ISSN 1816-0301 (Print)
ISSN 2617-6963 (Online)

ИНФОРМАТИКА

INFORMATICS

ТОМ 19
VOL.

1 | 2022

ОТ РЕДАКЦИИ

В журнале «Информатика» публикуются оригинальные и обзорные статьи, описывающие результаты фундаментальных и прикладных исследований специалистов академического и вузовского профиля в области информатики и информационных технологий.

Основной целью журнала является публикация наиболее значимых новых результатов в указанной области. Приветствуются статьи, описывающие заключительные результаты научных проектов и диссертационных исследований, открывающие новые направления исследований, которые находятся на стыке информатики и других наук.

Журнал рассчитан на широкий круг специалистов в области информатики и информационных технологий.

Основные разделы журнала:

- биоинформатика;
- математическое моделирование;
- защита информации и надежность систем;
- информационные технологии;
- логическое проектирование;
- обработка сигналов, изображений, речи, текста и распознавание образов;
- автоматизация проектирования;
- интеллектуальные системы.

Префикс DOI: 10.37661

Индексирование:

Высшей аттестационной комиссией Республики Беларусь журнал «Информатика» был включен в список научных изданий для опубликования результатов диссертационных исследований.

В декабре 2017 г. включен в базу данных Российского индекса научного цитирования (РИНЦ). С помощью инструментов и сервисов, доступных на платформе eLIBRARY (раздел «Личный кабинет»), можно самостоятельно корректировать список своих публикаций и цитирований в РИНЦ.

В июле 2017 г. включен в базу журналов открытого доступа Directory of Open Access Journals (DOAJ).

С помощью поисковых систем Google Scholar, WorldCat, Соционет можно получить свободный доступ к полному тексту научных публикаций журнала.

Адрес редакции:
ул. Сурганова, 6, к. 305, г. Минск, 220012, Беларусь
Тел. +375 (017) 351 26 22

Editorial address:
Surganova str., 6, of. 305, Minsk, 220012, Belarus
Phone +375 (017) 351 26 22

E-mail: rio@newman.bas-net.by
https://inf.grid.by/jour

THE EDITOR'S NOTE

The journal «Informatics» is a scientific publication in computer sciences and information technologies which reviews the results in basic and applied research of scientists from the universities and scientific centers.

The journal focuses on the most significant and modern papers of research projects results and PhD/DSc thesis in computer sciences.

The journal is edited for the specialists in IT and computer sciences research and application.

The main sections of the journal:

- bioinformatics;
- mathematical modeling;
- information protection and system reliability;
- information technology;
- logical design;
- signal, image, speech, text processing and pattern recognition;
- computer-aided design;
- artificial intelligence methods.

DOI Prefix: 10.37661

Indexation:

the journal «Informatics» is in the list of scientific publications recommended by the Higher Attestation Commission of the Republic of Belarus for scientists to publish the results of PhD/DSc research.

In December 2017 the journal was included in the database of the Russian Science Citation Index (RISC) and provides free access to reviewed electronic scientific paper, improving scientific information traffic and also raising quotation of works of the authors (please use https://elibrary.ru or section for authors https://elibrary.ru_author_tools).

In July 2017 included in the database of open access journals Directory of Open Access Journals (DOAJ).

Using the Google Scholar, WorldCat, Соционет search engine, you can get free access to full text of scientific publications of magazine.

ISSN 1816-0301 (Print)
ISSN 2617-6963 (Online)

ОБЪЕДИНЕННЫЙ ИНСТИТУТ ПРОБЛЕМ ИНФОРМАТИКИ
НАЦИОНАЛЬНОЙ АКАДЕМИИ НАУК БЕЛАРУСИ

ИНФОРМАТИКА

Informatika

Том 19, № 1, январь-март 2022

Ежеквартальный научный журнал

Издается с января 2004 г.

Учредитель и издатель – Объединенный институт проблем информатики
Национальной академии наук Беларуси (ОИПИ НАН Беларуси)

ИНФОРМАТИКА

Том 19, № 1, январь-март 2022

Ответственный за выпуск *Мойсейчик Светлана Сергеевна*
Редактор *Гончаренко Галина Борисовна*
Корректор *Михайлова Анна Антоновна*
Компьютерная верстка *Бутевич Ольга Борисовна*

Сдано в набор 18.02.2022. Подписано в печать 17.03.2022. Формат 60×84 1/8. Бумага офсетная. Гарнитура Таймс. Ризография. Усл. печ. л. 12,5. Уч.-изд. л. 12,8. Тираж 40 экз. Заказ 1.

Государственное научное учреждение «Объединенный институт проблем информатики Национальной академии наук Беларуси».
Свидетельство о государственной регистрации издателя, изготовителя, распространителя печатных изданий № 1/274 от 04.04.2014. ЛП № 02330/444 от 18.12.13. Ул. Сурганова, 6, 220012, Минск, Беларусь.

ISSN 1816-0301 (Print)
ISSN 2617-6963 (Online)

THE UNITED INSTITUTE OF INFORMATICS PROBLEMS
OF THE NATIONAL ACADEMY OF SCIENCES OF BELARUS

INFORMATICS

Vol. 19, no. 1, January-March 2022

Published quarterly

Issued since January 2004

Founder and publisher – the United Institute of Informatics Problems
of the National Academy of Sciences of Belarus (UIIP NASB)

INFORMATICS

Vol. 19, no. 1, January-March 2022

Issue Head *Sviatlana S. Maiseichyk*
Editor *Halina B. Hancharenka*
Corrector *Hanna A. Mikhailava*
Computer Imposition *Volha B. Butsevich*

Sent for press 18.02.2022. Output 17.03.2022. Format 60×84 1/8. Offset paper. Headset Times. Riesography. Printed sheets 12,5. Publisher’s signatures 12,8. Circulation 40 copies. Order 1.

State Scientific Institution "The United Institute of Informatics Problems of the National Academy of Sciences of Belarus".
Certificate on the state registration of the publisher, manufacturer, distributor of printing editions no. 1/274 dated 04.04.2014. License for the press no. 02330/444 dated 18.12.13.
6, Surganov Str., 220012, Minsk, Belarus.

ISSN 1816-0301 (Print)
ISSN 2617-6963 (Online)

СОДЕРЖАНИЕ

ISSN 1816-0301 (Print)
ISSN 2617-6963 (Online)

CONTENTS

ЛОГИЧЕСКОЕ ПРОЕКТИРОВАНИЕ
LOGICAL DESIGN

УДК 519.711
https://doi.org/10.37661/1816-0301-2022-19-1-7-18

Оригинальная статья
Original Paper

Синтез комбинационных схем с помощью алгебраической декомпозиции булевых функций

Ю. В. Поттосин

Объединенный институт проблем информатики
Национальной академии наук Беларуси,
ул. Сурганова, 6, Минск, 220012, Беларусь
✉*E-mail: pott@newman.bas-net.by*

Аннотация

Цели. Рассматривается задача синтеза комбинационных схем в базисе двухвходовых логических элементов, в качестве которых выступают элементы И, ИЛИ, И-НЕ и ИЛИ-НЕ. Целью работы является исследование возможности применения алгебраической декомпозиции булевых функций (в англоязычной литературе bi-decomposition) для синтеза комбинационных схем.

Методы. Используемый метод алгебраической декомпозиции сводится к поиску в графе двухблочного взвешенного покрытия полными двудольными подграфами (бикликами).

Результаты. Исходная булева функция задается двумя троичными матрицами, одна из которых представляет собой область булева пространства аргументов, где функция имеет значение 1, а другая – область булева пространства, где функция имеет значение 0. Рассматривается граф ортогональности строк троичных матриц, представляющих заданную булеву функцию. Описан способ получения двухблочного взвешенного покрытия бикликами графа ортогональности строк троичных матриц. Всем бикликам из получаемого покрытия в качестве веса определенным образом приписывается некоторое множество переменных, представляющих собой аргументы заданной функции. Каждая из этих биклик определяет булеву функцию, аргументами которой являются приписанные к биклике переменные. Полученные таким образом функции составляют разложение исходной функции.

Заключение. Процесс синтеза комбинационной схемы заключается в последовательном применении алгебраической декомпозиции к получаемым функциям. Предлагаемый метод позволяет строить схемы с малой задержкой.

Ключевые слова: синтез комбинационных схем, булева функция, декомпозиция булевой функции, троичная матрица, полный двудольный граф, биклика, двухблочное покрытие

Для цитирования. Поттосин, Ю. В. Синтез комбинационных схем с помощью алгебраической декомпозиции булевых функций / Ю. В. Поттосин // Информатика. – 2022. − Т. 19, № 1. – С. 7–18.
https://doi.org/10.37661/1816-0301-2022-19-1-7-18

Конфликт интересов. Автор заявляет об отсутствии конфликта интересов.

Поступила в редакцию | Received 22.11.2021
Подписана в печать | Accepted 14.12.2021
Опубликована | Published 29.03.2022

Synthesis of combinational circuits by means of bi-decomposition of Boolean functions

Yuri V. Pottosin

The United Institute of Informatics Problems
of the National Academy of Sciences of Belarus,
st. Surganova, 6, Minsk, 220012, Belarus
✉*E-mail: pott@newman.bas-net.by*

Abstract

Objectives. The problem of synthesis of combinational circuits in the basis of two-input gates is considered. Those gates are AND, OR, NAND and NOR. The objective of the paper is to investigate the possibilities of application of bi-decomposition of Boolean functions to the synthesis of combinational circuits.

Methods. The method for bi-decomposition is reduced to the search in a graph for a weighted two-block cover with complete bipartite subgraphs (bi-cliques).

Results. The initial Boolean function is given as two ternary matrices, one of which represents the domain of Boolean space where the function has the value 1, and the other is the domain of Boolean space where the function has the value 0. The orthogonality graph of rows of ternary matrices representing the given function is considered. The method for two-bi-clique covering the orthogonality graph of rows of ternary matrices is described. Every bi-clique in the obtained cover is assigned in a certain way with a set of variables that are the arguments of the function. This set is the weight of the bi-clique. Each of those bi-cliques defines a Boolean function whose arguments are the variables assigned to it. The functions obtained in such a way constitute the required decomposition.

Conclusion. The process of synthesis of a combinational circuit consists of a successive application of bi-decomposition to obtained functions. The suggested method allows obtaining the circuits with short delay.

Keywords: synthesis of combinational circuits, Boolean function, decomposition of Boolean functions, ternary matrix, complete bipartite graph, bi-clique, two-block cover

For citation. Pottosin Yu. V. *Synthesis of combinational circuits by means of bi-decomposition of Boolean functions*. Informatika [*Informatics*], 2022, vol. 19, no. 1, pp. 7−18 (In Russ.).
https://doi.org/10.37661/1816-0301-2022-19-1-7-18

Conflict of interest. The author declare of no conflict of interest.

Введение. Задача алгебраической декомпозиции (в англоязычной литературе bi-decomposition) ставится следующим образом. Для заданной булевой функции $y = f(\mathbf{x})$, где компонентами вектора $\mathbf{x} = (x_1, x_2, ..., x_n)$ являются булевы переменные, составляющие множество X, требуется найти суперпозицию $f(\mathbf{x}) = \varphi(g_1(\mathbf{z}_1), g_2(\mathbf{z}_2))$, где компонентами векторов $\mathbf{z}_1$ и $\mathbf{z}_2$ являются переменные из множеств $Z_1 \subset X$ и $Z_2 \subset X$ соответственно. Вид функции φ от двух переменных также задан. Это может быть любая из десяти булевых функций, существенно зависящих от обеих переменных и представляемых операциями алгебры логики. Обычно множества Z_1 и Z_2 заданы и $Z_1 \cap Z_2 = \varnothing$. Такая декомпозиция называется *разделительной* в отличие от *неразделительной* декомпозиции, где условие $Z_1 \cap Z_2 = \varnothing$ необязательно, но при этом на мощности множеств Z_1 и Z_2 могут быть наложены ограничения.

