\overline{\alpha_{k-1}}), \\ t(\overline{\alpha_k}) &= t(\overline{\alpha_{k-1}}) + 4, \\ d(\overline{\alpha_k}) &= d(\overline{\alpha_{k-1}}) + 1. \end{aligned}$$

Согласно функции переходов получим картину, изображённую на рисунке 5*в*.

В момент времени $t(\overline{\alpha_{k-1}}) = t(\overline{\alpha_k}) - 4$ в позиции $d(\overline{\alpha_{k-1}}) = d(\overline{\alpha_k}) - 1$ находится метка 1, в позиции $d(\overline{\alpha_{k-1}}) - 1 = d(\overline{\alpha_k}) - 2$ находится символ $\beta_k^\star = 3$.

В момент времени $t(\overline{\alpha_{k-1}}) + 1 = t(\overline{\alpha_k}) - 3$ символ в позиции $d(\overline{\alpha_{k-1}}) = d(\overline{\alpha_k}) - 1$ будет равен

$$q(d(\overline{\alpha_k}) - 1) = \varphi\,(3, 1, a) = 2, \text{ где } a \in Q\backslash L.$$

В момент времени $t(\overline{\alpha_{k-1}})+2 = t(\overline{\alpha_k})-2$ символ в позиции $d(\overline{\alpha_{k-1}})+1 = d(\overline{\alpha_k})$ будет равен

$$q(d(\overline{\alpha_k})) = \varphi\,(2, 0, a) = 3, \text{ где } a \in Q\backslash L.$$

В момент времени $t(\overline{\alpha_{k-1}})+3 = t(\overline{\alpha_k})-1$ символ в позиции $d(\overline{\alpha_{k-1}})+1 = d(\overline{\alpha_k})$ будет равен

$$q(d(\overline{\alpha_k})) = \varphi\,(2, 3, a) = 1, \text{ где } a \in Q\backslash L.$$

В момент времени $t(\overline{\alpha_{k-1}}) + 4 = t(\overline{\alpha_k})$ символ в позиции $d(\overline{\alpha_{k-1}}) + 1 = d(\overline{\alpha_k})$ будет равен

$$q(d(\overline{\alpha_k})) = \varphi\,(0, 1, a) = 1, \text{ где } a \in Q\backslash L.$$

Мы доказали, что в момент времени $t(\overline{\alpha_k})$ метка 1 будет находиться в позиции $d(\overline{\alpha_k})$ для всех возможных вариантов движения. Значит, утверждение 2 леммы верно.

□

Фактически в лемме 2 приведен автомат с 4 состояниями, который образует универсальный экран для множества законов движения S^1. С другой стороны лемма 1 показывает, что трех состояний не достаточно для построения универсального экрана для S^1. Тем самым мы доказали справедливость утверждения теоремы 1.

6. Законы движения со скоростью движения вперёд 1/4 и скоростью движения назад 1

Рассмотрим клеточный автомат K_2. Множество состояний автомата имеет вид $Q = \{0, 1, 2, 3\}$, множество меток — $L = \{1, 2\}$. Функция переходов

будет следующая:

$$
\begin{aligned}
\varphi(0,a,b) &= 0, \text{ где } a \in Q\backslash L, b \in \{0,1,3\},\\
\varphi(3,a,b) &= 3, \text{ где } a \in Q\backslash L, b \in \{0,1,3\},\\
\varphi(2,0,a) &= 3, \text{ где } a \in Q\backslash L,\\
\varphi(2,3,a) &= 1, \text{ где } a \in Q\backslash L,\\
\varphi(1,a,b) &= 0, \text{ где } a \in Q\backslash L, b \in Q\backslash L,\\
\varphi(0,1,a) &= 1, \text{ где } a \in Q\backslash L,\\
\varphi(3,1,a) &= 2, \text{ где } a \in Q\backslash L,\\
\varphi(0,0,2) &= 1,\\
\varphi(3,0,2) &= 2,\\
\varphi(0,2,a) &= 0, \text{ где } a \in Q\backslash L,\\
\varphi(3,2,3) &= 0,\\
\varphi(0,3,2) &= 0,\\
\varphi(3,3,2) &= 3,\\
\varphi(3,2,0) &= 2,\\
\varphi(a,b,c) &= 0, \text{ где хотя бы два из трёх: } a \text{ и } b, \text{или } b \text{ и } c, \text{или } a \text{ и } c \in L.
\end{aligned}
$$

Движение точки (1 и 2, выделенные жирным шрифтом) на экране вперёд изображено на рисунке 6*а*, движение назад — на рисунке 6*б*. Строки — моменты времени. Справа от вертикальной черты — картинка, которая получается на экране, слева — символ, который подаётся на управляющий вход; соответственно, символы слева от вертикальной черты, прочитанные сверху вниз — управляющая последовательность.

Эта функция переходов позволяет осуществлять движение точки вперёд со скоростью $1/4$, назад — со скоростью 1 на экране с 4 состояниями клеточного автомата, с одним лишь ограничением: закон движения должен начинаться с s. Докажем это строго.

Лемма 3. *Пусть S^2 — множество законов движения, состоящих из элементов множества $\{sfss, s, b\}$ и начинающихся с элемента s. И пусть дано слово $\alpha = s\alpha_1...\alpha_n... \in S^2$, сопоставим ему управляющую последовательность — слово $\beta = 22\beta_1...\beta_n...$, где*

$$\beta_i = \begin{cases} 0, & \text{если } \alpha_i = s,\\ 30, & \text{если } \alpha_i = b,\\ 333, & \text{если } \alpha_i = sfss. \end{cases}$$

Тогда, если на управляющий вход клеточного автомата K_2 подавать последовательность β, то справедливы следующие утверждения:

2	0 0 0 0 0 0 0
2	3 0 0 0 0 0 0
3	**1** 3 0 0 0 0 0
3	**2** 0 3 0 0 0 0
3	**2** 3 0 3 0 0 0
3	0 **1** 3 0 3 0 0
3	3 **1** 0 3 0 3 0
3	3 **2** 0 0 3 0 3
3	3 **2** 3 0 0 3 0
3	3 0 **1** 3 0 0 3
3	3 3 **1** 0 3 0 0
3	3 3 **2** 0 0 3 0
3	3 3 **2** 3 0 0 3
3	3 3 0 **1** 3 0 0

а

3	0 0 0 0 **1** 0 0
0	3 0 0 0 **1** 0 0
3	0 3 0 0 **1** 0 0
0	3 0 3 0 **1** 0 0
3	0 3 0 3 **1** 0 0
0	3 0 3 0 **2** 0 0
3	0 3 0 **2** 0 3 0
0	3 0 **2** 0 3 0 3
3	0 **2** 0 3 0 3 0

б

Рис. 6.

1) в момент времени $t(\overline{\alpha_{n-1}}) - 1$ *в позиции* $d(\overline{\alpha_{n-1}}) - 1$ *будет находиться символ* $\beta_n^\star$*, где*

$$\beta_n^\star = \begin{cases} 0, & \text{если } \alpha_n = s, \\ 3, & \text{если } \alpha_n = b, \\ 3, & \text{если } \alpha_n = sfss, \end{cases}$$

2) в момент времени $t(\overline{\alpha_n})$ *метка на экране будет в позиции* $d(\overline{\alpha_n})$.

Доказательство. Доказательство будем вести индукцией по длине n префикса закона движения.

Базис индукции: $n = 1$ и $n = 2$. Возможны следующие случаи.

1. $n = 1$. $\overline{\alpha_1} = \alpha_1 = b$ — данный закон движения не из класса S^2, $n = 2$ рассматривать не нужно.

2. $n = 1$. $\overline{\alpha_1} = \alpha_1 = s$, $t(\overline{\alpha_1}) = 2$, $d(\overline{\alpha_1}) = 0$.

Подадим на вход последовательность 220. Согласно функции переходов будем наблюдать картину, изображённую на рисунке 7*a*. Символ * — произвольный символ из множества $Q \backslash L$, а $l \in L = \{1, 2\}$.

