МИНОБРНАУКИ РОССИИ
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«САРАТОВСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИМЕНИ Н. Г. ЧЕРНЫШЕВСКОГО»
Кафедра дискретной математики и информационных технологий

АВТОМАТНЫЕ МОДЕЛИ БЛОКЧЕЙНА И СМАРТ-КОНТРАКТОВ

АВТОРЕФЕРАТ БАКАЛАВРСКОЙ РАБОТЫ

студента 4 курса 421 группы
направления 09.03.01 — Информатика и вычислительная техника
факультета КНиИТ
Королева Германа Константиновича

Научный руководитель
доцент, к. ф.-м. н. _____ Л. Б. Тяпаев

Заведующий кафедрой
доцент, к. ф.-м. н. _____ Л. Б. Тяпаев

Саратов 2021

ВВЕДЕНИЕ

Технология блокчейн стала широко известна относительно недавно в связи с ростом интереса к криптовалютам. Для того чтобы убедиться в безопасности и работоспособности блокчейн среды требуется её исследовать и, следовательно, требуется построить достоверную модель.

При построении математических моделей блокчейн-среды теоретико-автоматная модель возникает естественным образом, поскольку функционирование блокчейн-среды – это детерминированный процесс, в ходе которого решение о включении или невключении блока в реестр зависит как от предыдущих блоков, так и от времени. Таким образом, процесс изменения содержимого реестра можно описать с помощью понятия «автомат с метками времени» (Т-автомат, ТА). В свою очередь Т-автомат можно свести к более простому понятию – Т-функции.

Вышеизложенное подтверждает актуальность выбранной темы и определяет цель исследования: проверить возможности применения Т-функции в процессе моделирования блокчейн-среды.

В соответствии с целью работы были поставлены следующие задачи.

— Изучить понятия блокчейна, хеш-функции и смарт-контрактов.
— Рассмотреть возможность использования автоматов с метками времени для моделирования блокчейн-среды.
— Рассмотреть возможность сведения автомата с метками времени к Т-функции.
— Разработать приложения для анализа Т-функций.

1 Понятия блокчейна, хеш-функции и смарт-контрактов

Блокчейн – это выстроенная по определённым правилам непрерывная последовательная цепочка блоков (связный список), содержащих информацию. Иными словами это структура данных, использующаяся для создания децентрализованного регистра, содержащего растущий список записей, защищенного от несанкционированных изменений. Данные группируются в блоки, содержащие непосредственно сами данные, а также хеш предыдущего блока, время создания и т.д. Блоки формируют цепь за счет хеша предыдущего блока. Именно такое устройство обеспечивает защищенность и устойчивость данных. На данный момент, самым известным примером блокчейна являются криптовалюты.

Для повышения безопасности, при работе с блокчейном используются хеш-функции.

Хеш-функция, или функция свёртки – функция, осуществляющая преобразование массива входных данных произвольной длины в выходную битовую строку установленной длины, выполняемое определённым алгоритмом. Преобразование, производимое хеш-функцией, называется хешированием. Исходные данные называются входным массивом, «ключом» или «сообщением». Результат преобразования называется «хешем», «хеш-кодом», «хеш-суммой», «сводкой сообщения».

Дадим формальное определение хеш-функции. Пусть $\{0,1\}^m$ – множество всех двоичных строк длины m, $\{0,1\}^*$ – множество всех двоичных строк конечной длины. Тогда хеш-функцией h называется преобразование вида

$$h : \{0,1\}^* \to \{0,1\}^m,$$

где m – разрядность хеш-образа.

Для предоставления больших возможностей, как пользователю, так и разработчику, совместно с технологией блокчейн применяются смарт-контракты.

Смарт-контракт – это компьютерный алгоритм, использующийся для контроля, формирования и предоставления некоторой требуемой информации. В большинстве случаев используется при работе с блокчейном. При работе с криптовалютами под смарт-контрактом подразумевается набор данных или функций, которые находятся в блокчейне и имеют определенный адрес.