Известны примеры применения методов алгебраической декомпозиции для повышения быстродействия схем [1, 2] и при синтезе схем на базе программируемой вентильной матрицы (FPGA) [3]. Задача алгебраической декомпозиции при функции φ, выражаемой операцией сложения по модулю 2, при заданном разбиении (Z_1, Z_2) рассматривается в работе [4], где для ее решения предлагается использовать логические уравнения. Вероятность существования какой-либо декомпозиции для полностью определенных булевых функций весьма низка, но дело обстоит по-другому, когда рассматриваемые функции являются не полностью определенными (частичными), особенно когда они определены только на небольшой части булева пространства аргументов. Поэтому в литературе основное внимание уделялось декомпозиции (в том числе алгебраической) частичных булевых функций. Такой случай разделительной алгебраической

декомпозиции при заданном разбиении (Z_1, Z_2) подробно исследован в работе [5]. Эвристический метод алгебраической декомпозиции (разделительной или неразделительной) частичных булевых функций при незаданном разбиении (Z_1, Z_2), когда к решению предъявляется только требование, чтобы числа аргументов функций g_1 и g_2 были как можно меньше, чем число аргументов исходной функции f, описан в статье [6]. Тот же метод применим и для полностью определенных функций, однако, как уже было сказано, очень мала вероятность того, что будет выполнено указанное требование. Вместе с тем, если функция φ относится к классу нелинейных функций, то функции g_1 и g_2 оказываются проще функции f в том смысле, что степень их зависимости от некоторых переменных может быть меньше, чем у функции f. Данный параметр рассматривался в статье [7]. Под степенью зависимости функции f от переменной x_i здесь понимается число пар значений ($\mathbf{x}'$, $\mathbf{x}''$) вектора $\mathbf{x}$ с различными значениями i-й компоненты, для которых $f(\mathbf{x}') \neq f(\mathbf{x}'')$. Кроме того, если какая-то из функций g_i (i = 1, 2) оказалась с тем же числом аргументов, что и полностью определенная функция f, то эта функция g_i в любом случае будет не полностью определенной, что увеличивает вероятность ее разложимости. Далее предлагается метод синтеза комбинационных схем в базисе двухвходовых элементов, реализующих нелинейные функции. Имеются в виду базисы элементов И-НЕ, ИЛИ-НЕ, а также базис элементов И, ИЛИ при доступных инверсиях переменных. Метод основан на последовательном применении алгебраического разложения к получаемым функциям способом, описанным в статье [6].

Предлагаемый подход. Предполагается, что булева функция $f(\mathbf{x})$, полностью или не полностью определенная, задана двумя множествами: областью M^1 булева пространства, где функция имеет значение 1, и областью M^0 булева пространства, где она имеет значение 0. Эти множества будем задавать соответственно троичными матрицами $\mathbf{M}^1$ и $\mathbf{M}^0$, строки которых представляют собой интервалы из областей M^1 и M^0, а столбцы соответствуют аргументам $x_1, x_2, \ldots, x_n$ заданной функции.

Рассмотрим полный двудольный граф $G = (V^1, V^0, E)$, где вершины из множества V^1 соответствуют строкам матрицы $\mathbf{M}^1$, вершины из множества V^0 – строкам матрицы $\mathbf{M}^0$, а ребрами являются все пары вершин v^1v^0 ($v^1 \in V^1$, $v^0 \in V^0$), которым соответствуют ортогональные строки матриц. Два троичных вектора ортогональны по компоненте x_i, если в одном из них $x_i = 1$, а в другом $x_i = 0$ [8]. Естественно, любая вектор-строка $\mathbf{m}^1$ матрицы $\mathbf{M}^1$ ортогональна любой вектору-строке $\mathbf{m}^0$ матрицы $\mathbf{M}^0$. Поэтому двудольный граф G является полным.

Каждому ребру v^1v^0 ($v^1 \in V^1$, $v^0 \in V^0$) графа G припишем элементарную дизъюнкцию $x_i \vee x_j \vee \ldots \vee x_k$ аргументов заданной функции, если векторы-строки $\mathbf{m}^1$ и $\mathbf{m}^0$ матриц $\mathbf{M}^1$ и $\mathbf{M}^0$, соответствующие вершинам v^1 и v^0 (концам данного ребра), ортогональны по компонентам $x_i, x_j, \ldots, x_k$. Полному двудольному подграфу, или *биклике*, графа G припишем конъюнктивную нормальную форму (КНФ) с элементарными дизъюнкциями, приписанными ребрам, которые принадлежат данной биклике. После удаления возможных поглощаемых элементарных дизъюнкций преобразуем полученную КНФ, раскрыв скобки, в дизъюнктивную нормальную форму (ДНФ). Переменные, составляющие элементарную конъюнкцию минимального ранга в полученной ДНФ, припишем соответствующей биклике.

Пусть требуется выразить заданную (в общем случае не полностью определенную) функцию $f(\mathbf{x})$ как $f(\mathbf{x}) \prec \varphi(g_1(\mathbf{z}_1), g_2(\mathbf{z}_2))$, где φ – булева функция от двух переменных g_1 и g_2, которые являются функциями соответственно от векторных переменных $\mathbf{z}_1$ и $\mathbf{z}_2$, представляющих части вектора $\mathbf{x}$, а символ $\prec$ обозначает отношение реализации. Функция φ, частичная или полностью определенная, реализует частичную функцию f, если значения функции φ совпадают со значениями функции f везде, где они определены [8]. Далее удобно рассматривать отношение равенства функций как частный случай отношения реализации, поэтому для отношения реализации также будем использовать символ «=».

Функции g_1 и g_2 построим следующим образом. В графе G выделим две биклики $B_1 = (V_1^1, V_1^0, E_1)$ и $B_2 = (V_2^1, V_2^0, E_2)$ так, чтобы любое ребро графа G присутствовало хотя бы в одном из множеств E_1 или E_2, т. е. биклики B_1 и B_2 должны покрывать своими ребрами все множество E ребер графа G. Биклики B_1 и B_2 достаточно задать парами множеств (V_1^1, V_1^0) и (V_2^1, V_2^0), так как в биклике каждая вершина из одной доли связана ребрами со всеми вершинами другой доли.

Аргументами функции g_i (i = 1, 2) являются переменные, приписанные биклике B_i. Множество M_i^1 значений векторной переменной $\mathbf{z}_i$, где функция g_i имеет значение 1, составляют части векторов из M^1 или M^0 (в зависимости от вида функции φ), соответствующих вершинам из множества V_i^1. Части этих векторов определяются переменными, приписанными биклике B_i, т. е. эти переменные являются компонентами вектора $\mathbf{z}_i$. Аналогично формируется множество M_i^0 из частей векторов, соответствующих вершинам из множества V_i^0. Таким образом, каждому вектору из M^1 или M^0 соответствует пара значений функций g_1 и g_2. Если эта пара соответствует вектору из множества M^1, то она является элементом множества $M_\varphi{}^1$, где функция φ имеет значение 1. Если пара соответствует вектору из M^0, то она является элементом множества $M_\varphi{}^0$. Так будет задана функция φ. Заметим, что пары $(V_1{}^1, V_1{}^0)$ и $(V_2{}^1, V_2{}^0)$ следует считать упорядоченными, поскольку они связаны со значениями функций g_1 и g_2.

Описываемый метод предполагает дальнейшее подобное разложение функций g_1 и g_2 и последующих функций до получения функций от двух переменных из множества $X = \{x_1, x_2, ..., x_n\}$ аргументов заданной функции.

Получение покрытия графа *G* двумя бикликами. В таблице показано, какие значения должны иметь функции g_1 и g_2 при определенных значениях функции φ и при разных видах этой функции. Равенство $V_1{}^1 = V_2{}^1 = V^1$ должно выполняться для операции И, $V_1{}^0 = V_2{}^0 = V^0$ – для операции ИЛИ, $V_1{}^1 = V_2{}^1 = V^0$ – для операции И-НЕ и $V_1{}^0 = V_2{}^0 = V^1$ – для операции ИЛИ-НЕ.

Значения функций g_1 и g_2

The values of the functions g_1 and g_2

И *AND*	ИЛИ *OR*	И-НЕ *NAND*	ИЛИ-НЕ *NOR*
φ g_1 g_2	φ g_1 g_2	φ g_1 g_2	φ g_1 g_2
1 1 1	0 0 0	0 1 1	1 0 0
0 – 0	1 – 1	1 – 0	0 – 1
0 0 –	1 1 –	1 0 –	0 1 –

Таким образом, одна из долей биклики всегда определена видом функции φ как одна из долей полного двудольного графа G и присутствует в обеих бикликах. Другие доли биклик B_1 и B_2 образуются как блоки разбиения другой доли графа G. Например, если $V_1{}^0 = V_2{}^0 = V^1$, то $B_1 = (V_1{}^1, V^1)$ и $B_2 = (V_2{}^1, V^1)$, где $V_1{}^1 \cup V_2{}^1 = V^0$ и $V_1{}^1 \cap V_2{}^1 = \varnothing$.

Исходной информацией для получения искомого покрытия служит множество *звездных графов*, которые являются подграфами графа G. Звездным графом, или *звездой*, называется полный двудольный граф $K_{1,n}$ [9]. Одноэлементная его доля представляет собой центр звезды. В рассматриваемом случае упомянутое множество – это множество всех биклик, у которых одной долей является одноэлементное множество с вершиной $v \in V^0$, а другой – множество V^1 или у которых одной долей является одноэлементное множество с вершиной $v \in V^1$, а другой – множество V^0 двудольного графа G. Назовем их *звездными бикликами*.

Как было сказано выше, каждой биклике приписывается КНФ, которая преобразуется в ДНФ. Из ДНФ выберем элементарную конъюнкцию K минимального ранга и вместо ДНФ и КНФ припишем соответствующей звездной биклике B_i множество переменных X_i из конъюнкции K. Выберем две звездные биклики B_i и B_j, у которых пересечение $X_i \cap X_j$ имеет минимальную мощность среди всех пар рассматриваемых звездных биклик. Если таких вариантов несколько, то отдадим предпочтение множествам X_i и X_j максимальной мощности. Естественно, желателен вариант $X_i \cap X_j = \varnothing$. Примем пару (B_i, B_j) за начальное значение пары биклик, которая должна покрывать граф G, и обозначим ее (B_1, B_2).

Дальнейший процесс представляет собой последовательное расширение тех долей биклик B_1 и B_2, которые в начальных значениях были одноэлементными, за счет вершин, являющихся центрами рассматриваемых звездных биклик. Соответственно меняются множества X_1 и X_2. Пусть, например, $B_1 = (V_1{}^1, V_1{}^0)$, $B_2 = (V_2{}^1, V_2{}^0)$ и $V_1{}^1 \cup V_2{}^1 = V^0$, а множество V' состоит из вершин графа G, которые не принадлежат ни V^0, ни одному из $V_1{}^0$ и $V_2{}^0$. Выбираются верши-

на $v_k \in V'$, являющаяся центром некоторой звездной биклики B_k, и множество V_i^0 ($i \in 1, 2$), такие, что мощность множества $X_i \cup X_k$ отличается от мощности множества X_i или X_k на минимальную величину. Множество V_i^0 меняется на $V_i^0 \cup \{v_k\}$, а вершина v_k удаляется из V'. Процесс заканчивается, когда множество V' окажется пустым. Пара (B_1, B_2) представит искомое покрытие.

Пример. Пусть требуется построить логическую сеть из элементов И-НЕ, реализующую полностью определенную булеву функцию $f(x_1, x_2, x_3, x_4, x_5)$, которая представлена следующими матрицами (используется сквозная нумерация строк):

$$\mathbf{M}^1 = \begin{matrix} x_1 & x_2 & x_3 & x_4 & x_5 \end{matrix}\begin{bmatrix} - & - & 1 & 1 & - \\ 0 & - & 1 & - & - \\ - & 1 & - & 1 & - \\ 0 & 0 & - & - & - \\ - & 1 & 0 & - & 0 \\ 1 & 1 & - & - & 1 \\ 1 & - & 0 & 0 & 1 \end{bmatrix}\begin{matrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \end{matrix}, \quad \mathbf{M}^0 = \begin{matrix} x_1 & x_2 & x_3 & x_4 & x_5 \end{matrix}\begin{bmatrix} 0 & 1 & 0 & 0 & 1 \\ 1 & - & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 & - \\ 1 & 0 & 1 & 0 & - \end{bmatrix}\begin{matrix} 8 \\ 9 \\ 10 \\ 11 \end{matrix}.$$

Для сокращения размеров рассматриваемых двудольных графов желательно представлять область определения функции минимальным количеством интервалов. Двудольный граф $G = (V^1, V^0, E)$ представим матрицей, подобной матрице смежности:

$$\mathbf{G} = \begin{matrix} 8 & 9 & 10 & 11 \end{matrix}\begin{bmatrix} x_3 \vee x_4 & x_4 & x_3 & x_4 \\ x_3 & x_1 & x_1 \vee x_3 & x_1 \\ x_4 & x_4 & x_2 & x_2 \vee x_4 \\ x_2 & x_1 & x_1 & x_1 \\ x_5 & x_3 & x_2 & x_2 \vee x_3 \\ x_1 & x_5 & x_2 & x_2 \\ x_1 & x_3 \vee x_5 & x_4 & x_3 \end{bmatrix}\begin{matrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \end{matrix}.$$

Строки матрицы **G** соответствуют вершинам из множества $V^1 = \{v_1, v_2, v_3, v_4, v_5, v_6, v_7\}$ (строкам матрицы $\mathbf{M}^1$), а столбцы – вершинам из множества $V^0 = \{v_8, v_9, v_{10}, v_{11}\}$ (строкам матрицы $\mathbf{M}^0$). На пересечении i-й строки и j-го столбца находится элементарная дизъюнкция или одиночная переменная, приписанные ребру $v_i v_j$.