В момент времени $t = 0$ (момент появления метки на экране) в позиции $d = -1$, то есть на входе, будет находиться $\beta_1^\star = 0$. В момент времени $t(\overline{\alpha_1}) = 2$ метка на экране будет находиться в позиции $d(\overline{\alpha_1}) = 0$. Верно.

2.1. $n = 2$. $\overline{\alpha_2} = \alpha_1\alpha_2 = sb$ — данный закон движения не из класса S^2.

2	0	0	0	0
2	3	0	0	0
0	**1**	3	0	0
*	**1**	0	3	0
*	***l***	0	0	3

а

2	0	0	0	0
2	3	0	0	0
0	**1**	3	0	0
0	**1**	0	3	0
*	**1**	0	0	3
*	***l***	0	0	0

б

2	0	0	0	0
2	3	0	0	0
0	**1**	3	0	0
3	**1**	0	3	0
3	**2**	0	0	3
3	**2**	3	0	0
*	0	**1**	3	0
*	*	**1**	0	3
*	*	***l***	0	0

в

Рис. 7.

2.2. $n = 2$. $\overline{\alpha_2} = \alpha_1\alpha_2 = ss$, $t(\overline{\alpha_2}) = 3$, $d(\overline{\alpha_2}) = 0$.

Подадим на вход последовательность 2200. Согласно функции переходов будем наблюдать картину, изображённую на рисунке 7*б*.

В момент времени $t(\overline{\alpha_1}) - 1 = 1$ в позиции $d(\overline{\alpha_1}) - 1 = -1$, то есть на входе, будет находиться $\beta_2^\star = 0$. В момент времени $t(\overline{\alpha_2}) = 3$ метка на экране будет находиться в позиции $d(\overline{\alpha_2}) = 0$. Верно.

2.3. $n = 2$. $\overline{\alpha_2} = \alpha_1\alpha_2 = ssfss$, $t(\overline{\alpha_2}) = 6$, $d(\overline{\alpha_2}) = 1$.

Подадим на вход последовательность 220333. Согласно функции переходов будем наблюдать картину, изображённую на рисунке 7*в*.

В момент времени $t(\overline{\alpha_1}) - 1 = 1$ в позиции $d(\overline{\alpha_1}) - 1 = -1$, то есть на входе, будет находиться $\beta_2^\star = 3$. В момент времени $t(\overline{\alpha_2}) = 6$ метка на экране будет находиться в позиции $d(\overline{\alpha_2}) = 1$. Верно.

3. $n = 1$. $\overline{\alpha_1} = \alpha_1 = sfss$, $t(\overline{\alpha_1}) = 5$, $d(\overline{\alpha_1}) = 1$.

Подадим на вход последовательность 22333. Согласно функции переходов будем наблюдать картину, изображённую на рисунке 8*а*. Символ * — произвольный символ из множества $Q\backslash L$, а $l \in L = \{1, 2\}$.

В момент времени $t = 0$ (момент появления метки на экране) в позиции $d = -1$, то есть на входе, будет находиться $\beta_1^\star = 3$. В момент времени $t(\overline{\alpha_1}) = 5$ метка на экране будет находиться в позиции $d(\overline{\alpha_1}) = 1$. Верно.

3.1. $n = 2$. $\overline{\alpha_2} = \alpha_1\alpha_2 = sfssb$, $t(\overline{\alpha_2}) = 6$, $d(\overline{\alpha_2}) = 0$.

Подадим на вход последовательность 2233330. Согласно функции переходов будем наблюдать картину, изображённую на рисунке 8*б*.

В момент времени $t(\overline{\alpha_1}) - 1 = 4$ в позиции $d(\overline{\alpha_1}) - 1 = 0$ будет находиться $\beta_2^\star = 3$. В момент времени $t(\overline{\alpha_2}) = 6$ метка на экране будет находиться в позиции $d(\overline{\alpha_2}) = 0$. Верно.

3.2. $n = 2$. $\overline{\alpha_2} = \alpha_1\alpha_2 = sfsss$, $t(\overline{\alpha_2}) = 6$, $d(\overline{\alpha_2}) = 1$.

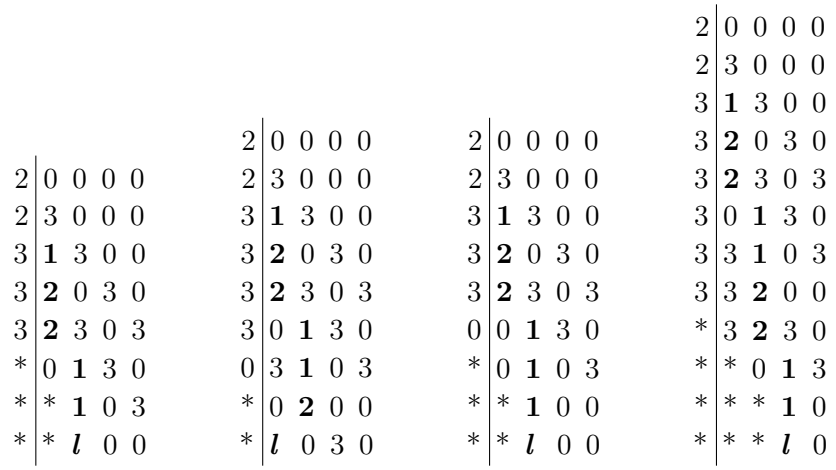


Рис. 8.

Подадим на вход последовательность 223330. Согласно функции переходов будем наблюдать картину, изображённую на рисунке 8*в*.

В момент времени $t(\overline{\alpha_1}) - 1 = 4$ в позиции $d(\overline{\alpha_1}) - 1 = 0$ будет находиться $\beta_2^\star = 0$. В момент времени $t(\overline{\alpha_2}) = 6$ метка на экране будет находиться в позиции $d(\overline{\alpha_2}) = 1$. Верно.

3.3. $n = 2$. $\overline{\alpha_2} = \alpha_1\alpha_2 = sfsssfss$, $t(\overline{\alpha_2}) = 9$, $d(\overline{\alpha_2}) = 2$.

Подадим на вход последовательность 22333333. Согласно функции переходов будем наблюдать картину, изображённую на рисунке 8*г*.

В момент времени $t(\overline{\alpha_1}) - 1 = 4$ в позиции $d(\overline{\alpha_1}) - 1 = 0$ будет находиться $\beta_2^\star = 3$. В момент времени $t(\overline{\alpha_2}) = 9$ метка на экране будет находиться в позиции $d(\overline{\alpha_2}) = 2$. Верно.

Индуктивный переход. Пусть при $n = k - 1$ утверждение выполнено.

То есть в момент времени $t(\overline{\alpha_{k-2}}) - 1$ в позиции $d(\overline{\alpha_{k-2}}) - 1$ будет находиться символ $\beta_{k-1}^\star$. А в момент времени времени $t(\overline{\alpha_{k-1}})$ метка будет находиться в позиции $d(\overline{\alpha_{k-1}})$.

Докажем, что при $n = k$ утверждение также выполняется.

1) Покажем, что в момент времени $t(\overline{\alpha_{k-1}}) - 1$ в позиции $d(\overline{\alpha_{k-1}}) - 1$ будет находиться символ $\beta_k^\star$.

Сигнал (0 или 3) до встречи с меткой движется со скоростью 1 — это следует из рассматриваемой функции переходов. То есть, зная положение какого-то определённого сигнала в какой-то момент времени, можно однозначно определить, где он был несколько тактов назад, или где будет несколько тактов вперёд.

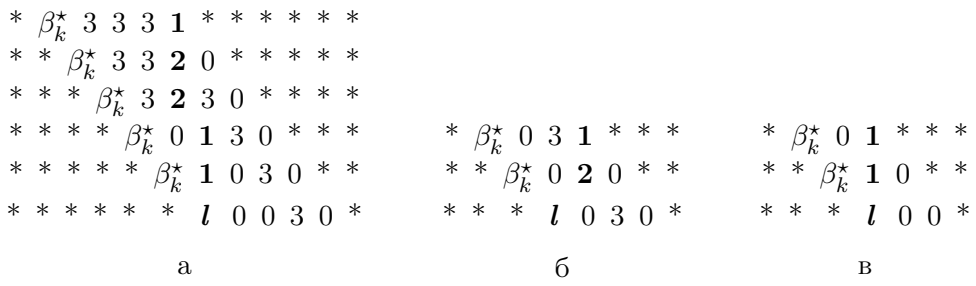


Рис. 9.