2 Теоретико-автоматные модели блокчейн-среды

В процессе конструирования математической модели блокчейн-среды, появляется естественно-логическая необходимость построения теоретико-автоматной модели. Суть действия блокчейн-среды – в процессе определения: резолюция о введении или запрете функционирования блока в реестре находится в зависимости от двух факторов – от предшествующих блоков (логический тип данных – булев) и от времени (тип данных – действительные числа). Получается следующая схема изложения процесса: данное положение реестра можно фиксировать как предшествующее, а состояние реестра тотчас же после запуска в него нового блока – как предстоящее состояние. Описать этот процесс изменения в содержимом реестра исследователи предлагают с помощью понятия «автомат с метками времени» (timed automaton) (TA).

Еще один подход к конструированию, теоретико-автоматных моделей действия блокчейн-среды, точнее функционирования смарт-контрактов в этой среде, принципиально другой, предлагает исследователь В. С. Анашин, на котором основывается данное исследование. Суть его – во взгляде на модель как на автомат с временными метками, при этом физическое время в данном случае представлено не действительными, а 2-адическими числами. Автором обосновывается применение данного подхода рядом причин.

1. Конечный результат модели подразумевает автомат в традиционном понимании данного слова. Но предопределенное автоматом изменение слов выполняется не в форме реестра переходов состояний, а в форме программы без ветвления в виде цепочки стандартных команд процессора, конкретно арифметических и поразрядных логических операций.
2. Так как это все-таки традиционный автомат, то представление его действия можно передать в виде изучения функции, заданной и получающей значения в пространстве целых 2-адических чисел.
3. Новый подход позволяет изучать подобные автоматы, в том числе и процесс функционирования смарт-контрактов в блокчейн-среде, с помощью сформированного аппарата p-адического анализа.
4. 2-адическое и более широкое p-адическое время как релевантная модель физического времени достаточно широко применяется и активно исследуется в течение трех последних десятилетий.

3 Сведение T-автоматной модели блокчейн-среды

В реальной жизни время, разделяющее два следующих один за другим события, не может быть произвольно малым, оно всегда ограничено снизу некоторой величиной. Отсюда следует, что любой временной интервал кратен некоторому минимальному временному интервалу (в предельном случае – планковскому времени) и, таким образом, с точностью до множителя, равного длине этого минимального интервала, является натуральным числом.

Таким образом, для моделирования блокчейн-среды было бы удобно построить аналог ТА, в котором в качестве «исходных» меток времени выступали бы натуральные числа , но, тем не менее, чтобы сохранялась и возможность «предельного перехода» как в целях получения описания поведения всей системы во времени на качественном уровне, так и получения оценок «точности» модели. Поскольку речь идет о предельном переходе, то необходимо задать некоторую метрику на множестве всех натуральных чисел, относительно которой такой предельный переход был бы возможен и относительно которой натуральные числа образовывали бы всюду плотное подмножество подобно тому, как множество $Q_{\geq 0}$ является всюду плотным подмножеством в $R_{\geq 0}$ относительно действительной метрики. Такие метрики существуют: это p-адические метрики.

Рассмотрим t-слова с метками времени из $N_0 = \{0, 1, 2, 3, ...\}$ над конечным алфавитом A, содержащим хотя бы два символа. Без ограничения общности можно считать, что если $p \geq 2$ – это мощность алфавита A, то символами алфавита A являются числа $0, 1, ..., p-1$. Отметим, что такие t-слова представляют собой частный случай так называемых слов с данными, а именно: когда множество данных D совпадает с N_0. Слова с метками времени также представляют собой слова с данными для случая, когда данные лежат в $R_{\geq 0}$. На основе понятия слов с данными естественным образом вводится понятие языка с данными, а также автомата с данными.

Автомат $\mathfrak{A}$ с данными D есть кортеж $\mathfrak{A} = (I, S, F, T, k, \sim, s_0)$, где I, S, F, s_0 – те же что и в определении автомата-определителя;

- k есть натуральное число (называемое числом регистров данных);
- $\sim$ есть отношение эквивалентности конечного индекса, определенное на D^k;
- $T \subseteq S \times D^k \sim \times I \times D^k \sim \times S$ есть конечное множество переходов;

— U есть множество модификаций состояний регистров $udp : D^k \to D^k$, удовлетворяющих следующим ограничениям:

– для любого кортежа

$$(s, g, a) \in S \times D^k \sim \times I$$

существует (единственная модификация регистров $udp \in U$ такая, что если $(s, g, a, udp', g', s') \in T$ для некоторого $udp' \in U$, то $udp' = udp$;

– если (s, g, a, udp, g', s), $(s, g, a, udp, g', s') \in T$, то $s' = s$.