Биклики $B_1 = (V_1^1, V_1^0)$ и $B_2 = (V_2^1, V_2^0)$, покрывающие граф G, имеют одну общую долю: согласно таблице для выбранного базиса И-НЕ $V_1^0 = V_2^0 = V^1$. Звездные биклики с приписанными переменными имеют вид

$$\begin{aligned} &(\{v_8, v_9, v_{10}, v_{11}\}, \{v_1\}) - x_3\, x_4, && (\{v_8, v_9, v_{10}, v_{11}\}, \{v_2\}) - x_1\, x_3, \\ &(\{v_8, v_9, v_{10}, v_{11}\}, \{v_3\}) - x_2\, x_4, && (\{v_8, v_9, v_{10}, v_{11}\}, \{v_4\}) - x_1\, x_2, \\ &(\{v_8, v_9, v_{10}, v_{11}\}, \{v_5\}) - x_2\, x_3\, x_5, && (\{v_8, v_9, v_{10}, v_{11}\}, \{v_6\}) - x_1\, x_2\, x_5, \\ &(\{v_8, v_9, v_{10}, v_{11}\}, \{v_7\}) - x_1\, x_3\, x_4. \end{aligned}$$

За начальные значения биклик B_1 и B_2 примем $(\{v_8, v_9, v_{10}, v_{11}\}, \{v_1\})$ и $(\{v_8, v_9, v_{10}, v_{11}\}, \{v_6\})$, поскольку пересечение соответствующих множеств $X_1 = \{x_3, x_4\}$ и $X_2 = \{x_1, x_2, x_5\}$ пусто, а X_2 имеет максимальную мощность. В результате выполнения следующего шага получаем пару

$$(\{v_8, v_9, v_{10}, v_{11}\}, \{v_1\}) - x_3\, x_4, \qquad (\{v_8, v_9, v_{10}, v_{11}\}, \{v_4, v_6\}) - x_1\, x_2\, x_5.$$

В приведенной ниже последовательности преобразований биклик B_1 и B_2 пара в последней строке представляет покрытие графа G:

$$(\{v_8, v_9, v_{10}, v_{11}\}, \{v_1, v_7\}) - x_1 x_3 x_4, \qquad (\{v_8, v_9, v_{10}, v_{11}\}, \{v_4, v_6\}) - x_1 x_2 x_5;$$
$$(\{v_8, v_9, v_{10}, v_{11}\}, \{v_1, v_2, v_7\}) - x_1 x_3 x_4, \qquad (\{v_8, v_9, v_{10}, v_{11}\}, \{v_4, v_6\}) - x_1 x_2 x_5;$$
$$(\{v_8, v_9, v_{10}, v_{11}\}, \{v_1, v_2, v_3, v_7\}) - x_1 x_2 x_3 x_4, \quad (\{v_8, v_9, v_{10}, v_{11}\}, \{v_4, v_6\}) - x_1 x_2 x_5;$$
$$(\{v_8, v_9, v_{10}, v_{11}\}, \{v_1, v_2, v_3, v_7\}) - x_1 x_2 x_3 x_4, \quad (\{v_8, v_9, v_{10}, v_{11}\}, \{v_4, v_5, v_6\}) - x_1 x_2 x_3 x_5.$$

Заданная функция $f(x_1, x_2, x_3, x_4, x_5)$ разлагается на две функции $f_1(x_1, x_2, x_3, x_4)$ и $f_2(x_1, x_2, x_3, x_5)$, связанные операцией И-НЕ, или «штрих Шеффера»: $f = f_1 \mid f_2$. Их можно задать матрицами, которые получаются следующим образом: для формирования $\mathbf{M}^1{}_1$ из $\mathbf{M}^0$ удаляется столбец x_5, для формирования $\mathbf{M}^0{}_1$ из $\mathbf{M}^1$ удаляются строки 4, 5, 6 и столбец x_5, для формирования $\mathbf{M}^1{}_2$ из $\mathbf{M}^0$ удаляется столбец x_4, для формирования $\mathbf{M}^0{}_2$ из $\mathbf{M}^1$ удаляются строки 1, 2, 3, 7 и столбец x_4. Формирование матриц сопровождается выполнением над ними операций простого поглощения и простого склеивания, после чего они приобретают следующий вид:

$$\mathbf{M}^1{}_1 = \begin{matrix} x_1 & x_2 & x_3 & x_4 \end{matrix}\begin{bmatrix} 0 & 1 & 0 & 0 \\ 1 & - & 1 & 0 \\ 1 & 0 & 0 & 1 \end{bmatrix}\begin{matrix} 1 \\ 2 \\ 3 \end{matrix}, \quad \mathbf{M}^0{}_1 = \begin{matrix} x_1 & x_2 & x_3 & x_4 \end{matrix}\begin{bmatrix} - & - & 1 & 1 \\ 0 & - & 1 & - \\ - & 1 & - & 1 \\ 1 & - & 0 & 0 \end{bmatrix}\begin{matrix} 4 \\ 5 \\ 6 \\ 7 \end{matrix};$$

$$\mathbf{M}^1{}_2 = \begin{matrix} x_1 & x_2 & x_3 & x_5 \end{matrix}\begin{bmatrix} 0 & 1 & 0 & 1 \\ 1 & - & 1 & 0 \\ 1 & 0 & - & - \end{bmatrix}\begin{matrix} 1 \\ 2 \\ 3 \end{matrix}, \quad \mathbf{M}^0{}_2 = \begin{matrix} x_1 & x_2 & x_3 & x_5 \end{matrix}\begin{bmatrix} 0 & 0 & - & - \\ - & 1 & 0 & 0 \\ 1 & 1 & - & 1 \end{bmatrix}\begin{matrix} 4 \\ 5 \\ 6 \end{matrix},$$

где нижние индексы у символов матриц совпадают с индексами соответствующих функций.

Нетрудно видеть, что функции f_1 и f_2 являются не полностью определенными. Значение функции f_1 не определено на наборе $(x_1, x_2, x_3, x_4) = (0, 0, 0, 0)$, а значение функции f_2 – на интервале, представляемом вектором $(0, 1, 1, –)$. Для дальнейшего разложения этих функций построим двудольные графы G_1 и G_2 с долями $V^{11} = \{v_1{}^1, v_2{}^1, v_3{}^1\}$, $V^{01} = \{v_4{}^1, v_5{}^1, v_6{}^1, v_7{}^1\}$ и $V^{12} = \{v_1{}^2, v_2{}^2, v_3{}^2\}$, $V^{02} = \{v_4{}^2, v_5{}^2, v_6{}^2\}$ соответственно. Графы G_1 и G_2 представим матрицами

$$\mathbf{G}_1 = \begin{matrix} 4 & 5 & 6 & 7 \end{matrix}\begin{bmatrix} x_3 \vee x_4 & x_3 & x_4 & x_1 \\ x_4 & x_1 & x_4 & x_3 \\ x_3 & x_1 \vee x_3 & x_2 & x_4 \end{bmatrix}\begin{matrix} 1 \\ 2 \\ 3 \end{matrix}, \qquad \mathbf{G}_2 = \begin{matrix} 4 & 5 & 6 \end{matrix}\begin{bmatrix} x_2 & x_5 & x_1 \\ x_1 & x_3 & x_5 \\ x_1 & x_2 & x_2 \end{bmatrix}\begin{matrix} 1 \\ 2 \\ 3 \end{matrix}.$$

Приведем слева звездные биклики для графа G_1, справа – для графа G_2:

$$(\{v_4{}^1, v_5{}^1, v_6{}^1, v_7{}^1\}, \{v_1{}^1\}) - x_1 x_3 x_4, \qquad (\{v_4{}^2, v_5{}^2, v_6{}^2\}, \{v_1{}^2\}) - x_1 x_2 x_5,$$
$$(\{v_4{}^1, v_5{}^1, v_6{}^1, v_7{}^1\}, \{v_2{}^1\}) - x_1 x_3 x_4, \qquad (\{v_4{}^2, v_5{}^2, v_6{}^2\}, \{v_2{}^2\}) - x_1 x_3 x_5,$$
$$(\{v_4{}^1, v_5{}^1, v_6{}^1, v_7{}^1\}, \{v_3{}^1\}) - x_2 x_3 x_4; \qquad (\{v_4{}^2, v_5{}^2, v_6{}^2\}, \{v_3{}^2\}) - x_1 x_2.$$

В том же порядке приведем пары биклик $(B_1{}^1, B_2{}^1)$ и $(B_1{}^2, B_2{}^2)$, покрывающие соответственно графы G_1 и G_2:

$$B_1{}^1 = (\{v_4{}^1, v_5{}^1, v_6{}^1, v_7{}^1\}, \{v_1{}^1, v_2{}^1\}) - x_1 x_3 x_4, \qquad B_1{}^2 = (\{v_4{}^2, v_5{}^2, v_6{}^2\}, \{v_1{}^2, v_3{}^2\}) - x_1 x_2 x_5,$$
$$B_2{}^1 = (\{v_4{}^1, v_5{}^1, v_6{}^1, v_7{}^1\}, \{v_3{}^1\}) - x_2 x_3 x_4; \qquad B_2{}^2 = (\{v_4{}^2, v_5{}^2, v_6{}^2\}, \{v_2{}^2\}) - x_1 x_3 x_5.$$

Теперь представим функции f_1 и f_2 как $f_1(x_1, x_2, x_3, x_4) = f_3(x_1, x_3, x_4) \mid f_4(x_2, x_3, x_4)$ и $f_2(x_1, x_2, x_3, x_4) = f_5(x_1, x_2, x_5) \mid f_6(x_1, x_3, x_5)$, функции f_3, f_4, f_5 и f_6 зададим в виде следующих матриц:

$$\mathbf{M}^1{}_3 = \begin{matrix} x_1 & x_3 & x_4 \\ \end{matrix}\begin{bmatrix} 0 & 1 & - \\ - & - & 1 \end{bmatrix}\begin{matrix} 1, \\ 2 \end{matrix} \quad \mathbf{M}^0{}_3 = \begin{matrix} x_1 & x_3 & x_4 \\ \end{matrix}\begin{bmatrix} 0 & 0 & 0 \\ 1 & 1 & 0 \end{bmatrix}\begin{matrix} 3; \\ 4 \end{matrix} \quad \mathbf{M}^1{}_4 = \begin{matrix} x_2 & x_3 & x_4 \\ \end{matrix}\begin{bmatrix} - & 1 & - \\ 1 & - & 1 \end{bmatrix}\begin{matrix} 1, \\ 2 \end{matrix} \quad \mathbf{M}^0{}_4 = \begin{matrix} x_2 & x_3 & x_4 \\ \end{matrix}\begin{bmatrix} 0 & 0 & 1 \end{bmatrix}\begin{matrix} 3; \end{matrix}$$

$$\mathbf{M}^1{}_5 = \begin{matrix} x_1 & x_2 & x_5 \\ \end{matrix}\begin{bmatrix} 0 & 0 & - \\ - & 1 & 0 \\ 1 & 1 & 1 \end{bmatrix}\begin{matrix} 1 \\ 2, \\ 3 \end{matrix} \quad \mathbf{M}^0{}_5 = \begin{matrix} x_1 & x_2 & x_5 \\ \end{matrix}\begin{bmatrix} 0 & 1 & 1 \\ 1 & 0 & - \end{bmatrix}\begin{matrix} 4; \\ 5 \end{matrix} \quad \mathbf{M}^1{}_6 = \begin{matrix} x_1 & x_3 & x_5 \\ \end{matrix}\begin{bmatrix} 0 & - & - \\ - & 0 & 0 \\ 1 & - & 1 \end{bmatrix}\begin{matrix} 1 \\ 2, \\ 3 \end{matrix} \quad \mathbf{M}^0{}_6 = \begin{matrix} x_1 & x_3 & x_5 \\ \end{matrix}\begin{bmatrix} 1 & 1 & 0 \end{bmatrix}\begin{matrix} 4. \end{matrix}$$