Рассмотрим момент времени $t(\overline{\alpha_{k-1}})$. Метка находится в позиции $d(\overline{\alpha_{k-1}})$ (предположение индукции).

1. Если последним в законе движения было движение вправо $(sfss)$, значит, в момент времени $t(\overline{\alpha_{k-1}})-5$ тройки стояли в позициях $d(\overline{\alpha_{k-1}})-2$, $d(\overline{\alpha_{k-1}})-3$ и $d(\overline{\alpha_{k-1}})-4$, а $\beta_k^\star$ — в $d(\overline{\alpha_{k-1}})-5$. То есть в $t(\overline{\alpha_{k-1}})-1$ $\beta_k^\star$ будет в $d(\overline{\alpha_{k-1}})-1$ (рисунок 9*а*). Символ * — произвольный символ из множества $Q\backslash L$, а $l \in L = \{1, 2\}$.

2. Если последним в законе движения было движение влево (b), значит, в момент времени $t(\overline{\alpha_{k-1}})-2$ тройка стояла в позиции $d(\overline{\alpha_{k-1}})$, 0 — в позиции $d(\overline{\alpha_{k-1}})-1$, а $\beta_k^\star$ — в $d(\overline{\alpha_{k-1}})-2$. То есть в $t(\overline{\alpha_{k-1}})-1$ $\beta_k^\star$ будет в $d(\overline{\alpha_{k-1}})-1$ (рисунок 9*б*).

3. Если последней в законе движения была остановка (s), значит, в момент времени $t(\overline{\alpha_{k-1}})-2$ ноль стоял в позиции $d(\overline{\alpha_{k-1}})-1$, а $\beta_k^\star$ — в $d(\overline{\alpha_{k-1}})-2$. То есть в $t(\overline{\alpha_{k-1}})-1$ $\beta_k^\star$ будет в $d(\overline{\alpha_{k-1}})-1$ (рисунок 9*в*).

2) Покажем, что в момент времени $t(\overline{\alpha_k})$ метка будет находиться в позиции $d(\overline{\alpha_k})$.

В момент времени времени $t(\overline{\alpha_{k-1}})$ метка находится в позиции $d(\overline{\alpha_{k-1}})$ (предположение индукции).

В момент времени времени $t(\overline{\alpha_{k-1}})-1$ символ $\beta_k^\star$ будет находиться в позиции $d(\overline{\alpha_{k-1}})-1$ (согласно утверждению пункта 1 леммы).

Отдельно рассмотрим три случая для каждого возможного значения α_k.

1. $\alpha_k = s$. В этом случае выполнены соотношения:

$$\begin{aligned} I_{ST}(\overline{\alpha_k}) &= I_{ST}(\overline{\alpha_{k-1}}) + 1, \\ I_{FW}(\overline{\alpha_k}) &= I_{FW}(\overline{\alpha_{k-1}}), \\ I_{BC}(\overline{\alpha_k}) &= I_{BC}(\overline{\alpha_{k-1}}), \\ t(\overline{\alpha_k}) &= t(\overline{\alpha_{k-1}}) + 1, \\ d(\overline{\alpha_k}) &= d(\overline{\alpha_{k-1}}). \end{aligned}$$

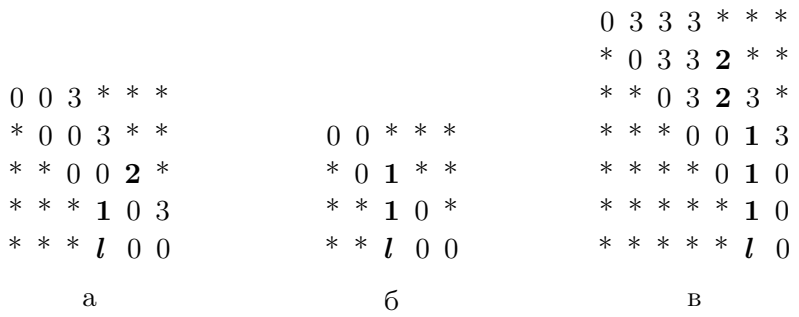


Рис. 10.

Есть два состояния, интерпретируемые как метки, и функция переходов определена так, что после одного и того же движения состояние, отвечающее за метку, может быть различным в зависимости от следующих символов в законе движения. Поэтому рассмотрим три случая значения α_{k-1}:

1.1. $\alpha_{k-1} = b$.

Согласно функции переходов получим картину, изображённую на рисунке 10*а*. Символ * — произвольный символ из множества $Q \backslash L$, а $l \in L = \{1, 2\}$.

В момент времени $t(\overline{\alpha_k}) = t(\overline{\alpha_{k-1}}) + 1$ символ в позиции $d(\overline{\alpha_k}) = d(\overline{\alpha_{k-1}})$. Верно.

1.2. $\alpha_{k-1} = s$.

Согласно функции переходов получим картину, изображённую на рисунке 10*б*.

В момент времени $t(\overline{\alpha_k}) = t(\overline{\alpha_{k-1}}) + 1$ символ в позиции $d(\overline{\alpha_k}) = d(\overline{\alpha_{k-1}})$. Верно.

1.3. $\alpha_{k-1} = sfss$.

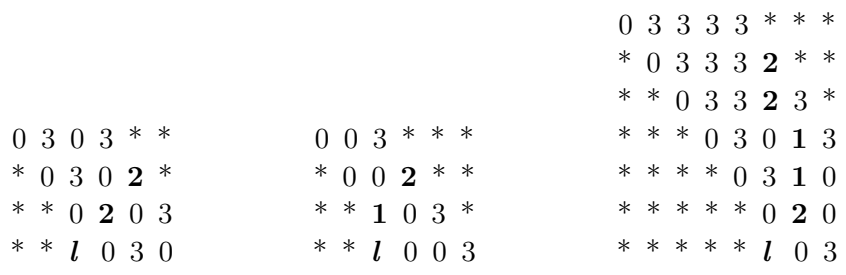
Согласно функции переходов получим картину, изображённую на рисунке 10*в*.

В момент времени $t(\overline{\alpha_k}) = t(\overline{\alpha_{k-1}}) + 1$ символ в позиции $d(\overline{\alpha_k}) = d(\overline{\alpha_{k-1}})$. Верно.

2. $\alpha_k = b$. В этом случае выполнены соотношения:

$$\begin{aligned} I_{BC}(\overline{\alpha_k}) &= I_{BC}(\overline{\alpha_{k-1}}) + 1, \\ I_{FW}(\overline{\alpha_k}) &= I_{FW}(\overline{\alpha_{k-1}}), \\ I_{ST}(\overline{\alpha_k}) &= I_{ST}(\overline{\alpha_{k-1}}), \\ t(\overline{\alpha_k}) &= t(\overline{\alpha_{k-1}}) + 1, \\ d(\overline{\alpha_k}) &= d(\overline{\alpha_{k-1}}) - 1. \end{aligned}$$

Как и в предыдущем пункте, рассмотрим три случая значения α_{k-1}:



а б в

Рис. 11.

2.1. $\alpha_{k-1} = b$.

Согласно функции переходов получим картину, изображённую на рисунке 11*а*. Символ * — произвольный символ из множества $Q \backslash L$, а $l \in L = \{1, 2\}$.

В момент времени $t(\overline{\alpha_k}) = t(\overline{\alpha_{k-1}}) + 1$ символ в позиции $d(\overline{\alpha_k}) = d(\overline{\alpha_{k-1}}) - 1$. Верно.

2.2. $\alpha_{k-1} = s$.

Согласно функции переходов получим картину, изображённую на рисунке 11*б*.

В момент времени $t(\overline{\alpha_k}) = t(\overline{\alpha_{k-1}}) + 1$ символ в позиции $d(\overline{\alpha_k}) = d(\overline{\alpha_{k-1}}) - 1$. Верно.