Для любого детерминированного T-автомата, имеющего n таймеров, существует D-автомат с $2n+2$ регистрами, распознающий в точности тот же самый язык. Далее построим автоматы-преобразователи, аппроксимирующие с любой наперед заданной точностью данный детерминированный T-автомат и тем самым сведем задачу моделирования функционирования блокчейн-среды T-автоматами к моделированию «обычными» автоматами с двоичными входами и двоичными выходами. Для этого сначала понадобится ввести новый частный тип D-автоматов, а именно: автоматы с p-адическим временем.

Зафиксируем некоторое простое число p (в контексте данного исследования наиболее важным является случай $p = 2$) и рассматрим в качестве меток данных целые p-адические числа, т. е. элементы пространства Z_p целых p-адических чисел.

Множество Z_p можно рассматривать как множество $W^\infty(A)$ всех бесконечных слов над алфавитом $A = \{0, 1, ..., p-1\}$, символы которого можно считать элементами кольца Z/pZ вычетов по модулю p, т. е. элементами поля из p элементов. На множестве Z_p можно задать операции сложения и умножения с помощью стандартных «школьных» алгоритмов сложения и умножения «в столбик» чисел, представленных в системе счисления с основанием p. Если $p = 2$, то бесконечные бинарные строчки можно мыслить себе как представления чисел в обобщенном обратном двоичном коде.

Множество Z_p является полным компактным метрическим пространством относительно p-адической метрики d_p, которая задается следующим образом: $d_p(a, b) = 1p_i$ тогда и только тогда, когда $a = ...a_{i+1}a_ic_{i-1}...c_0$, $b = ...b_{i+1}b_ic_{i-1}...c_0$ и $a_i \neq b_i$. Абсолютная величина $\|a\|_p$ вводится стандартным образом как расстояние до числа 0 (этому числу соответствует беско-

нечная строчка из одних только нулей): $\|a\|_p = d_p(a, 0)$.

Рассмотрим понятия «приведения по модулю p_n» и «сравнения помодулю p_n» для целых p-адических чисел. Приведение по модулю p_n бесконечного слова в алфавите $A = \{0, 1, ..., p-1\}$ означает переход к конечному начальному отрезку длины n этого бесконечного слова, т. е. $modp^n : W^\infty(A) \to W^n(A)$ есть множество всех слов длины n над алфавитом A. Отметим, что элементы множества $W^n(A)$ естественным образом отождествляются с числами $0, 1, ..., p^{n-1}$, представленными в системе счисления с основанием p, а эти числа, в свою очередь, отождествляются с элементами кольца Z/p^nZ вычетов по модулю p^n. Более того, любое отображение $f_{\mathfrak{A}} : W^\infty(A) \to W^\infty(A)$, задаваемое автоматом-преобразователем $\mathfrak{A}$, входной и выходной алфавиты которого суть A, т. е. $I = Q = A = \{0, 1, ..., p-1\}$ есть функция, определенная на Z_p и принимающая значения в Z_p, которая удовлетворяет p-адическому условию Липшица с константой 1 (и, следовательно, является непрерывной относительно метрики d_p функцией).

Отметим, что каждый из двух типов автоматов: автоматы-определители и автоматы-преобразователи – может быть сведен один к другому. Таким образом, в случае $p = 2$ задачи о T-автоматах и распознаваемых ими языках могут быть сведены к задачам о функциях, удовлетворяющих 2-адическому условию Липшица с константой 1. Такие функции называются в литературе также функциями треугольного вида, двоичными совместимыми функциями, T-функциями.