Графы G_3, G_4, G_4 и G_4, соответствующие функциям f_3, f_4, f_5 и f_6, представим матрицами

$$\mathbf{G}_3 = \begin{matrix} 3 & 4 \\ \end{matrix}\begin{bmatrix} x_3 & x_1 \\ x_4 & x_4 \end{bmatrix}\begin{matrix} 1, \\ 2 \end{matrix} \quad \mathbf{G}_4 = \begin{matrix} 3 \\ \end{matrix}\begin{bmatrix} x_3 \\ x_2 \end{bmatrix}\begin{matrix} 1, \\ 2 \end{matrix} \quad \mathbf{G}_5 = \begin{matrix} 4 & 5 \\ \end{matrix}\begin{bmatrix} x_2 & x_1 \\ x_5 & x_2 \\ x_1 & x_2 \end{bmatrix}\begin{matrix} 1 \\ 2, \\ 3 \end{matrix} \quad \mathbf{G}_6 = \begin{matrix} 4 \\ \end{matrix}\begin{bmatrix} x_1 \\ x_3 \\ x_5 \end{bmatrix}\begin{matrix} 1 \\ 2. \\ 3 \end{matrix}$$

Пары биклик, покрывающие указанные графы, имеют следующий вид:

$$B_1{}^3 = (\{v_3{}^3, v_4{}^3\}, \{v_1{}^3\}) - x_1 x_3, \qquad B_1{}^4 = (\{v_3{}^4\}, \{v_1{}^4\}) - x_3,$$
$$B_2{}^3 = (\{v_3{}^3, v_4{}^3\}, \{v_2{}^3\}) - x_4; \qquad B_2{}^4 = (\{v_3{}^4\}, \{v_2{}^4\}) - x_2;$$

$$B_1{}^5 = (\{v_4{}^5, v_5{}^5\}, \{v_1{}^5, v_3{}^5\}) - x_1 x_2, \quad B_1{}^6 = (\{v_4{}^6\}, \{v_1{}^6, v_3{}^6\}) - x_1 x_5,$$
$$B_2{}^5 = (\{v_4{}^5, v_5{}^5\}, \{v_2{}^5\}) - x_2 x_5; \qquad B_2{}^6 = (\{v_4{}^6\}, \{v_2{}^6\}) - x_3.$$

Отсюда получим разложения на функции с числом аргументов не более двух:

$$f_3(x_1, x_3, x_4) = f_7(x_1, x_3) \mid \bar{x}_4, \qquad f_4(x_2, x_3, x_4) = \bar{x}_2 \mid \bar{x}_3,$$
$$f_5(x_1, x_2, x_5) = f_8(x_1, x_2) \mid f_9(x_2, x_5), \qquad f_6(x_1, x_3, x_5) = f_{10}(x_1, x_5) \mid x_3.$$

Функции f_7, f_8, f_9 и f_{10} можно задать матрицами

$$\mathbf{M}^1{}_7 = \begin{matrix} x_1 & x_3 \\ \end{matrix}\begin{bmatrix} 0 & 0 \\ 1 & 1 \end{bmatrix}, \quad \mathbf{M}^0{}_7 = \begin{matrix} x_1 & x_3 \\ \end{matrix}\begin{bmatrix} 0 & 1 \end{bmatrix}; \qquad \mathbf{M}^1{}_8 = \begin{matrix} x_1 & x_2 \\ \end{matrix}\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, \quad \mathbf{M}^0{}_8 = \begin{matrix} x_1 & x_2 \\ \end{matrix}\begin{bmatrix} 0 & 0 \\ 1 & 1 \end{bmatrix};$$

$$\mathbf{M}^1{}_9 = \begin{matrix} x_2 & x_5 \\ \end{matrix}\begin{bmatrix} - & 1 \\ 0 & - \end{bmatrix}, \quad \mathbf{M}^0{}_9 = \begin{matrix} x_2 & x_5 \\ \end{matrix}\begin{bmatrix} 1 & 0 \end{bmatrix}; \qquad \mathbf{M}^1{}_{10} = \begin{matrix} x_1 & x_5 \\ \end{matrix}\begin{bmatrix} 1 & 0 \end{bmatrix}, \quad \mathbf{M}^0{}_{10} = \begin{matrix} x_1 & x_5 \\ \end{matrix}\begin{bmatrix} 0 & - \\ - & 1 \end{bmatrix}.$$

Значение функции f_7 не определено на наборе (0, 1), и она реализуется как $f_7 = \bar{x}_1 \mid x_3$. Для остальных функций справедливы следующие соотношения:

$$f_8 = x_1 \oplus x_2 = f_{11} \mid f_{12}, \qquad f_9 = x_2 \mid \bar{x}_5, \qquad f_{10} = x_1\, \bar{x}_5 = \bar{f}_{13},$$
$$f_{11} = x_1 \mid \bar{x}_2, \qquad f_{12} = \bar{x}_1 \mid x_2, \qquad f_{13} = x_1 \mid \bar{x}_5.$$

Таким образом, получена структура, представленная системой функций $f, f_1, \ldots, f_{13}$. Соответствующая комбинационная схема из элементов И-НЕ, дополненная инверторами, изображена на рис. 1.

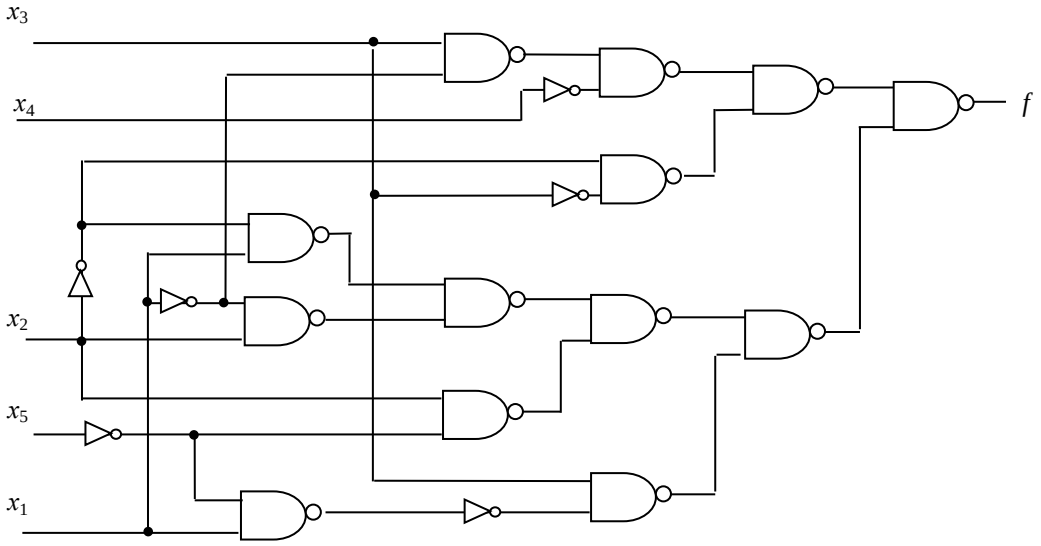


Рис. 1. Схема из элементов И-НЕ

Fig. 1. Circuit with NAND gates

Получим теперь комбинационную схему, реализующую ту же функцию в базисе элементов И и ИЛИ, с доступными инверсиями входных сигналов. Согласно таблице в биклики $B_1 = (V_1^1, V_1^0)$ и $B_2 = (V_2^1, V_2^0)$, покрывающих граф G, $V_1^1 = V_2^1 = V^1$ для операции И и $V_1^0 = V_2^0 = V^0$ для операции ИЛИ. По матрице **G** можно заметить, что для лучшего варианта разложения $f = \varphi(g_1, g_2)$ желательно выбрать для функции φ операцию И, если у матрицы $\mathbf{M}^0$ больше строк, чем у матрицы $\mathbf{M}^1$, и наоборот, операцию ИЛИ, если у $\mathbf{M}^1$ строк больше, чем у $\mathbf{M}^0$. Лучшим считаем вариант, когда функции g_1 и g_2 имеют меньшее число существенных аргументов. Для выходной функции выбираем операцию ИЛИ ($f = f_1 \vee f_2$). Тогда звездные биклики, из которых надо выбрать пару биклик для начальных значений B_1 и B_2, имеют следующий вид:

$$(\{v_1\}, \{v_8, v_9, v_{10}, v_{11}\}) - x_3\, x_4; \qquad (\{v_2\}, \{v_8, v_9, v_{10}, v_{11}\}) - x_1\, x_3;$$
$$(\{v_3\}, \{v_8, v_9, v_{10}, v_{11}\}) - x_2\, x_4; \qquad (\{v_4\}, \{v_8, v_9, v_{10}, v_{11}\}) - x_1\, x_2;$$
$$(\{v_5\}, \{v_8, v_9, v_{10}, v_{11}\}) - x_2\, x_3\, x_5; \qquad (\{v_6\}, \{v_8, v_9, v_{10}, v_{11}\}) - x_1\, x_2\, x_5;$$
$$(\{v_7\}, \{v_8, v_9, v_{10}, v_{11}\}) - x_1\, x_3\, x_4.$$

Такое множество звездных биклик совпадает с точностью до порядка задания долей с тем множеством, которое было получено на первом этапе реализации заданной функции в базисе И-НЕ. Тем же путем получаем $B_1 = (\{v_1, v_2, v_3, v_7\}, \{v_8, v_9, v_{10}, v_{11}\})$ с переменными x_1, x_2, x_3, x_4 и $B_2 = (\{v_4, v_5, v_6\}, \{v_8, v_9, v_{10}, v_{11}\})$ с переменными x_1, x_2, x_3, x_5. Для разложения $f = f_1 \vee f_2$ имеем те же троичные матрицы с той лишь разницей, что $\mathbf{M}^1{}_i$ и $\mathbf{M}^0{}_i$ ($i = 1, 2$) поменялись местами:

$$\mathbf{M}^1{}_1 = \begin{matrix} x_1 & x_2 & x_3 & x_4 \end{matrix}\begin{bmatrix} - & - & 1 & 1 \\ 0 & - & 1 & - \\ - & 1 & - & 1 \\ 1 & - & 0 & 0 \end{bmatrix}\begin{matrix} 1 \\ 2 \\ 3 \\ 4 \end{matrix}, \quad \mathbf{M}^0{}_1 = \begin{matrix} x_1 & x_2 & x_3 & x_4 \end{matrix}\begin{bmatrix} 0 & 1 & 0 & 0 \\ 1 & - & 1 & 0 \\ 1 & 0 & 0 & 1 \end{bmatrix}\begin{matrix} 5 \\ 6 \\ 7 \end{matrix};$$

$$\mathbf{M}^1{}_2 = \begin{matrix} x_1 & x_2 & x_3 & x_5 \end{matrix}\begin{bmatrix} 0 & 0 & - & - \\ - & 1 & 0 & 0 \\ 1 & 1 & - & 1 \end{bmatrix}\begin{matrix} 1 \\ 2 \\ 3 \end{matrix}, \quad \mathbf{M}^0{}_2 = \begin{matrix} x_1 & x_2 & x_3 & x_5 \end{matrix}\begin{bmatrix} 0 & 1 & 0 & 1 \\ 1 & - & 1 & 0 \\ 1 & 0 & - & - \end{bmatrix}\begin{matrix} 4 \\ 5 \\ 6 \end{matrix}.$$

Графы G_1 и G_2, соответствующие функциям f_1 и f_2, представлены матрицами

$$\mathbf{G}_1 = \begin{array}{c} \begin{matrix} 5 & 6 & 7 \end{matrix} \\ \begin{bmatrix} x_3 \vee x_4 & x_4 & x_3 \\ x_3 & x_1 & x_1 \vee x_3 \\ x_4 & x_4 & x_2 \\ x_1 & x_3 & x_4 \end{bmatrix} \end{array} \begin{matrix} 1 \\ 2 \\ 3 \\ 4 \end{matrix}, \qquad \mathbf{G}_2 = \begin{array}{c} \begin{matrix} 5 & 6 & 7 \end{matrix} \\ \begin{bmatrix} x_2 & x_1 & x_1 \\ x_5 & x_3 & x_2 \\ x_1 & x_5 & x_2 \end{bmatrix} \end{array} \begin{matrix} 1 \\ 2 \\ 3 \end{matrix}.$$