2.3. $\alpha_{k-1} = sfss$.

Согласно функции переходов получим картину, изображённую на рисунке 11*в*.

В момент времени $t(\overline{\alpha_k}) = t(\overline{\alpha_{k-1}}) + 1$ символ в позиции $d(\overline{\alpha_k}) = d(\overline{\alpha_{k-1}}) - 1$. Верно.

3. $\alpha_k = ssfs$. В этом случае выполнены соотношения:

$$\begin{aligned} I_{FW}(\overline{\alpha_k}) &= I_{FW}(\overline{\alpha_{k-1}}) + 1, \\ I_{BC}(\overline{\alpha_k}) &= I_{BC}(\overline{\alpha_{k-1}}), \\ I_S(\overline{\alpha_k}) &= I_S(\overline{\alpha_{k-1}}), \\ t(\overline{\alpha_k}) &= t(\overline{\alpha_{k-1}}) + 4, \\ d(\overline{\alpha_k}) &= d(\overline{\alpha_{k-1}}) + 1. \end{aligned}$$

Как и в предыдущих пунктах, рассмотрим три случая значения α_{k-1}:

3.1. $\alpha_{k-1} = b$.

Согласно функции переходов получим картину, изображённую на рисунке 12*а*. Символ * — произвольный символ из множества $Q \backslash L$, а $l \in L = \{1, 2\}$.

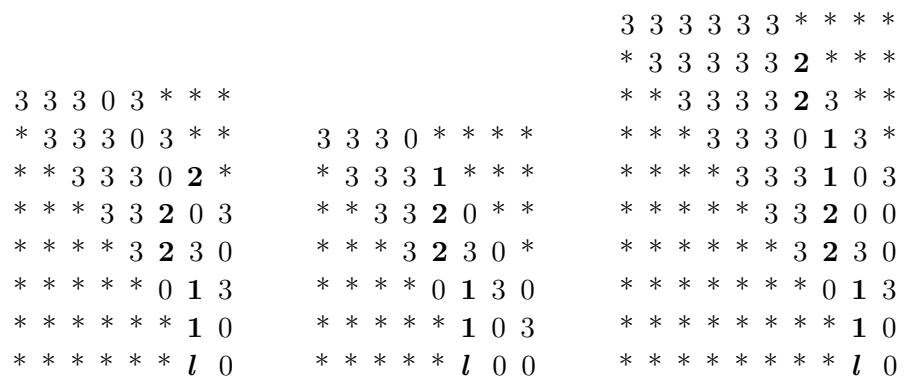


Рис. 12.

В момент времени $t(\overline{\alpha_k}) = t(\overline{\alpha_{k-1}}) + 4$ символ в позиции $d(\overline{\alpha_k}) = d(\overline{\alpha_{k-1}}) + 1$. Верно.

3.2. $\alpha_{k-1} = s$.

Согласно функции переходов получим картину, изображённую на рисунке 12*б*.

В момент времени $t(\overline{\alpha_k}) = t(\overline{\alpha_{k-1}}) + 4$ символ в позиции $d(\overline{\alpha_k}) = d(\overline{\alpha_{k-1}}) + 1$. Верно.

3.3. $\alpha_{k-1} = sfss$.

Согласно функции переходов получим картину, изображённую на рисунке 12*в*.

В момент времени $t(\overline{\alpha_k}) = t(\overline{\alpha_{k-1}}) + 4$ символ в позиции $d(\overline{\alpha_k}) = d(\overline{\alpha_{k-1}}) + 1$. Верно.

Мы доказали, что в момент времени $t(\overline{\alpha_k})$ метка будет находиться в позиции $d(\overline{\alpha_k})$ для всех возможных вариантов движения. Значит, утверждение 2 леммы верно.

□

Фактически в лемме 3 приведен автомат с 4 состояниями, который образует универсальный экран для множества законов движения S^2. С другой стороны лемма 1 показывает, что трех состояний не достаточно для построения универсального экрана для S^2. Тем самым мы доказали справедливость утверждения теоремы 2.

7. Законы движения со скоростью движения вперёд 1/3, скоростью движения назад 1 и чётным количеством остановок

Рассмотрим клеточный автомат K_3. Множество состояний автомата имеет вид $Q = \{0, 1, 2, 3\}$, множество меток — $L = \{1, 2\}$. Функция переходов будет следующая:

$$
\begin{aligned}
\varphi(a, 0, b) &= a, \text{ где } a \in Q \backslash L, b \in Q, \\
\varphi(0, 1, a) &= 2, \text{ где } a \in Q \backslash L, \\
\varphi(0, 2, 0) &= 2, \\
\varphi(0, 2, 3) &= 0, \\
\varphi(a, 3, b) &= a, \text{ где } a \in Q \backslash L, b \in \{0, 2, 3\}, \\
\varphi(a, 3, 1) &= 1, \text{ где } a \in Q \backslash L, \\
\varphi(1, a, b) &= 0, \text{ где } a \in Q \backslash L, b \in Q, \\
\varphi(2, 0, a) &= 3, \text{ где } a \in Q \backslash L, \\
\varphi(2, 3, a) &= 1, \text{ где } a \in Q \backslash L, \\
\varphi(3, 1, a) &= 0, \text{ где } a \in Q \backslash L, \\
\varphi(3, 2, 0) &= 1, \\
\varphi(3, 2, 3) &= 3, \\
\varphi(a, b, c) &= 0, \text{ где два из трёх: } a \text{ и } b, \text{ или } b \text{ и } c, \text{ или } a \text{ и } c \in L.
\end{aligned}
$$

Движение точки (1 и 2, выделенные жирным шрифтом) на экране вперёд изображено на рисунке 13*а*, движение назад — на рисунке 13*б*, остановка — на рисунке 13*в*. Строки — моменты времени. Справа от вертикальной черты — картинка, которая получается на экране, слева — символ, который подаётся на управляющий вход; соответственно, символы слева от вертикальной черты, прочитанные сверху вниз — управляющая последовательность.

Эта функция переходов позволяет осуществлять движение точки вперёд со скоростью 1/3, назад — со скоростью 1 и останавливаться на чётное количество тактов на экране с 4 состояниями клеточного автомата. Докажем это строго.

Лемма 4. *Пусть S^3 — множество законов движения, состоящих из элементов множества $\{ssf, ss, b\}$. И пусть дано слово $\alpha = \alpha_1 ... \alpha_n ... \in S^3$, сопоставим ему управляющую последовательность — слово $\beta =$*

2	0	0	0	0
2	3	0	0	0
0	**1**	3	0	0
0	**2**	0	3	0
0	**2**	3	0	3
0	0	**1**	3	0
0	0	**2**	0	3
0	0	**2**	3	0
0	0	0	**1**	3

а

3	3	3	3	**1**
3	3	3	**1**	0
3	3	**1**	0	0
3	**1**	0	0	0

б

3	0	**1**	0	0
0	3	**2**	0	0
3	0	**1**	3	0
0	3	**2**	0	3
3	0	**1**	3	0
0	3	**2**	0	3

в

Рис. 13.

$22\beta_1...\beta_n...$, *где*

$$\beta_i = \begin{cases} 03, & \text{если } \alpha_i = ss, \\ 33, & \text{если } \alpha_i = b, \\ 00, & \text{если } \alpha_i = ssf. \end{cases}$$

Тогда, если на управляющий вход клеточного автомата K_3 *подавать последовательность* β*, то справедливы следующие утверждения:*

1) в момент времени $t(\overline{\alpha_{n-1}})$ *в позиции* $d(\overline{\alpha_{n-1}})-1$ *будет находиться символ* $\beta_n^\star$*, где*

$$\beta_n^\star = \begin{cases} 0, & \text{если } \alpha_n = ss, \\ 3, & \text{если } \alpha_n = b, \\ 0, & \text{если } \alpha_n = ssf, \end{cases}$$

2) в момент времени $t(\overline{\alpha_n})$ *метка на экране будет в позиции* $d(\overline{\alpha_n})$*.*

Доказательство. Доказательство будем вести индукцией по длине n префикса закона движения.

Базис индукции: $n = 1$ и $n = 2$. Возможны следующие случаи.