Сказанное остается в силе и для автоматов, алфавиты которых (входной и выходной) состоят из соответственно 2^n и 2^m символов, поскольку такие автоматы можно рассматривать как автоматы, имеющие n двоичных входов и m двоичных выходов, а значит, как многомерные T-функции. Для T-функций имеется хорошо развитая математическая теория, основанная на 2-адическом анализе и имеющая многочисленные (в первую очередь – криптографические) приложения. Приведем формальное определение автомата с 2-адическим временем.

D-автомат из предыдущего определения назовем автоматом с 2-адическим временем (Z_2-автоматом), если $I = \{0, 1\}$ и $D = Z_2$.

Разумеется, похожим образом можно сформулировать и понятие Z_2-автомата с входным алфавитом из 2^r символов, т. е. Z_2-автомата с r двоич-

ными входами. Язык, распознаваемый Z_2-автоматом, определяется обычным образом на основе определения автомата с 2-адическим временем.

Любой T-автомат можно рассматривать как «автомат с двумя входами»: временным и алфавитным, где на каждом такте работы подается на алфавитный вход очередной символ входного слова, а на временной вход – действительное число, служащее меткой времени этого входного символа. С этой точки зрения Z_2-автомат тоже имеет два входа; при этом на алфавитный вход подается символ входного алфавита, т. е. 0 или 1, а на временной вход – метка времени, т. е. целое 2-адическое число. Все t-слова могут быть равномерно приближены словами с 2-адическими метками времени (далее – Z_2-словами) в следующем смысле. Вначале все символы входного алфавита T-автомата пронумеруем и запишем в виде двоичных представлений соответствующих чисел. Таким образом, можно считать, что на алфавитный вход автомата всегда подается r бинарных последовательностей, где r – число двоичных разрядов, необходимых для записи всех символов входного алфавита.

Далее зафиксируем любое действительное $\varepsilon > 0$ и выбирает рациональные числа $z_i(w)$, представимые в виде простых несократимых дробей с нечетными знаменателями (все эти рациональные числа лежат в Z_2) так, чтобы $|\tau_i(w) - z_i(w)| < \varepsilon$, где $\tau_i(w)$ есть i-я метка времени в t-слове w. Такой выбор всегда можно сделать, например, следующим образом. Представим

$$\tau_i(w) = \lfloor \tau_i(w) \rfloor + (\tau_i(w) - \lfloor \tau_i(w) \rfloor),$$

где $\lfloor \tau_i(w) \rfloor$ есть целая (с недостатком) часть числа $\tau_i(w)$. Выберем $h \in N$ таким, чтобы $1/3^h < \varepsilon$, запишем дробную часть $(\tau_i(w) - \lfloor \tau_i(w) \rfloor)$ числа $\tau_i(w)$ в троичной системе счисления с точностью до h троичных разрядов после запятой. Тогда эта дробная часть есть число вида $c/3^h$, где $c \in \{0, 1, ..., 3^{h-1}\}$, и, следовательно, является целым 2-адическим числом. Прибавляя к полученному таким образом числу целую (с недостатком) часть $\tau_i(w) - \lfloor \tau_i(w) \rfloor$ числа $\tau_i(w)$, получаем целое 2-адическое число $z_i(w)$. В этом смысле каждое t-слово $w = ((a_i, \tau_i))_{i=0}^{\infty}$ приближается с точностью не хуже чем ε словом $((a_i, z_i(w)))_{i=0}^{\infty}$, которое является входным Z_2-словом для Z_2-автомата с r алфавитными входами, причем алфавит каждого алфавитного входа бинарный.

Далее все Z_2-слова могут быть равномерно приближены Z_2-словами с метками времени из N_0 (и даже из $Z/2^hZ$) с любой наперед заданной 2-

адической точностью $1/2^h$. Действительно, для этого достаточно каждую из 2-адических меток времени в каждом Z_2-слове привести по модулю 2^h. Таким образом на основе вышеописанной процедуры «аппроксимации» t-автомата Z_2-автоматом можно построить последовательность Z_2-автоматов $\mathfrak{Y}_h$ сметками времени из $Z/2^hZ$, $h = 1, 2, 3, ...$, аппроксимирующих в вышеуказанном смысле исходный t-автомат.