Для функций f_1 и f_2 получим разложения $f_1 = f_3 \vee f_4$ и $f_2 = f_5 \vee f_6$. Звездные биклики графов G_1 и G_2 имеют следующий вид:

$$\begin{array}{ll} (\{v_1^{\,1}\}, \{v_5^{\,1}, v_6^{\,1}, v_7^{\,1}\}) - x_3\, x_4, & (\{v_1^{\,2}\}, \{v_5^{\,2}, v_6^{\,2}, v_7^{\,2}\}) - x_1\, x_2, \\ (\{v_2^{\,1}\}, \{v_5^{\,1}, v_6^{\,1}, v_7^{\,1}\}) - x_1\, x_3, & (\{v_2^{\,2}\}, \{v_5^{\,2}, v_6^{\,2}, v_7^{\,2}\}) - x_2\, x_3\, x_5, \\ (\{v_3^{\,1}\}, \{v_5^{\,1}, v_6^{\,1}, v_7^{\,1}\}) - x_2\, x_4, & (\{v_3^{\,2}\}, \{v_5^{\,2}, v_6^{\,2}, v_7^{\,2}\}) - x_1\, x_2\, x_5, \\ (\{v_4^{\,1}\}, \{v_5^{\,1}, v_6^{\,1}, v_7^{\,1}\}) - x_1\, x_3\, x_4. & \end{array}$$

Из этих звездных биклик получим пары $(B_1^{\,1}, B_2^{\,1})$ и $(B_1^{\,2}, B_2^{\,2})$, покрывающие соответственно графы G_1 и G_2:

$$\begin{array}{ll} B_1^{\,1} = (\{v_1^{\,1}, v_3^{\,1}\}, \{v_5^{\,1}, v_6^{\,1}, v_7^{\,1}\}) - x_2\, x_3\, x_4, & B_1^{\,2} = (\{v_1^{\,2}, v_3^{\,2}\}, \{v_5^{\,2}, v_6^{\,2}, v_7^{\,2}\}) - x_1\, x_2\, x_5, \\ B_2^{\,1} = (\{v_2^{\,1}, v_4^{\,1}\}, \{v_5^{\,1}, v_6^{\,1}, v_7^{\,1}\}) - x_1\, x_3\, x_4; & B_2^{\,2} = (\{v_2^{\,2}\}, \{v_5^{\,2}, v_6^{\,2}, v_7^{\,2}\}) - x_2\, x_3\, x_5. \end{array}$$

Не полностью определенные функции f_3, f_4, f_5 и f_6 представим в виде матриц

$$\mathbf{M}^1{}_3 = \begin{array}{c} \begin{matrix} x_2 & x_3 & x_4 \end{matrix} \\ \begin{bmatrix} - & 1 & 1 \\ 1 & - & 1 \end{bmatrix} \end{array} \begin{matrix} 1, \\ 2 \end{matrix} \qquad \mathbf{M}^0{}_3 = \begin{array}{c} \begin{matrix} x_2 & x_3 & x_4 \end{matrix} \\ \begin{bmatrix} 1 & 0 & 0 \\ - & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \end{array} \begin{matrix} 3 \\ 4; \\ 5 \end{matrix} \qquad \mathbf{M}^1{}_4 = \begin{array}{c} \begin{matrix} x_1 & x_3 & x_4 \end{matrix} \\ \begin{bmatrix} 0 & 1 & - \\ 1 & 0 & 0 \end{bmatrix} \end{array} \begin{matrix} 1, \\ 2 \end{matrix} \qquad \mathbf{M}^0{}_4 = \begin{array}{c} \begin{matrix} x_1 & x_3 & x_4 \end{matrix} \\ \begin{bmatrix} 0 & 0 & 0 \\ 1 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix} \end{array} \begin{matrix} 3 \\ 4; \\ 5 \end{matrix}$$

$$\mathbf{M}^1{}_5 = \begin{array}{c} \begin{matrix} x_1 & x_2 & x_5 \end{matrix} \\ \begin{bmatrix} 0 & 0 & - \\ 1 & 1 & 1 \end{bmatrix} \end{array} \begin{matrix} 1, \\ 2 \end{matrix} \qquad \mathbf{M}^0{}_5 = \begin{array}{c} \begin{matrix} x_1 & x_2 & x_5 \end{matrix} \\ \begin{bmatrix} 0 & 1 & 1 \\ 1 & - & 0 \\ 1 & 0 & - \end{bmatrix} \end{array} \begin{matrix} 3 \\ 4; \\ 5 \end{matrix} \qquad \mathbf{M}^1{}_6 = \begin{array}{c} \begin{matrix} x_2 & x_3 & x_5 \end{matrix} \\ \begin{bmatrix} 1 & 0 & 0 \end{bmatrix} \end{array} 1, \qquad \mathbf{M}^0{}_6 = \begin{array}{c} \begin{matrix} x_2 & x_3 & x_5 \end{matrix} \\ \begin{bmatrix} 1 & 0 & 1 \\ - & 1 & 0 \\ 0 & - & - \end{bmatrix} \end{array} \begin{matrix} 2 \\ 3. \\ 4 \end{matrix}$$

Соответствующие функциям f_3, f_4, f_5 и f_6 графы G_3, G_4, G_4 и G_4 представлены матрицами

$$\mathbf{G}_3 = \begin{array}{c} \begin{matrix} 3 & 4 & 5 \end{matrix} \\ \begin{bmatrix} x_3 \vee x_4 & x_4 & x_3 \\ x_4 & x_4 & x_2 \end{bmatrix} \end{array} \begin{matrix} 1, \\ 2 \end{matrix} \qquad \mathbf{G}_4 = \begin{array}{c} \begin{matrix} 3 & 4 & 5 \end{matrix} \\ \begin{bmatrix} x_3 & x_1 & x_1 \vee x_3 \\ x_1 & x_3 & x_4 \end{bmatrix} \end{array} \begin{matrix} 1, \\ 2 \end{matrix} \qquad \mathbf{G}_5 = \begin{array}{c} \begin{matrix} 3 & 4 & 5 \end{matrix} \\ \begin{bmatrix} x_2 & x_1 & x_1 \\ x_1 & x_5 & x_2 \end{bmatrix} \end{array} \begin{matrix} 1, \\ 2 \end{matrix} \qquad \mathbf{G}_6 = \begin{array}{c} \begin{matrix} 2 & 3 & 4 \end{matrix} \\ \begin{bmatrix} x_5 & x_3 & x_2 \end{bmatrix} \end{array} 1.$$

Для реализации функций f_3, f_4, f_5 и f_6 выберем операцию И. Тогда звездные биклики графов G_3, G_4, G_5 и G_6 будут иметь следующий вид:

$$\begin{array}{llll} (\{v_1^{\,3}, v_2^{\,3}\}, \{v_3^{\,3}\}) - x_4, & (\{v_1^{\,4}, v_2^{\,4}\}, \{v_3^{\,4}\}) - x_1\, x_3, & (\{v_1^{\,5}, v_2^{\,5}\}, \{v_3^{\,5}\}) - x_1\, x_2, & (\{v_1^{\,6}\}, \{v_2^{\,6}\}) - x_5, \\ (\{v_1^{\,3}, v_2^{\,3}\}, \{v_4^{\,3}\}) - x_4, & (\{v_1^{\,4}, v_2^{\,4}\}, \{v_4^{\,4}\}) - x_1\, x_3, & (\{v_1^{\,5}, v_2^{\,5}\}, \{v_4^{\,5}\}) - x_1\, x_5, & (\{v_1^{\,6}\}, \{v_3^{\,6}\}) - x_3, \\ (\{v_1^{\,3}, v_2^{\,3}\}, \{v_5^{\,3}\}) - x_2\, x_3, & (\{v_1^{\,4}, v_2^{\,4}\}, \{v_5^{\,4}\}) - x_3\, x_4, & (\{v_1^{\,5}, v_2^{\,5}\}, \{v_5^{\,5}\}) - x_1\, x_2, & (\{v_1^{\,6}\}, \{v_4^{\,6}\}) - x_2. \end{array}$$

Покрытиями графов G_3, G_4, G_5 и G_6 будут пары

$$\begin{array}{ll} B_1^{\,3} = (\{v_1^{\,3}, v_2^{\,3}\}, \{v_3^{\,3}, v_4^{\,3}\}) - x_4, & B_1^{\,4} = (\{v_1^{\,4}, v_2^{\,4}\}, \{v_3^{\,4}, v_4^{\,4}\}) - x_1\, x_3, \\ B_2^{\,3} = (\{v_1^{\,3}, v_2^{\,3}\}, \{v_5^{\,3}\}) - x_2\, x_3; & B_2^{\,4} = (\{v_1^{\,4}, v_2^{\,4}\}, \{v_5^{\,4}\}) - x_3\, x_4; \end{array}$$

$$B_1^5 = (\{v_1^5, v_2^5\}, \{v_3^5, v_5^5\}) - x_1 x_2, \quad B_1^6 = (\{v_1^6\}, \{v_2^6, v_3^6\}) - x_3 x_5,$$
$$B_2^5 = (\{v_1^5, v_2^5\}, \{v_4^5\}) - x_1 x_5; \qquad B_2^6 = (\{v_1^6\}, \{v_4^6\}) - x_2.$$

Приведенные пары биклик определяют разложения

$$f_3(x_2, x_3, x_4) = f_7(x_2, x_3) \wedge x_4, \qquad f_4(x_2, x_3, x_4) = f_8(x_1, x_3) \wedge f_9(x_3, x_4),$$
$$f_5(x_1, x_2, x_5) = f_{10}(x_1, x_2) \wedge f_{11}(x_1, x_5), \quad f_6(x_1, x_3, x_5) = f_{12}(x_3, x_5) \wedge x_2.$$

Для функций, составляющих разложения, матрицы имеют следующий вид:

$$\mathbf{M}^1{}_7 = \begin{matrix} x_2\ x_3 \\ \begin{bmatrix} - & 1 \\ 1 & - \end{bmatrix} \end{matrix}, \quad \mathbf{M}^0{}_7 = \begin{matrix} x_2\ x_3 \\ \begin{bmatrix} 0 & 0 \end{bmatrix} \end{matrix}; \qquad \mathbf{M}^1{}_8 = \begin{matrix} x_1\ x_3 \\ \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \end{matrix}, \quad \mathbf{M}^0{}_8 = \begin{matrix} x_1\ x_3 \\ \begin{bmatrix} 0 & 0 \\ 1 & 1 \end{bmatrix} \end{matrix};$$

$$\mathbf{M}^1{}_9 = \begin{matrix} x_3\ x_4 \\ \begin{bmatrix} 1 & - \\ - & 0 \end{bmatrix} \end{matrix}, \quad \mathbf{M}^0{}_9 = \begin{matrix} x_3\ x_4 \\ \begin{bmatrix} 0 & 1 \end{bmatrix} \end{matrix}; \qquad \mathbf{M}^1{}_{10} = \begin{matrix} x_1\ x_2 \\ \begin{bmatrix} 0 & 0 \\ 1 & 1 \end{bmatrix} \end{matrix}, \quad \mathbf{M}^0{}_{10} = \begin{matrix} x_1\ x_2 \\ \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \end{matrix};$$

$$\mathbf{M}^1{}_{11} = \begin{matrix} x_1\ x_5 \\ \begin{bmatrix} 0 & - \\ - & 1 \end{bmatrix} \end{matrix}, \quad \mathbf{M}^0{}_{11} = \begin{matrix} x_1\ x_5 \\ \begin{bmatrix} 1 & 0 \end{bmatrix} \end{matrix}; \qquad \mathbf{M}^1{}_{12} = \begin{matrix} x_3\ x_5 \\ \begin{bmatrix} 0 & 0 \end{bmatrix} \end{matrix}, \quad \mathbf{M}^0{}_{12} = \begin{matrix} x_3\ x_5 \\ \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \end{matrix}.$$

По этим матрицам получим алгебраические представления полностью определенных функций и реализации не полностью определенной функции f_{12}:

$$f_7 = x_2 \vee x_3, \qquad f_8 = x_1 \oplus x_3 = \bar{x}_1 x_3 \vee x_1 \bar{x}_3, \qquad f_9 = x_3 \vee \bar{x}_4,$$
$$f_{10} = x_1 \sim x_2 = \bar{x}_1 \bar{x}_2 \vee x_1 x_2, \qquad f_{11} = \bar{x}_1 \vee x_5, \qquad f_{12} = \bar{x}_3 x_5.$$

Соответствующая комбинационная схема из элементов И и ИЛИ изображена на рис. 2.