1. $n = 1$. $\overline{\alpha_1} = \alpha_1 = b$ — данный закон движения не из класса S^3, $n = 2$ рассматривать не нужно.

2. $n = 1$. $\overline{\alpha_1} = \alpha_1 = ss$, $t(\overline{\alpha_1}) = 2$, $d(\overline{\alpha_1}) = 0$.

Подадим на вход последовательность 2203. Согласно функции переходов будем наблюдать картину, изображённую на рисунке 14*a*. В дальнейшем символом * будем обозначать произвольный символ из множества $Q\backslash L$.

В момент времени $t = 0$ (момент появления метки на экране) в позиции $d = -1$, то есть на входе, будет находиться $\beta_1^\star = 0$. В момент времени $t(\overline{\alpha_1}) = 2$ метка на экране будет находиться в позиции $d(\overline{\alpha_1}) = 0$. Верно.

2	0	0	0	0
2	3	0	0	0
0	**1**	3	0	0
3	**2**	0	3	0
*	**1**	3	0	3

а

2	0	0	0	0
2	3	0	0	0
0	**1**	3	0	0
3	**2**	0	3	0
0	**1**	3	0	3
3	**2**	0	3	0
*	**1**	3	0	3

б

2	0	0	0	0
2	3	0	0	0
0	**1**	3	0	0
3	**2**	0	3	0
0	**1**	3	0	3
0	**2**	0	3	0
*	**2**	3	0	3
*	*	**1**	3	0

в

Рис. 14.

2.1. $n = 2$. $\overline{\alpha_2} = \alpha_1\alpha_2 = ssb$ — данный закон движения не из класса S^3.

2.2. $n = 2$. $\overline{\alpha_2} = \alpha_1\alpha_2 = ssss$, $t(\overline{\alpha_2}) = 4$, $d(\overline{\alpha_2}) = 0$.

Подадим на вход последовательность 220303. Согласно функции переходов будем наблюдать картину, изображённую на рисунке 14*б*.

В момент времени $t(\overline{\alpha_1}) = 2$ в позиции $d(\overline{\alpha_1}) - 1 = -1$, то есть на входе, будет находиться $\beta_2^\star = 0$. В момент времени $t(\overline{\alpha_2}) = 4$ метка на экране будет находиться в позиции $d(\overline{\alpha_2}) = 0$. Верно.

2.3. $n = 2$. $\overline{\alpha_2} = \alpha_1\alpha_2 = sssf$, $t(\overline{\alpha_2}) = 5$, $d(\overline{\alpha_2}) = 1$.

Подадим на вход последовательность 220300. Согласно функции переходов будем наблюдать картину, изображённую на рисунке 14*в*.

В момент времени $t(\overline{\alpha_1}) = 2$ в позиции $d(\overline{\alpha_1}) - 1 = -1$, то есть на входе, будет находиться $\beta_2^\star = 0$. В момент времени $t(\overline{\alpha_2}) = 5$ метка на экране будет находиться в позиции $d(\overline{\alpha_2}) = 1$. Верно.

3. $n = 1$. $\overline{\alpha_1} = \alpha_1 = ssf$, $t(\overline{\alpha_1}) = 3$, $d(\overline{\alpha_1}) = 1$.

Подадим на вход последовательность 2200. Согласно функции переходов будем наблюдать картину, изображённую на рисунке 15*а*.

В момент времени $t = 0$ (момент появления метки на экране) в позиции $d = -1$, то есть на входе, будет находиться $\beta_1^\star = 0$. В момент времени $t(\overline{\alpha_1}) = 3$ метка на экране будет находиться в позиции $d(\overline{\alpha_1}) = 1$. Верно.

3.1. $n = 2$. $\overline{\alpha_2} = \alpha_1\alpha_2 = ssfb$, $t(\overline{\alpha_2}) = 4$, $d(\overline{\alpha_2}) = 0$.

Подадим на вход последовательность 220033. Согласно функции переходов будем наблюдать картину, изображённую на рисунке 15*б*.

В момент времени $t(\overline{\alpha_1}) = 3$ в позиции $d(\overline{\alpha_1}) - 1 = 0$ будет находиться $\beta_2^\star = 3$. В момент времени $t(\overline{\alpha_2}) = 4$ метка на экране будет находиться в позиции $d(\overline{\alpha_2}) = 0$. Верно.

3.2. $n = 2$. $\overline{\alpha_2} = \alpha_1\alpha_2 = ssfss$, $t(\overline{\alpha_2}) = 5$, $d(\overline{\alpha_2}) = 1$.

2	0	0	0	0
2	3	0	0	0
0	**1**	3	0	0
0	**2**	0	3	0
*	**2**	3	0	3
*	*	**1**	3	0

а

2	0	0	0	0
2	3	0	0	0
0	**1**	3	0	0
0	**2**	0	3	0
3	**2**	3	0	3
3	3	**1**	3	0
*	**1**	0	0	3

б

2	0	0	0	0
2	3	0	0	0
0	**1**	3	0	0
0	**2**	0	3	0
0	**2**	3	0	3
3	0	**1**	3	0
*	3	**2**	0	3
*	*	**1**	3	0

в

2	0	0	0	0
2	3	0	0	0
0	**1**	3	0	0
0	**2**	0	3	0
0	**2**	3	0	3
0	0	**1**	3	0
*	0	**2**	0	3
*	*	**2**	3	0
*	*	*	**1**	3

г

Рис. 15.

Подадим на вход последовательность 220003. Согласно функции переходов будем наблюдать картину, изображённую на рисунке 15*в*.

В момент времени $t(\overline{\alpha_1}) = 3$ в позиции $d(\overline{\alpha_1}) - 1 = 0$ будет находиться $\beta_2^\star = 0$. В момент времени $t(\overline{\alpha_2}) = 5$ метка на экране будет находиться в позиции $d(\overline{\alpha_2}) = 1$. Верно.

3.3. $n = 2$. $\overline{\alpha_2} = \alpha_1\alpha_2 = ssfssf$, $t(\overline{\alpha_2}) = 6$, $d(\overline{\alpha_2}) = 2$.

Подадим на вход последовательность 220000. Согласно функции переходов будем наблюдать картину, изображённую на рисунке 15*г*.

В момент времени $t(\overline{\alpha_1}) = 3$ в позиции $d(\overline{\alpha_1}) - 1 = 0$ будет находиться $\beta_2^\star = 0$. В момент времени $t(\overline{\alpha_2}) = 6$ метка на экране будет находиться в позиции $d(\overline{\alpha_2}) = 2$. Верно.

Отметим, что в конце реализации слов движения вперёд, назад и остановки на экране находится метка 1.

Индуктивный переход. Пусть при $n = k - 1$ утверждение выполнено.

То есть в момент времени $t(\overline{\alpha_{k-2}})$ в позиции $d(\overline{\alpha_{k-2}}) - 1$ будет находиться символ $\beta_{k-1}^\star$. А в момент времени времени $t(\overline{\alpha_{k-1}})$ метка будет находиться в позиции $d(\overline{\alpha_{k-1}})$.

Докажем, что при $n = k$ утверждение также выполняется.

1) Покажем, что в момент времени $t(\overline{\alpha_{k-1}})$ в позиции $d(\overline{\alpha_{k-1}}) - 1$ будет находиться символ $\beta_k^\star$.

Сигнал (0 или 3) до встречи с меткой движется со скоростью 1 — это следует из рассматриваемой функции переходов. То есть, зная положение какого-то определённого сигнала в какой-то момент времени, можно однозначно определить, где он был несколько тактов назад, или где будет несколько тактов вперёд.

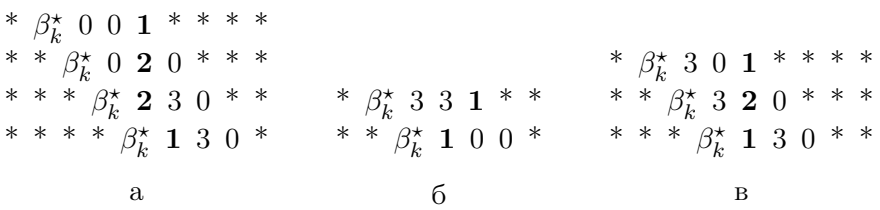


Рис. 16.