Используя описанную выше процедуру построения автомата-преобразователя на основе данного автомата-определителя, можно любому D-автомату сопоставить детерминированную функцию с метками времени, считая, например, что i-й символ выходного слова имеет ту же метку времени, что и i-й символ соответствующего ему входного слова. Таким образом, на основе данного Z_2-автомата из определения автомата с 2-адическим временем можно построить детерминированную функцию с метками времени из Z_2, полагая i-й выходной символ равным 1, если автомат находится в принимающем состоянии (т. е. в состоянии из множества F), и 0 в противном случае.

Наконец, этим способом каждому из построенных выше аппроксимирующих автоматов $\mathfrak{Y}_h$ можно сопоставить детерминированную функцию с метками времени из $Z/2^hZ$. Итак, для данного T-автомата построена последовательность аппроксимирующих его (в вышеописанном смысле) T-функций, т. е. «обычных» автоматов-преобразователей, имеющих $r + h$ двоичных входов и $h + 1$ двоичный выход.

4 Разработка приложения

В ходе работы было разработано приложение для анализа T-функций при работе с 2-адическими числами, благодаря которому можно оценивать полученные результаты. При создании программы использовались стандартные средства языка Python (вер. 3.8.2), а также библиотеки pandas для сохранения значений в файле формата «.xlsx» и matplotlib для создания графика исследуемой функции в формате «.png».

Программа состоит из трех файлов: padic.py, modp.py и main.py. В первом файле содержится ряд классов для работы с p-адическими числами. Во втором – класс для более удобной работы с остатком от деления. В третьем – несколько необходимых функций, в том числе исследуемая T-функция.

Рассмотрим содержимое файла padic.py. PAdic – базовый класс, который описывает принципы взаимодействия классов-наследников, переопределяя базовые операции – сложение, вычитание, умножение, отрицание, побитовое «И» и побитовое «ИЛИ». Кроме этого, он содержит функцию get, которая позволяет получить p-адическое число с заданной точностью. Класс PAdicConst описывает p-адическое число для дальнейших преобразований. В свою очередь, PAdicAdd необходим для получение результата сложения двух p-адических чисел. PAdicNeg – результат отрицания p-адического числа. PAdicMul – итог умножения двух p-адических чисел. PAdicAnd и PAdicOr отвечают соответственно за операции побитового «И» и побитового «ИЛИ». Каждый из классов-наследников хранит необходимые числа для получения очередного требуемого бита p-адического числа.

Рассмотрим содержимое файла modp.py. Класс modp используется при работе с p-адическими числами и нужен, преимущественно для более удобного вычисления остатка от деления и дальнейшей работы с ним.

Рассмотрим содержимое файла main.py. Это основная часть программы, результатом работы которой становятся файл в формате «.xlsx» с точками функции, которая описывается в function(x), и график этой функции в формате «.png».

Поскольку работа осуществлялась с 2-адическими числами, которые по своей сути являются бесконечной последовательностью, для отображения

результата на графике x и $f(x)$ применялась формула

$$l_k(x) = (\frac{x\ mod\ 2^k}{2^k}, \frac{f(x)\ mod\ 2^k}{2^k}).$$

Она позволяет отобразить результаты работы функции в единичном квадрате на плоскости. В данной формуле k – точность, количество младших разрядов, с которыми производится работа. При запуске приложение позволяет пользователю задать желаемую точность. Работоспособность приложения была проверена на 3 различных функциях. Первая T-функция,

$$f(x) = 2 * x + 1,$$

проверялась с точностью $k = 4, 8, 16$. Вторая T-функция,

$$f(x) = (x * 1664525) + x + 1013904223,$$

проверялась с точностью $k = 8, 12, 16$. Третья T-функция – функция Климова-Шамира.

$$f(x) = x + (x^2 ORC)$$

Данная функция обладает свойством равномерного распределения пар $(x, f(x))$, если константа C сравнима по модулю 8 с 5 или 7. Для проверки работы программа использовалась константа $C = -131065$. Результаты запуска программы для точностей $k = 8, 16$ приведены на Рис. 1, 2, соответственно.