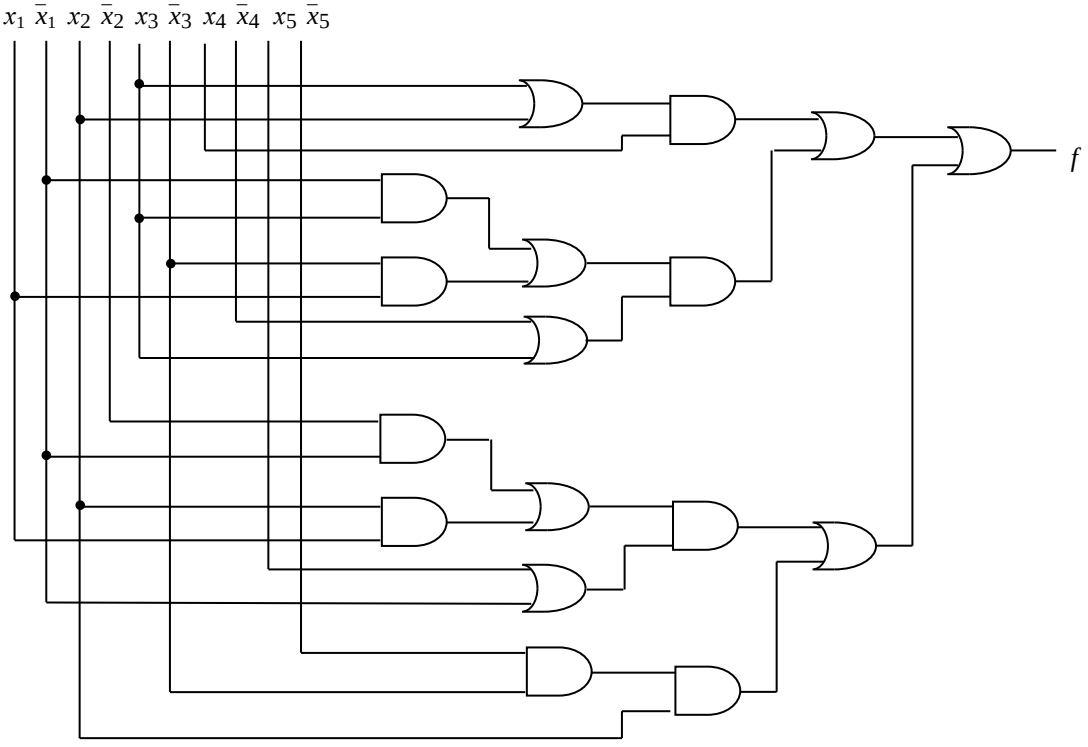


Рис. 2. Схема из элементов И и ИЛИ

Fig. 2. Circuit with AND and OR gates

Заключение. В статье показано, как можно применить метод алгебраической декомпозиции для синтеза комбинационных схем. Достоинством предложенного подхода является возможность получения схем с повышенным быстродействием, которое характеризуется числом уровней, или глубиной схемы. Представленный метод конкурентоспособен по отношению к факторизационному методу, описанному в работе [8]. Схема, реализующая систему булевых функций из примера в работе [8] и полученная описанным выше методом, содержит на единицу больше элементов, чем схема, полученная факторизационным методом при использовании двухвходовых элементов, но имеет на единицу меньше уровней. Для совместной реализации булевых функций из заданной системы должно быть предусмотрено выявление совпадения получаемых функций на каждом уровне декомпозиции.

Список использованных источников

1. Cortadella, J. Timing-driven logic bi-decomposition / J. Cortadella // IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems. – 2003. – Vol. 22, no. 6. – P. 675–685.

2. Mishchenko, A. An algorithm for bi-decomposition of logic functions / A. Mishchenko, B. Steinbach, M. Perkowski // Proc. of the 38th Annual Design Automation Conf. (DAC'2001), Las Vegas, USA, 18–22 June 2001. – Las Vegas, 2001. – P. 103–108.

3. Chang, S.-C. Technology mapping for TLU FPGA's based on decomposition of binary decision diagrams / S.-C. Chang, M. Marek-Sadowska, T. Hwang // IEEE Transactions Computer-Aided Design. – 1996. – Vol. 15, no. 10. – P. 1226–1235.

4. Бибило, П. Н. Декомпозиция булевых функций на основе решения логических уравнений / П. Н. Бибило. – Минск : Беларус. навука, 2009. – 211 с.

5. Zakrevskij, A. D. On a special kind decomposition of weakly specified Boolean functions / A. D. Zakrevskij // Second Intern. Conf. on Computer-Aided Design of Discrete Devices (CAD DD'97), Minsk, Belarus, 12–14 Nov. 1997 / National Academy of Sciences of Belarus, Institute of Engineering Cybernetics. – Minsk, 1997. – Vol. 1. – P. 36–41.

6. Поттосин, Ю. В. Эвристический метод алгебраической декомпозиции частичных булевых функций / Ю. В. Поттосин // Информатика. – 2020. – Т. 17, № 3. – С. 44–53.

7. Поттосин, Ю. В. Параллельно-последовательная декомпозиция системы частичных булевых функций / Ю. В. Поттосин, Е. А. Шестаков // Прикладная дискретная математика. – 2010. – № 4(10). – С. 55–63.

8. Закревский, А. Д. Логические основы проектирования дискретных устройств / А. Д. Закревский, Ю. В. Поттосин, Л. Д. Черемисинова. – М. : Физматлит, 2007. – 592 с.

9. Евстигнеев, В. А. Толковый словарь по теории графов в информатике и программировании / В. А. Евстигнеев, В. Н. Касьянов. – Новосибирск : Наука, 1999. – 291 с.

References

1. Cortadella J. Timing-driven logic bi-decomposition. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 2003, vol. 22, no. 6, pp. 675–685.

2. Mishchenko A., Steinbach B., Perkowski M. An algorithm for bi-decomposition of logic functions. *Proceedings of the 38th Annual Design Automation Conference (DAC'2001), Las Vegas, USA, 18–22 June 2001*. Las Vegas, 2001, pp. 103–108.

3. Chang S.-C., Marek-Sadowska M., Hwang T. Technology mapping for TLU FPGA's based on decomposition of binary decision diagrams. *IEEE Transactions Computer-Aided Design*, 1996, vol. 15, no. 10, pp. 1226–1235.

4. Bibilo P. N. Dekompozicija bulevyh funkcij na osnove reshenija logicheskih uravnenij. *Decomposition of Boolean Functions on the Base of Solving Logical Equations*. Minsk, Belaruskaja navuka, 2009, 211 p. (In Russ.).

5. Zakrevskij A. D. On a special kind decomposition of weakly specified Boolean functions. *Second International Conference on Computer-Aided Design of Discrete Devices (CAD DD'97), Minsk, Belarus, 12–14 November 1997*. National Academy of Sciences of Belarus, Institute of Engineering Cybernetics. Minsk, 1997, vol. 1, pp. 36–41.

6. Pottosin Yu. V. *A heuristic method for bi-decomposition of partial Boolean functions*. Informatika [*Informatics*], 2020, vol. 17, no. 3, pp. 44–53 (In Russ.).

7. Pottosin Yu. V., Shestakov E. A. *Series-parallel decomposition of a system of partial Boolean functions*. Prikladnaya diskretnaya matematika [*Applied Discrete Mathematics*], 2010, no. 4(10), pp. 55–63 (In Russ.).

8. Zakrevskij A. D., Pottosin Yu. V., Cheremisinova L. D. Logicheskie osnovy proektirovanija diskretnyh ustrojstv. *Logical Fundamentals of Discrete Devices Design*. Moscow, Fizmatlit, 2007, 592 p. (In Russ.).

9. Evstigneev V. A., Kas'janov V. N. Tolkovyj slovar' po teorii grafov v informatike i programmirovanii. *Explanatory Dictionary of Graph Theory Terminology in Informatics and Programming*. Novosibirsk, Nauka, 1999, 291 p. (In Russ.).

Информация об авторе

Поттосин Юрий Васильевич, кандидат физико-математических наук, ведущий научный сотрудник, Объединенный институт проблем информатики Национальной академии наук Беларуси.
E-mail: pott@newman.bas-net.by

Information about the author

Yuri V. Pottosin, Ph. D. (Phys.-Math.), Leading Researcher, The United Institute of Informatics Problems of the National Academy of Sciences of Belarus.
E-mail: pott@newman.bas-net.by

ЗАЩИТА ИНФОРМАЦИИ И НАДЕЖНОСТЬ СИСТЕМ
INFORMATION PROTECTION AND SYSTEM RELIABILITY

УДК 004.056.53
https://doi.org/10.37661/1816-0301-2022-19-1-19-31

Оригинальная статья
Original Paper

Детектирование признаков сетевой разведки с использованием модели дерева решений

Н. П. Шараев, С. Н. Петров✉

Белорусский государственный университет информатики и радиоэлектроники,
ул. П. Бровки, 6, Минск, 220013, Беларусь
✉*E-mail: sergpetrov@inbox.ru*

Аннотация

Цели. Своевременное обнаружение сетевой разведки позволяет снизить риски информационной безопасности организаций. Исследование проводилось с целью разработки программного модуля обнаружения признаков сетевой разведки с использованием методов машинного обучения.

Методы. Основными методами детектирования признаков сетевой разведки являлись: анализ открытых датасетов соответствующего назначения; формирование метрик, характерных для сетевой разведки; разработка набора данных разведки на основе определенных метрик. Исследовалась эффективность методов машинного обучения для задачи классификации.

Результаты. Спроектированы топология и тестовый сегмент в корпоративной сети РУП «Белтелеком» для создания датасета. Для детектирования и анализа событий разработано средство мониторинга, результаты работы которого использовались в качестве основы для нового датасета.

Реализация метода дерева принятия решений в виде программного кода позволила увеличить скорость работы модуля приблизительно в два раза (0,147 мс). Практические испытания разработанного модуля показали факт сработки на все типы сканирования сетей с помощью утилит Nmap и Masscan.

Заключение. Анализ датасета методом главных компонент показал наличие пограничной области между событиями легального трафика и трафика сетевой разведки, что положительно сказалось на обучении модели. Изучены и протестированы наиболее перспективные методы машинного обучения с использованием различных гиперпараметров. Наилучшие результаты показал метод дерева принятия решений с параметрами criterion = gini и splitter = random и скоростью работы 0,333 мс.

Ключевые слова: сетевая разведка, аномалии сетевого трафика, машинное обучение, метрики признаков разведки, датасеты

Для цитирования. Шараев, Н. П. Детектирование признаков сетевой разведки с использованием модели дерева решений / Н. П. Шараев, С. Н. Петров // Информатика. – 2022. − Т. 19, № 1. – С. 19–31.
https://doi.org/10.37661/1816-0301-2022-19-1-19-31

Конфликт интересов. Авторы заявляют об отсутствии конфликта интересов.

Поступила в редакцию | Received 12.08.2021
Подписана в печать | Accepted 10.12.2021
Опубликована | Published 29.03.2022

Detection of network intelligence features with the decision tree model

Nikita P. Sharaev, Sergei N. Petrov✉

Belarusian State University
of Informatics and Radioelectronics,
st. P. Brovki, 6, Minsk, 220013, Belarus
✉*E-mail: sergpetrov@inbox.ru*

Abstract
Objectives. Early detection of network intelligence allows to reduce the risks of information security of organizations. The study was carried out to develop software module for detecting the features of network intelligence by machine learning methods.
Methods. Analysis of open datasets of appropriate destination; formation of metrics characteristic of network intelligence; development of a dataset based on certain metrics; study of the effectiveness of machine learning methods for classification task.
Results. The topology was designed and a test segment was created in the corporate network of RUE "Beltelecom" to create a dataset. A monitoring tool has been developed for detecting and analyzing the events, the results of which were used as the basis for a new dataset.
The implementation of the decision tree method in the form of program code allowed to increase the speed of the module by about 2 times (0,147 ms). Practical tests of the developed module have shown the alarm on all types of network scanning using Nmap and Masscan utilities.
Conclusion. The analysis of the dataset by principal component method showed the presence of a border area between the events of legal traffic and network intelligence traffic, which had a positive effect on the training of the model. The most promising machine learning methods have been studied and tested using various hyperparameters. The best results were shown by the decision tree method with the parameters criterion = gini and splitter = random and speed as 0,333 ms.