Рассмотрим момент времени $t(\overline{\alpha_{k-1}})$. Единица находится в позиции $d(\overline{\alpha_{k-1}})$ (предположение индукции).

1. Если последним в законе движения было движение вправо (ssf), значит, в момент времени $t(\overline{\alpha_{k-1}}) - 3$ нули стояли в позициях $d(\overline{\alpha_{k-1}}) - 2$ и $d(\overline{\alpha_{k-1}}) - 3$, а $\beta_k^\star$ — в $d(\overline{\alpha_{k-1}}) - 4$. То есть в $t(\overline{\alpha_{k-1}})$ $\beta_k^\star$ будет в $d(\overline{\alpha_{k-1}}) - 1$ (рисунок 16*а*).

2. Если последним в законе движения было движение влево (b), значит, в момент времени $t(\overline{\alpha_{k-1}})-1$ тройки стояли в позициях $d(\overline{\alpha_{k-1}})$ и $d(\overline{\alpha_{k-1}})-1$, а $\beta_k^\star$ — в позиции $d(\overline{\alpha_{k-1}}) - 2$. То есть в $t(\overline{\alpha_{k-1}})$ $\beta_k^\star$ будет в $d(\overline{\alpha_{k-1}}) - 1$ (рисунок 16*б*).

3. Если последней в законе движения была остановка (ss), значит, в момент времени $t(\overline{\alpha_{k-1}}) - 2$ ноль стоял в позиции $d(\overline{\alpha_{k-1}}) - 1$, тройка — в позиции $d(\overline{\alpha_{k-1}}) - 2$, а $\beta_k^\star$ — в $d(\overline{\alpha_{k-1}}) - 3$. То есть в $t(\overline{\alpha_{k-1}})$ $\beta_k^\star$ будет в $d(\overline{\alpha_{k-1}}) - 1$ (рисунок 16*в*).

2) Покажем, что в момент времени $t(\overline{\alpha_k})$ метка будет находиться в позиции $d(\overline{\alpha_k})$.

В момент времени времени $t(\overline{\alpha_{k-1}})$ метка находится в позиции $d(\overline{\alpha_{k-1}})$ (предположение индукции).

В момент времени времени $t(\overline{\alpha_{k-1}})$ символ $\beta_k^\star$ будет находиться в позиции $d(\overline{\alpha_{k-1}}) - 1$ (согласно утверждению пункта 1 леммы).

Отдельно рассмотрим три случая для каждого возможного значения α_k.

1. $\alpha_k = ss$. В этом случае выполнены соотношения:

$$\begin{aligned} I_{ST}(\overline{\alpha_k}) &= I_{ST}(\overline{\alpha_{k-1}}) + 1, \\ I_{FW}(\overline{\alpha_k}) &= I_{FW}(\overline{\alpha_{k-1}}), \\ I_{BC}(\overline{\alpha_k}) &= I_{BC}(\overline{\alpha_{k-1}}), \\ t(\overline{\alpha_k}) &= t(\overline{\alpha_{k-1}}) + 2, \\ d(\overline{\alpha_k}) &= d(\overline{\alpha_{k-1}}). \end{aligned}$$

Согласно функции переходов получим картину, изображённую на рисунке 17*а*.

* 3 0 **1** * * *		* 0 0 **1** * * * *
* * 3 **2** 0 * *	* 3 3 **1** * *	* * 0 **2** 0 * * *
* * * **1** 3 0 *	* * **1** 0 0 *	* * * **2** 3 0 * *
		* * * * **1** 3 0 *
а	б	в

Рис. 17.

В момент времени $t(\overline{\alpha_{k-1}}) = t(\overline{\alpha_k}) - 2$ в позиции $d(\overline{\alpha_k}) = d(\overline{\alpha_{k-1}})$ находится метка 1, в позиции $d(\overline{\alpha_{k-1}}) - 1 = d(\overline{\alpha_k}) - 1$ находится символ $\beta_k^\star = 0$.

В момент времени $t(\overline{\alpha_{k-1}}) + 1 = t(\overline{\alpha_k}) - 1$ символ в позиции $d(\overline{\alpha_k})$ будет равен

$$q(d(\overline{\alpha_k})) = \varphi(\beta_k^\star, 1, a) = \varphi(0, 1, a) = 2, \text{ где } a \in Q \backslash L.$$

В момент времени $t(\overline{\alpha_k})$ символ в позиции $d(\overline{\alpha_k})$ будет равен

$$q(d(\overline{\alpha_k})) = \varphi(3, 2, 0) = 1.$$

2. $\alpha_k = b$. В этом случае выполнены соотношения:

$$\begin{aligned} I_{BC}(\overline{\alpha_k}) &= I_{BC}(\overline{\alpha_{k-1}}) + 1, \\ I_{FW}(\overline{\alpha_k}) &= I_{FW}(\overline{\alpha_{k-1}}), \\ I_{ST}(\overline{\alpha_k}) &= I_{ST}(\overline{\alpha_{k-1}}), \\ t(\overline{\alpha_k}) &= t(\overline{\alpha_{k-1}}) + 1, \\ d(\overline{\alpha_k}) &= d(\overline{\alpha_{k-1}}) - 1. \end{aligned}$$

Согласно функции переходов получим картину, изображённую на рисунке 17*б*.

В момент времени $t(\overline{\alpha_{k-1}}) = t(\overline{\alpha_k}) - 1$ в позиции $d(\overline{\alpha_{k-1}}) = d(\overline{\alpha_k}) + 1$ находится метка 1, в позиции $d(\overline{\alpha_{k-1}}) - 1 = d(\overline{\alpha_k})$ находится символ $\beta_k^\star = 3$.

В момент времени $t(\overline{\alpha_k})$ символ в позиции $d(\overline{\alpha_k})$ будет равен

$$q(d(\overline{\alpha_k})) = \varphi\left(\beta_k^\star, 3, 1\right) = \varphi\left(3, 3, 1\right) = 1.$$

3. $\alpha_k = ssf$. В этом случае выполнены соотношения:

$$\begin{aligned} I_{FW}(\overline{\alpha_k}) &= I_{FW}(\overline{\alpha_{k-1}}) + 1, \\ I_{BC}(\overline{\alpha_k}) &= I_{BC}(\overline{\alpha_{k-1}}), \\ I_{ST}(\overline{\alpha_k}) &= I_{ST}(\overline{\alpha_{k-1}}), \\ t(\overline{\alpha_k}) &= t(\overline{\alpha_{k-1}}) + 3, \\ d(\overline{\alpha_k}) &= d(\overline{\alpha_{k-1}}) + 1. \end{aligned}$$

Согласно функции переходов получим картину, изображённую на рисунке 17*в*.

В момент времени $t(\overline{\alpha_{k-1}}) = t(\overline{\alpha_k}) - 3$ в позиции $d(\overline{\alpha_{k-1}}) = d(\overline{\alpha_k}) - 1$ находится метка 1, в позиции $d(\overline{\alpha_{k-1}}) - 1 = d(\overline{\alpha_k}) - 2$ находится символ $\beta_k^\star = 0$.