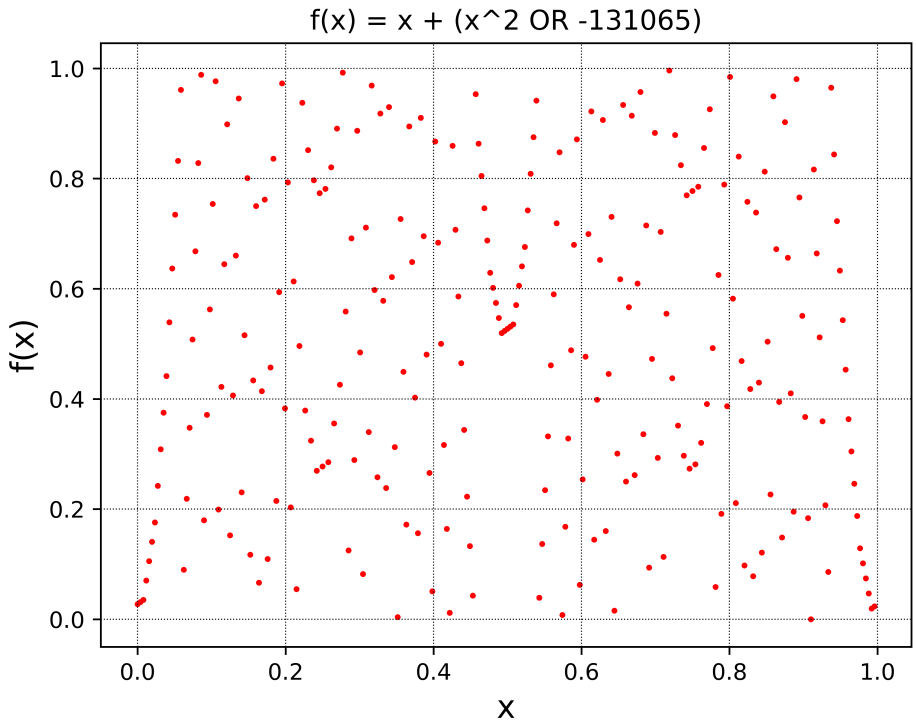

Рисунок 1 – График функции $f(x) = x + (x^2 ORC)$ для точности $k = 8$.

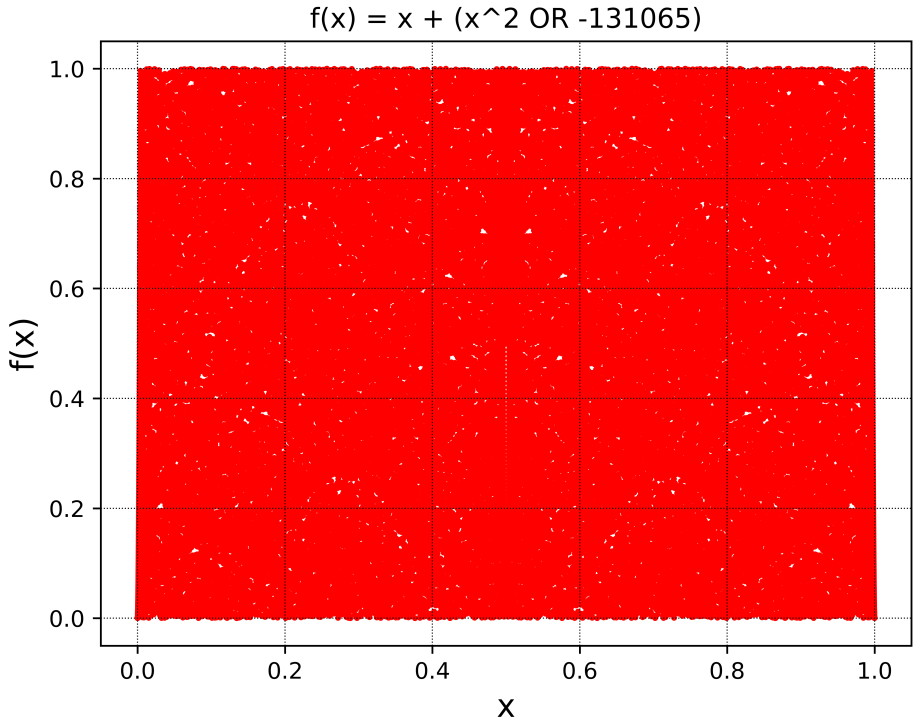

Рисунок 2 – График функции $f(x) = x + (x^2 ORC)$ для точности $k = 16$.