Keywords: network intelligence, network traffic anomalies, machine learning, intelligence feature metrics, datasets

For citation. Sharaev N. P., Petrov S. N. *Detection of network intelligence features with the decision tree model*. Informatika [*Informatics*], 2022, vol. 19, no. 1, pp. 19–31 (In Russ.).
https://doi.org/10.37661/1816-0301-2022-19-1-19-31

Conflict of interest. The authors declare of no conflict of interest.

Введение. Сетевая разведка является первым этапом целенаправленной атаки на информационные системы, обнаружение которой позволяет снизить риски информационной безопасности. Она представляет собой комплекс мероприятий по получению и обработке данных об информационной системе, функционирующих в ней информационных ресурсах, средствах защиты информации и используемом программном обеспечении (ПО) [1].

Перечислим наиболее частые способы проведения сетевой разведки:

– анализ открытой (доступной в сети Интернет) информации с использованием расширенных запросов веб-браузеров;

– получение информации (например, списка выделенных IP-адресов, наименования владельца домена) от whois-серверов;

– получение информации от DNS-серверов;

– сканирование информационной сети;

– сканирование портов транспортного уровня.

Повлиять на доступ злоумышленника к информации от whois- и DNS-серверов, а также к открытой информации в сети Интернет невозможно. Наиболее эффективный способ обнаружения сетевой разведки – это выявление фактов сканирования информационной сети и портов транспортного уровня.

Анализ сетевой модели стека сетевых протоколов OSI (URL: http://www.williamspublishing.com/PDF/5-8459-0589-3/part.pdf) показал, что эффективное проведение сетевой разведки возможно только на втором, третьем, четвертом и седьмом (канальном, сетевом, транспортном и прикладном) уровнях модели OSI.

Сетевая разведка на втором уровне неэффективна, так как информация на этом уровне предназначена для передачи сообщений в рамках одного широковещательного домена локальной сети и не может выйти за пределы сети организации. При проведении злоумышленником сетевой разведки полученная им информация об инфраструктуре организации либо невалидна, либо относится к сетевой инфраструктуре используемого провайдера.

Проведение сетевой разведки на седьмом уровне модели OSI является сложной задачей из-за большого числа различных прикладных протоколов и служб.

Универсальным способом сетевой разведки остается сканирование третьего и четвертого уровней модели OSI. Доступ к ним злоумышленник может получить как из периметра сети организации, так и извне. Служебная информация, передаваемая на данных уровнях, содержит IP-адреса информационных систем организации, прослушиваемые и открытые транспортные порты, а также информацию о службах, функционирующих на прослушиваемых портах.

Целью сканирования информационной сети является поиск доступных устройств, находящихся в режиме онлайн. Обычно для этого используется сетевой протокол ICMP, предназначенный для передачи сообщений об ошибках и других исключительных ситуациях, возникших при передаче данных. В частности, это происходит, когда запрашиваемая услуга недоступна или хост (маршрутизатор) не отвечает. В подавляющем большинстве ОС (Windows, Linux, Cisco IOS, Huawei VRP и др.) для отправки ICMP-пакетов используются утилиты Ping и Tracert. Указанные утилиты распространены и легализованы производителями, и злоумышленники могут использовать их для изучения топологии корпоративной сети организации. В настоящее время на компьютерах под управлением ОС Windows трафик ICMP по умолчанию блокируют на межсетевом экране. Вместе с тем очень распространенной практикой является настройка исключений в межсетевом экране для данного типа трафика. Это связано с необходимостью быстрой проверки доступности отдельных хостов в корпоративной сети (troubleshooting).

Целью сканирования портов транспортного уровня является определение функционирующих на конкретном узле служб. Для этого проводятся попытки инициализации соединения на транспортных портах, лежащих в диапазоне 1–65535. Согласно спецификации первые 1024 порта зарезервированы под общеизвестные службы ОС, такие как FTP, SSH, SMTP, HTTP, HTTPS, SMB, и не могут быть переназначены. Зарегистрированные порты в диапазоне 1024–49151 могут переопределяться, но обычно не изменяются и назначаются на стандартные службы (RDP, Apache Tomcat, L2TP, MySQL, NFS и т. п.). Поэтому для получения информации о функционирующих и доступных извне службах злоумышленнику необходимо провести либо сканирование общеизвестных и зарегистрированных портов транспортного уровня (1–49151), что может занять ощутимый промежуток времени, либо выполнить частичное сканирование популярных портов. Перечень популярных портов транспортного уровня чаще всего зависит от ПО, которым пользуется злоумышленник. Общий перечень допустимых диапазонов портов транспортного уровня представлен в табл. 1.

Таблица 1
Сетевая разведка различных диапазонов портов транспортного уровня модели OSI

Table 1
Network intelligence of different ranges of ports of the transport layer of the OSI model

Название диапазона *Range name*	Диапазон портов *Port range*	Проведение сетевой разведки *Network intelligence*
Общеизвестные	1–1024	Целесообразно
Зарегистрированные	1024–49151	Целесообразно
Динамические	49152–65535	Нецелесообразно
Общие	1–65535	Нецелесообразно
Популярные	7, 20, 21, 22, 23, 25, 3306, 3389, 5432, 5601, 8080	Целесообразно

Наиболее популярными сканерами являются Nmap (Windows, Linux) и Masscan (Linux), хотя сканирование можно выполнять и с помощью стандартных средств ОС (Powershell) и (или) языков программирования Python, C++ и Ruby. По результатам сканирования нарушитель может построить приблизительную сетевую топологию организации и выявить возможные уязвимости в информационных системах.

В мировой практике задача обнаружения признаков сетевой разведки сопоставима с задачей обнаружения аномалий сетевого трафика, так как в обоих случаях возникает большое количество подозрительного трафика на сетевом и транспортном уровнях сети. Следовательно, признаки аномалий сетевого трафика валидны и могут быть применены для обнаружения сетевой разведки.

Обнаружение аномалии сетевого трафика. В соответствии с работой [2] для выявления аномалии сетевого трафика на сетевом уровне анализируются следующие данные: IP-адреса источника и назначения, дата и время получения IP-пакета, размер IP-пакета. Аномалия сетевого трафика на транспортном уровне определяется исходя из следующих признаков: количества входящих, исходящих либо внутрисетевых IP-, TCP- и UDP-пакетов; количества опросов неразрешенных портов UDP; количества завершенных запросов по протоколу UDP; количества незавершенных запросов по протоколу UDP; превышения продолжительности таймаута ответа узла; количества опросов портов TCP; количества опросов разрешенных портов TCP; количества соединений TCP, находящихся в состоянии установления (SYN); количества соединений TCP, находящихся в открытом состоянии (ESTABLISHED); количества соединений TCP, находящихся в состоянии закрытия (FIN) в единицу времени; отношения количества опросов разрешенных портов протокола TCP к количеству опросов всех портов этого протокола; отношения количества открываемых соединений TCP к общему количеству соединений [3]. Приведенные перечни признаков могут быть неполными, так как в различных источниках выделяют разные признаки.

Датасеты для обучения нейронных сетей. В настоящее время особой популярностью пользуются алгоритмы, основанные на нейронных сетях и глубоком обучении. Такие алгоритмы применяются для анализа большого количества сложных данных с неопределенными признаками. Анализ задач машинного обучения для определения признаков сетевой разведки показал, что наиболее подходящими являются алгоритмы классификации, регрессии, прогнозирования, кластеризации и сокращения размерности. Их эффективность можно увеличить путем использования групп одинаковых алгоритмов, объединенных для исправления ошибок друг друга. При этом несколько алгоритмов обучения могут показать результат выше, чем каждый из них в отдельности. Обычно объединяют сильно зависящие от входных данных алгоритмы (в частности, регрессию и дерево принятия решений). Это позволяет стабилизировать полученный результат, несмотря на возможное наличие сильных искажений в множестве входных объектов.

Следует отметить два вида совместного использования алгоритмов: бэггинг (bootstrap aggregating) и бустинг (boosting). *Бэггинг* состоит из параллельных однотипных алгоритмов, обученных на разных выборках одного множества входных объектов, и выводит усредненный результат их работы. Основным преимуществом бэггинга является возможность его применения в масштабе реального времени [4]. *Бустинг* состоит из последовательно выполняемых однотипных алгоритмов, обученных на специальных выборках множества входных объектов и выводящих окончательный результат. Каждая специальная выборка включает исходные объекты множества и части данных, на которых алгоритм на предыдущем шаге отработал неправильно [5]. В сравнении с бэггингом бустинг показывает лучший результат, но процесс его выполнения занимает больше времени.

В качестве дополнения к указанным методам исследовалась эффективность многослойного персептрона в связи с возможностью его использования для анализа признаков сетевой разведки в сравнении со сверточными и рекуррентными нейронными сетями. *Многослойный персептрон* (multilayer perceptron, MLP) – простейшая, но эффективная нейронная сеть с несколькими скрытыми слоями, предназначенная для анализа большого количества данных с определяемыми признаками.

Эффективность методов машинного обучения во многом базируется на объеме и качестве данных, используемых для тренировки моделей алгоритмов. Согласно опубликованному отчету консалтинговой компании International Data Corporation «The Digitization of the World from Edge to Core» (URL: https://www.seagate.com/files/www-content/our-story/trends/files/idc-seagate-dataage-whitepaper.pdf) объем накопленных данных для алгоритмов машинного обучения на 2018 г. составляет 33 ЗБ, а к 2025 г. их число может вырасти до 175 ЗБ.

Наборы данных, содержащие большое количество размеченных показателей, пригодных для машинного обучения, называются *датасетами*. Существуют следующие датасеты, связанные с обнаружением аномалий сетевого трафика:

Network Intrusion Detection (URL: https://www.kaggle.com/sampadab17/network-intrusion-detection/);

UNSW_NB15 (URL: https://www.kaggle.com/mrwellsdavid/unsw-nb15/);

2019 Trendmicro CTF Wildcard 400 (URL: https://www.kaggle.com/hawkcurry/2019-trendmicro-ctf-wildcard-400/);

Kitsune Network Attack (URL: https://www.kaggle.com/ymirsky/network-attack-dataset-kitsune/);

NSL-KDD (URL: https://www.unb.ca/cic/datasets/nsl.html);

NTwPSA (Loughborough University Network Traffic with Port Scanning Attack; URL: https://figshare.com/articles/dataset/Loughborough_University_Network_Traffic_with_Port_Scanning_Attack/4630282/3).

Каждый датасет содержит в себе конечное множество метрик, характерных для конкретного события «объект – ответ». Последние три датасета включают данные для обнаружения сетевой разведки. Наиболее популярным датасетом для обнаружения аномалий сетевого трафика и сетевой разведки является NSL-KDD, который представляет собой новую версию устаревшего датасета KDD99, разработанного в 1999 г. для проведения сравнительного тестирования алгоритмов в IPS/IDS, и содержит около 150 тыс. событий. В NSL-KDD устранены отдельные недостатки KDD99 [6], удалены события, влияющие на частотные характеристики (избыточность, дублирование), и создан более продуманный подход к формированию тестовой и обучающей выборок. События NSL-KDD – это последовательности сегментов (дейтаграмм) TCP и UDP, а также ICMP-пакетов, полученные в определенный промежуток времени. Для каждого события проведен расчет 41 метрики, добавлена метка атаки и уровень сложности. Эти датасеты на практике имеют ограниченное применение вследствие малой или, наоборот, избыточной информативности отдельных метрик. Так, метрики датасета NSL-KDD root, file creations, shells и др. обладают избыточной информативностью и не используются при обнаружении признаков сетевой разведки, а также сложны в определении. В расчете метрик датасета NTwPSA mac_src_second и mac_dst_second тоже нет необходимости.

Определение метрик, характерных для сетевой разведки. Для создания эффективного датасета необходимо определить информацию, доступную для извлечения из заголовков протоколов IP, TCP и UDP. Наиболее информативными полями являются: полная длина IP-пакета, IP-адрес источника, IP-адрес назначения, исходящий порт и порт назначения. Для протокола TCP дополнительно внесено поле «Код», описывающее тип запроса при рукопожатии (SYN, ACK, FIN и т. д.). При этом анализ полей должен проводиться не для одного события, а для выборки, что позволяет сформировать связь между событиями и контекст. Предполагается, что наиболее информативными будут метрики, состоящие из отношения количества содержащих отдельный параметр событий к общему количеству событий в выборке. Общее количество событий в выборке сложно определить теоретически, поэтому на начальном этапе используется дифференциация множества событий на совокупности по 30 событий.