В момент времени $t(\overline{\alpha_{k-1}}) + 1 = t(\overline{\alpha_k}) - 2$ символ в позиции $d(\overline{\alpha_{k-1}}) = d(\overline{\alpha_k}) - 1$ будет равен

$$q(d(\overline{\alpha_k})) = \varphi\left(\beta_k^\star, 1, a\right) = \varphi\left(0, 1, a\right) = 2, \text{ где } a \in Q \backslash L.$$

В момент времени $t(\overline{\alpha_{k-1}}) + 2 = t(\overline{\alpha_k}) - 1$ символ в позиции $d(\overline{\alpha_{k-1}}) = d(\overline{\alpha_k}) - 1$ будет равен

$$q(d(\overline{\alpha_k})) = \varphi\left(0, 2, 0\right) = 2.$$

В момент времени $t(\overline{\alpha_{k-1}})+2 = t(\overline{\alpha_k})-1$ символ в позиции $d(\overline{\alpha_{k-1}})+1 = d(\overline{\alpha_k})$ будет равен

$$q(d(\overline{\alpha_k})) = \varphi\left(2, 0, a\right) = 3, \text{ где } a \in Q \backslash L.$$

В момент времени $t(\overline{\alpha_{k-1}}) + 3 = t(\overline{\alpha_k})$ символ в позиции $d(\overline{\alpha_{k-1}}) + 1 = d(\overline{\alpha_k})$ будет равен

$$q(d(\overline{\alpha_k})) = \varphi\left(2, 3, a\right) = 1, \text{ где } a \in Q \backslash L.$$

Мы доказали, что в момент времени $t(\overline{\alpha_k})$ метка 1 будет находиться в позиции $d(\overline{\alpha_k})$ для всех возможных вариантов движения. Значит, утверждение 2 леммы верно.

□

Фактически в лемме 4 приведен автомат с 4 состояниями, который образует универсальный экран для множества законов движения S^3. С другой стороны лемма 1 показывает, что трех состояний не достаточно для построения универсального экрана для S^3. Тем самым мы доказали справедливость утверждения теоремы 3.

Список литературы

[1] Кузнецова Е.В., “Число состояний универсального автомата бесконечного экрана, реализующего двунаправленное движение на луче”, *Интеллектуальные системы. Теория и приложения*, **25**:1 (2021), 127–148.

[2] Титова Е.Е., “Конструирование изображений клеточными автоматами”, *Интеллектуальные системы*, **12**:2 (2008), 105–121.

[3] Титова Е.Е., “Линейное по времени конструирование изображений клеточными автоматами”, *Интеллектуальные системы*, **16**:2 (2012), 215–234.

[4] Калачев Г.В.,Титова Е.Е., “О мере множества законов движения точки, реализуемых клеточными автоматами”, *Интеллектуальные системы. Теория и приложения*, **22**:3 (2018), 105–125.

[5] Гасанов Э.Э., “Клеточные автоматы с локаторами”, *Интеллектуальные системы. Теория и приложения*, **24**:2 (2020), 119–132.

[6] Гасанов Э.Э., “Клеточные автоматы с локаторами как модель устройств с беспроводной связью”, *Математические вопросы кибернетики*, **21** (2023), 5–51.

[7] Васильев Д. И., Гасанов Э. Э., “Нижняя оценка сложности задачи поиска ближайшего соседа на прямой с помощью клеточного автомата с локаторами”, *Вестник МГУ. Серия 1. Математика и механика*, **5** (2023), 33–39.

[8] Гасанов Э.Э., Хайбуллин Б.Ф., “Быстрые алгоритмы умножения и деления натуральных чисел с помощью клеточных автоматов с локаторами”, *Интеллектуальные системы. Теория и приложения*, **28**:3 (2024), 103–130.

[9] Титова Е.Е., “Конструирование движущихся изображений клеточными автоматами”, *Интеллектуальные системы*, **18**:1 (2014), 153–180.

Classes of bidirectional motion on a beam implemented by 4-state automata

Kuznetsova E.V.

In [1] it is shown that there is a universal screen with 5 states for the class of all laws of motion with a forward speed of no more than 1/2, while there is no universal screen with 4 states for this class of laws of motion. This paper presents 3 classes of bidirectional motion cocoons on a ray that can be implemented by a cellular automaton with 4 states.

Keywords: cellular automaton, number of states, infinite screen, bidirectional motion, image construction.

References

[1] Kuznetsova E. V., “The number of states of a universal infinite screen automaton implementing bidirectional motion on a ray”, *Intelligent Systems. Theory and Applications*, **25**:1 (2021), 127–148 (In Russian).

[2] Titova E. E., “Image construction by cellular automata”, *Intelligent Systems*, **12**:2 (2008), 105–121 (In Russian).

[3] Titova E. E., “Linear-time image construction by cellular automata”, *Intelligent Systems*, **16**:2 (2012), 215–234 (In Russian).

[4] Kalachev G. V.,Titova E. E., “On the measure of the set of laws of motion of a point realized by cellular automata”, *Intelligent Systems. Theory and Applications*, **22**:3 (2018), 105–125 (In Russian).

[5] Gasanov E. E., “Cellular automata with locators”, *Intelligent Systems. Theory and Applications*, **24**:2 (2020), 119–132 (In Russian).

[6] Gasanov E. E., “Cellular automata with locators as a model for wireless communication devices”, *Mathematical issues of cybernetics*, **21** (2023), 5–51 (In Russian).

[7] Vasilev D. I., Gasanov E. E., “Lower bound for the complexity of the nearest neighbor problem on a line using a cellular automaton with locators”, *MSU Bulletin. Series 1. Mathematics and Mechanics*, **5** (2023), 33–39 (In Russian).

[8] Gasanov E. E., Khaybullin B. F., “Fast algorithms for multiplication and division of natural numbers using cellular automata with locators”, *Intelligent Systems. Theory and Applications*, **28**:3 (2024), 103–130 (In Russian).

[9] Titova E. E., “Construction of moving images by cellular automata”, *Intelligent Systems*, **18**:1 (2014), 153–180 (In Russian).

Представление СФЭ

М. В. Носов[1]

В работе получено представление всех схем из функциональных элементов с одинаковым числом элементов в базисе из штриха Шеффера для булевской функции в виде натурального числа. Расшифровка этого числа позволяет построить все схемы.

Ключевые слова: булевская функция, схема из функциональных элементов.

В данной работе используются описания СФЭ и задаваемой булевской функции, которые даны в статье автора "Об аналитическом представлении функции сложности минимальной схемы в базисе из штриха Шеффера" (Интеллектуальные системы. Теория и приложения.21:2 (2017), 193-196).

Рассмотрим СФЭ в базисе из Штриха Шеффера, схема должна задавать булевскую функцию $f(y_1,\dots,y_n)$. Элементов в схеме L, они пронумерованы числами $n+1,\dots,n+L$; k-ый элемент схемы имеет входами и (или) входы схемы, и (или) выходы элементов с меньшими номерами, таким образом, появляются двойки $(i_k,j_k), 1\le i_k\le k-1, 1\le j_k\le k-1$. Пусть Y- матрица $2^n\times(n+L)$, у которой первые n столбцов есть вектора E_2^n, а остальные элементы – свободные переменные, принимающие значения 0 или 1. Определим функцию

$$\Xi((i_{n+1},j_{n+1}),\dots,(i_{n+L},j_{n+L}),f)=\sum_Y g((i_{n+1},j_{n+1}),\dots,(i_{n+L},j_{n+L}),f,Y),$$

где

$$g((i_{n+1},j_{n+1}),\dots,(i_{n+L},j_{n+L}),f,Y)=\prod_{m=1}^{2^n}\prod_{k=n+1}^{n+L}(1-(y_{mk}-(1-y_{mi_k}y_{mj_k}))^2)\cdot\prod_{m=1}^{2^n}(1-(y_{mn+L}-f(y_{m1},\dots,y_{mn}))^2)$$

Функция $g(\cdot)$ равна 1, если выходы каждого элемента совпадают с соответствующими значениями элементов матрицы Y и последний столбец матрицы есть столбец функции f, в противном случае, равна 0.Так как по заданной схеме выходы определяются однозначно, то функция Ξ равна 1, если схема реализует f и 0, в противном случае, т.е. только на одной матрице Y функция g может равняться 1. Определим число

[1] *Носов Михаил Васильевич* — с.н.с. каф. математической теории интеллектуальных систем мех.-мат. ф-та МГУ, e-mail: mvnosov@rambler.ru.