На основании данных графиков, можно сделать вывод, что распределение точек в среднем равномерное, однако если взять точность $k = 18$, то

график будет выглядеть несколько иначе (Рис. 3).

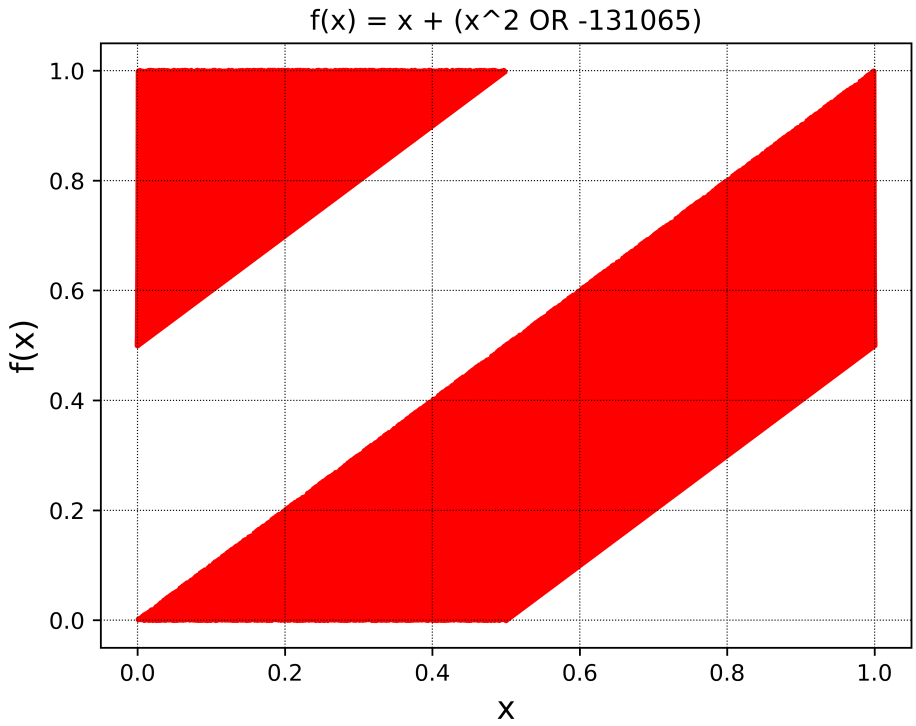

Рисунок 3 – График функции $f(x) = x + (x^2 ORC)$ для точности $k = 18$.

График функции Климова-Шамира обладает свойством равномерного распределения пар, при константе C по модулю 8 сравнимой с 5 или 7, однако в данном случае, поскольку мы работаем с 2-адическими числами, при увеличении точности у константы $C = -131065$ в 2-адическом представлении в старших разрядах появляются 1, из-за чего получается такой результат. Таким образом, можно сделать вывод, что T-функцию Климова-Шамира можно скомпрометировать.

ЗАКЛЮЧЕНИЕ

В ходе работы был изучен теоретический материал по блокчейну, хеш-функциям и смарт-контрактам, рассмотрены возможности использования T-автоматов для моделирования блокчейн-среды и их сведения к более простым для реализации T-функциям, разработано приложение для работы с p-адическими числами, а также была проверена работоспособность этого приложения на ряде функций.

Построение графиков T-функций – есть инструмент исследования функций на предмет распределения пар при различных значениях точности k. T-функции могут быть реализованы в виде программ без ветвления, выполненных как последовательности стандартных команд любого процессора, что позволяет надеяться на относительную простоту их программной реализации и высокое быстродействие соответствующих программ.

Результаты исследования T-функций приводят к следующим выводам.

— Есть вероятность скомпрометировать T-функцию.

— Нет связи между свойством транзитивности функции и равномерным распределением точек графика.

Поэтому, поиск подходящих T-функций - это актуальная задача, как для использования при моделировании блокчейн-среды, так и для применения в криптографии.