Nosov Michail Vasilevich-senior researcher, Lomonosov Moscow State University, Faculty of Mechanics and Mathematics, Chair of Mathematical Theory of Intellectual Systems.

$$N(f,L)=\sum_{(i_{n+1},j_{n+1}),\ldots,(i_{n+L},j_{n+L})} 2^{m((i_{n+1},j_{n+1}),\ldots,(i_{n+L},j_{n+L}))}\cdot\Xi(\cdot),$$

где

$$m((i_{n+1},j_{n+1}),\ldots,(i_{n+L},j_{n+L}))=$$
$$\sum_{k=1}^{L}(i_{n+k}(n+L)^{2(k-1)}+j_{n+k}(n+L)^{2(k-1)+1})$$

Число $N(f,L)$ – искомое. Зная это число, надо разложить его по степеням 2, каждую степень разложить по основанию $(n+L)$, коэффициенты разложения будут задавать соответствующую схему.

Произведём преобразования

$$\prod_{m=1}^{2^n}\prod_{k=n+1}^{n+L}(1-(y_{mk}-(1-y_{mi_k}y_{mj_k}))^2)=$$

$$\prod_{m=1}^{2^n}\prod_{k=n+1}^{n+L}(y_{mk}+y_{mi_k}y_{mj_k}-2y_{mk}y_{mi_k}y_{mj_k})=$$

$$\prod_{m=1}^{2^n}\prod_{k=n+1}^{n+L}(y_{mk}+(1-2y_{mk})y_{mi_k}y_{mj_k})=$$

$$\prod_{k=n+1}^{n+L}(\sum_{S,S\subseteq\{1,\ldots,2^n\}}\prod_{s\in S}y_{sk}\cdot\prod_{t\notin S}(1-2y_{tk})y_{ti_k}y_{tj_k})=$$

$$\prod_{k=n+1}^{n+L}(\sum_{S,S\subseteq\{1,\ldots,2^n\}}\prod_{s\in S}y_{sk}\cdot\prod_{t\notin S}(-1)^{y_{tk}}y_{ti_k}y_{tj_k})=$$

$$\prod_{k=n+1}^{n+L}(\sum_{S,S\subseteq\{1,\ldots,2^n\}}\prod_{s\in S}y_{sk}\cdot(-1)^{\sum\limits_{t\notin S}y_{tk}}\cdot\prod_{t\notin S}y_{ti_k}y_{tj_k})=$$

$$\prod_{k=n+1}^{n+L}(\sum_{S,S\subseteq\{1,\ldots,2^n\}}\prod_{s\in S}y_{sk}\cdot(-1)^{\sum\limits_{m=1}^{2^n}y_{mk}+|S|}\cdot\prod_{t\notin S}y_{ti_k}y_{tj_k})=$$

$$\prod_{k=n+1}^{n+L}(\sum_{S,S\subseteq\{1,\ldots,2^n\}}(-1)^{||Y_k||}\cdot(-1)^{|S|}\cdot\prod_{s\in S}y_{sk}\cdot\prod_{t\notin S}y_{ti_k}y_{tj_k})=$$

$$(-1)^{||Y||}\prod_{k=n+1}^{n+L}(\sum_{S,S\subseteq\{1,\ldots,2^n\}}\cdot(-1)^{|S|}\cdot\prod_{s\in S}y_{sk}\cdot\prod_{t\notin S}y_{ti_k}y_{tj_k}),$$

где $||\ Y_k\ ||$ – число единиц в k-ом столбце матрицы Y, $||\ Y\ ||$ – число единиц в матрице Y (число единиц в первых n столбцах чётное).

Для удобства введём обозначения

$$u_{i_{n+k}} = 2^{i_{n+k}(n+L)^{2(k-1)}}, \qquad v_{j_{n+k}} = 2^{j_{n+k}(n+L)^{2(k-1)+1}}$$

Тогда

$$N(f,L) = \sum_{Y} \sum_{(i_{n+1},j_{n+1}),\ldots,(i_{n+L},j_{n+L})} (-1)^{||Y||} u_{i_{n+1}} v_{j_{n+1}}, \ldots, u_{i_{n+L}} v_{j_{n+L}} \cdot$$

$$\sum_{S_{n+1},\ldots,S_{n+L} \subseteq \{1,\ldots,2^n\}} ((-1)^{|S_{n+1}|} \cdot \prod_{s \in S_{n+1}} y_{sk} \cdot \prod_{t \notin S_{n+1}} y_{ti_k} y_{tj_k}) \cdot \ldots \cdot ((-1)^{|S_{n+L}|} \cdot$$

$$\prod_{s \in S_{n+L}} y_{sk} \cdot \prod_{t \notin S_{n+L}} y_{ti_k} y_{tj_k}) \cdot \prod_{m=1}^{2^n} (1 - (y_{mn+L} - f(y_{m1}, \ldots, y_{mn}))^2) =$$

$$\sum_{Y} (-1)^{||Y||} \prod_{k=n+1}^{n+L} (\sum_{S_k} \sum_{i_k, j_k} (-1)^{|S_k|} \cdot u_{i_k} v_{j_k} \prod_{s \in S_k} y_{sk} \cdot \prod_{t \notin S_k} y_{ti_k} y_{tj_k}) \cdot$$

$$\prod_{m=1}^{2^n} (1 - (y_{mn+L} - f(y_{m1}, \ldots, y_{mn}))^2) =$$

$$\sum_{Y} (-1)^{||Y||} \prod_{k=n+1}^{n+L} (\sum_{S_k} (-1)^{|S_k|} \cdot \prod_{s \in S_k} y_{sk} \cdot (\sum_{l_k=1}^{k-1} u_{l_k} \prod_{t \notin S_k} y_{tl_k}) (\sum_{l_k=1}^{k-1} v_{l_k} \prod_{t \notin S_k} y_{tl_k})) \cdot$$

$$\prod_{m=1}^{2^n} (1 - (y_{mn+L} - f(y_{m1}, \ldots, y_{mn}))^2)$$

Преобразуем последнее произведение

$$\prod_{m=1}^{2^n} (1 - (y_{mn+L} - f(y_{m1}, \ldots, y_{mn}))^2) =$$

$$\prod_{m=1}^{2^n} ((1 - y_{mn+L}) + (2y_{mn+L} - 1) f(y_{m1}, \ldots, y_{mn})) =$$

$$\sum_{U, U \subseteq \{1,\ldots,2^n\}} \prod_{m, m \notin U} (1 - y_{mn+L}) \cdot \prod_{m \in U} (2y_{mn+L} - 1) \cdot \prod_{m \in U} f(y_{m1}, \ldots, y_{mn}) =$$

$$\sum_{U, U \subseteq \{1,\ldots,2^n\}} ((-1)^{|U|} \prod_{m, m \notin U} (1 - y_{mn+L}) \cdot (-1)^{\sum_{m \in U} y_{mn+L}}) \cdot \prod_{m \in U} f(y_{m1}, \ldots, y_{mn})$$

Окончательно

$$N(f,L)=\sum_{U,U\subseteq\{1,\ldots,2^n\}}(\sum_{Y}(-1)^{||Y||}$$

$$\prod_{k=n+1}^{n+L}(\sum_{S_k,S_k\subseteq\{1,\ldots,2^n\}}(-1)^{|S_k|}\cdot\prod_{s\in S_k}y_{sk}\cdot(\sum_{l_k=1}^{k-1}u_{l_k}\prod_{t\notin S_k}y_{tl_k})(\sum_{l_k=1}^{k-1}v_{l_k}\prod_{t\notin S_k}y_{tl_k}))\cdot$$

$$(-1)^{|U|}\prod_{m,m\notin U}(1-y_{mn+L})\cdot(-1)^{\sum\limits_{m\in U}y_{mn+L}})\cdot\prod_{m\in U}f(y_{m1},\ldots,y_{mn})$$

Наименьшее L, начиная с которого имеет смысл искать схемы, – минимальная сложность схемы, реализующая f, в базисе из штриха Шеффера, поиск минимальной сложности описан в указанной статье.

Вычисление коэффициентов при $\prod\limits_{m\in U}f(y_{m1},\ldots,y_{mn})$ наталкивается на проблему "большой размерности".

Representation of the SFE

Nosov M.V.

In this paper, we obtain a representation of all schemes of functional elements with the same number of elements in the basis from the Schaeffer stroke for a Boolean function in the form of a natural number. Decoding this number allows you to build all the schemes.

Keywords: Boolean function, a scheme of functional elements.

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: 25.12.2024
Дата выхода: 19.01.2025
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).