Кроме метрик serror rate и dst host rate, предложенных в NSL-KDD, а также port_dst_second от NTwPSA требуется рассмотреть метрики, напрямую связанные с признаками сетевой разведки. В качестве таких метрик могут выступать коды, содержащиеся в TCP-заголовке. Из них наиболее часто используются коды FIN, SYN, ACK, MAIMON, XMAS или же NULL. Данные метрики также необходимо представить в виде отношения, что позволит точнее определить принадлежность события к сетевой разведке. Кроме того, требуется рассчитать

отношение других кодов к выборке, что связано с возможностью проведения злоумышленником ранее неизвестных типов атак. Итог анализа и расчета метрик приведен в табл. 2 [7].

Таблица 2
Формулы расчета и описание предлагаемых метрик

Table 2
Calculation formulas and description of the proposed metrics

Метрика *Metric*	Формула расчета *Calculation formula*	Описание *Description*
count	count_ip_events(all, all) / count_all_events	Отношение количества отправленных сегментов (дейтаграмм) с одного IP-адреса к общему количеству полученных сегментов (дейтаграмм) с различных IP-адресов. Позволяет определить «шумные» хосты
udp	count_ip_events(udp, all) / count_ip_events(all, all)	Отношение количества отправленных дейтаграмм с одного IP-адреса к общему количеству отправленных с этого же IP-адреса сегментов (дейтаграмм). Определяет UDP-трафик
tcp	1.0 - udp	Отношение количества отправленных сегментов с одного IP-адреса к общему количеству отправленных с этого же IP-адреса сегментов (дейтаграмм). Определяет TCP-трафик
tcp_syn	count_ip_events(tcp, syn) / count_ip_events(tcp, all)	Отношение количества отправленных с флагом SYN сегментов с одного IP-адреса к общему количеству отправленных с этого же IP-адреса сегментов. Определяет объем сегментов с флагом SYN
tcp_ack	count_ip_events(tcp, ack) / count_ip_events(tcp, all)	Отношение количества отправленных с флагом ACK сегментов с одного IP-адреса к общему количеству отправленных с этого же IP-адреса сегментов. Определяет объем сегментов с флагом ACK
tcp_fin	count_ip_events(tcp, fin) / count_ip_events(tcp, all)	Отношение количества отправленных с флагом FIN сегментов с одного IP-адреса к общему количеству отправленных с этого же IP-адреса сегментов. Определяет объем сегментов с флагом FIN
tcp_null	count_ip_events(tcp, null) / count_ip_events(tcp, all)	Отношение количества отправленных с флагом NULL сегментов с одного IP-адреса к общему количеству отправленных с этого же IP-адреса сегментов. Позволяет определить объем сегментов с флагом NULL
tcp_xmas	count_ip_events(tcp, xmas) / count_ip_events(tcp, all)	Отношение количества отправленных с флагом XMAS сегментов с одного IP-адреса к общему количеству отправленных с этого же IP-адреса сегментов. Определяет объем сегментов с флагом XMAS
tcp_maimon	count_ip_events(tcp, maimon) / count_ip_events(tcp, all)	Отношение количества отправленных с флагом MAIMON сегментов с одного IP-адреса к общему количеству отправленных с этого же IP-адреса сегментов. Определяет объем сегментов с флагом MAIMON
tcp_other	1.0 - tcp_syn - tcp_ack - tcp_fin - tcp_null - tcp_xmas - tcp_maimon	Отношение количества отправленных сегментов с другими флагами с одного IP-адреса к общему количеству отправленных с этого адреса сегментов. Определяет объем сегментов с другими флагами
uniq_ports	count_uniq_ports / count_ip_events(tcp, all)	Отношение количества уникальных портов, на которые были отправлены сегменты с одного IP-адреса, к общему количеству отправленных с этого адреса сегментов. Определяет перебор портов
flag	–	Флагу выставляется значение единица, если обнаружена сетевая разведка, или значение ноль, если трафик нормальный

Методика проведения эксперимента. Для создания датасета, основанного на реальном трафике, была выбрана топология, использованная для разработки датасета NTwPSA. С небольшими модификациями она внедрена на базе тестового сегмента корпоративной сети РУП «Белтелеком», что позволило создать реалистичный фоновый трафик (рис. 1).

Особенностью данной топологии является наличие двух атакующих хостов, проводящих сетевую разведку с различных ОС с использованием разного ПО, а также встроенного средства мониторинга и хранилища на самом тестовом ПК.

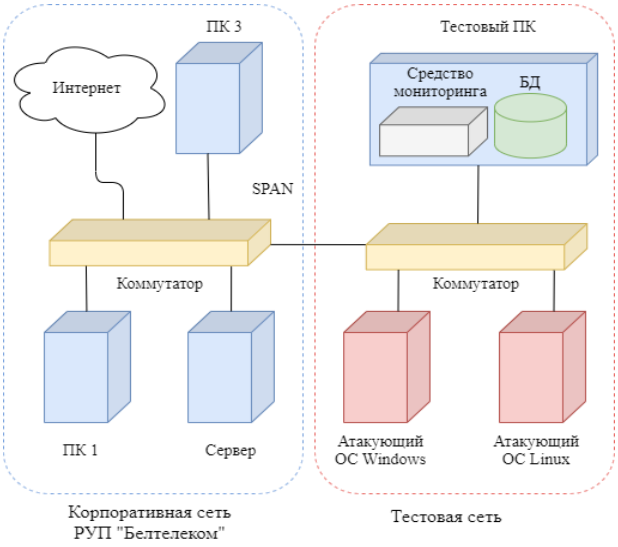


Рис. 1. Топология локальной сети для разработки датасета

Fig. 1. Local network topology for dataset development

В качестве средства мониторинга тестового ПК использовались разработанный для этой цели на языке Python версии 3 программный модуль (URL: https://github.com/ Moon1705/ dissertation) и открытая библиотека sklearn. Модуль разработан с применением свободно распространяемого ПО и находится в открытом доступе под лицензией MIT.

Изначально происходит импортирование библиотек json и numpy, предназначенных для работы с JSON-объектами и математическими исчислениями соответственно. После этого выполняется импортирование следующих классов из библиотеки sklearn: LabelEncoder для нормализации множества ответов Y; train_test_split для разделения датасета на обучающую и тестовую выборки; LogisticRegression для создания логистической регрессии; QuadraticDiscriminantAnalysis для создания алгоритма квадратичного дискриминантного анализа; SVC для реализации метода опорных векторов; KNeighborsClassifier в качестве метода ближайших соседей; GaussianNB для создания «наивного» байесовского классификатора; DecisionTreeClassifier для разработки дерева принятия решений; MLPClassifier для создания нейронной сети прямого распределения; BaggingClassifier для повышения точности алгоритма путем использования нескольких алгоритмов параллельно; AdaBoostClassifier для повышения точности алгоритма путем последовательного использования нескольких алгоритмов.

Далее происходит загрузка созданного JSON-датасета в оперативную память ПК и конвертирование его в двухмерную матрицу признаков. На данном этапе множество ответов представлено в формате строки (good, bad) и для дальнейшей работы требуется перевести их в бинарный формат (1, 0), что реализуется использованием класса LabelEncoder. Матрица признаков, содержащая перекодированные ответы, с помощью класса train_test_split разделяется на обучающее (80 %, или 800 событий) и тестовое (20 %, или 200 событий) множества с помощью функции псевдослучайных чисел. Анализ созданных множеств показал наличие 201 события сетевой разведки в обучающей выборке (25 %) и 49 событий в тестовом множестве (25 %), что позволяет судить о сбалансированности выборок.

Для генерации более реалистичного трафика на тестовом ПК был поднят статический сайт на основе веб-сервера nginx на 80-м порте. Также с самого компьютера происходили частые выходы в сеть Интернет на различные информационные ресурсы, что позволило создать стандартный трафик на тестируемом компьютере. Сам процесс сетевой разведки осуществлялся с использованием утилит Nmap (Windows) и Masscan (Linux). При этом компьютер с ОС Linux настроен таким образом, что через него можно провести Idle scan (зомби-сканирование). Параметры для представленных утилит приведены в табл. 3.

Таблица 3
Настройки утилит сетевой разведки

Table 3
Settings of network intelligence utilities

Утилита *Utility program*	Тип сканирования *Scan type*	Порты *Ports*	Дополнительные опции *Additional options*
Nmap	SYN (-sS)	1–1024	-Pn
	TCP (-sT)	1024–2048	
	UDP (-sU)	2048–3072	
	Null (-sN)	3072–4096	
	FIN (-sF)	4096–5120	
	Xmas (-sX)	5120–6144	
	ACK (-sA)	6144–7168	
	Windows (-sW)	7168–8192	
	Maimon (-sM)	8192–9216	
	Idle (-sI)	9216–10240	
	SYN (-sS)	10240–11264	-sV -O
	SYN (-sS)	–	-F -sV
Masscan	SYN	1–1024	-rate 1024

Результатом проведения указанных типов сканирования тестового компьютера с включенным средством мониторинга стал текстовый файл, содержащий порядка 500 событий сетевой разведки (исключая легитимный трафик). Приведем фрагмент файла, содержащий описание такого события:

```
[
 {
 "count": 0.4878, "tcp": 0.7919191, "tcp_ack": 0.5619191,
"tcp_fin": 0.0, "tcp_maimon": 0.24, "tcp_null": 0.0, "tcp_other": 0.0,
"tcp_syn": 0.0, "tcp_xmas": 0.0, "udp": 0.2080808, "uniq_ports": 0.94,
 },
 …
].
```

Для удобства применения значения каждой переменной были округлены до сотых, добавлено поле detection: bad (good), указывающее, что данное событие является сетевой разведкой, изменены наименования переменных (добавлено ratio_) и извлечены только уникальные события. В результате было получено 250 событий сетевой разведки и 750 событий легитимного трафика.

Результаты исследования эффективности алгоритмов машинного обучения и разработанного модуля. Все события являются уникальными и нормализованными. Графическое представление созданного датасета с использованием метода главных компонент (principal component analysis, PCA) показано на рис. 2.

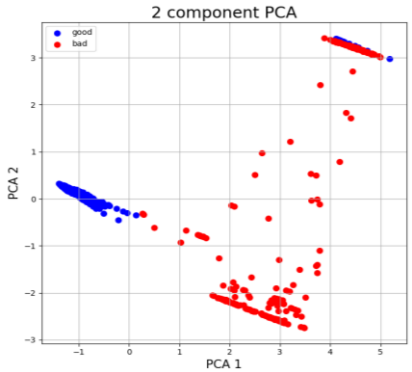


Рис. 2. Графическое представление датасета с помощью метода главных компонент
Fig. 2. Graphical representation of a dataset using the principal component analysis

На рис. 2 можно наблюдать дифференциацию датасета на три множества. Два из них представляют собой разделенный легитимный трафик и трафик сетевой разведки. Третье множество вызывает интерес по причине отсутствия явного разделения между аномалией, созданной злоумышленником, и обычным трафиком пользователей. Предположительно, это UDP-трафик, так как анализ числовых показателей датасета показал незначительное отличие bad-трафика от good-трафика. В общем случае сложности при анализе не возникнут, так как на корпоративных серверах редко допускается UDP-трафик. Тем не менее при разработке модели машинного обучения данный факт необходимо учесть и подобрать метод классификации, дающий минимальное отклонение в представленном множестве [8].

Из приведенных в табл. 4 данных видно, что на 100 % правильный результат дали четыре из семи исследуемых алгоритмов: SVM, K Neighbors, Decision Tree и MLP. При этом частично на их эффективность повлиял выбор отдельных параметров. Так, при использовании линейного и сигмоидного ядер в методе опорных векторов эффективность алгоритма уменьшается.

Таблица 4
Результаты исследования эффективности алгоритмов машинного обучения

Table 4
Results of the study of the effectiveness of machine learning algorithms

Алгоритм *Algorithm*	Параметр *Parameter*	Значение *Value*	Точность, % *Accuracy, %*	Время обучения, мс *Training time, ms*
Logistic Regression	solver	sag	97,0	4,052
		lbfgs	97,0	78,030
		liblinear	97,0	1,704
		saga	97,0	7,107
QDA	reg_param	0,0	95,0	58,084
		0,5	95,0	60,120
SVM	kernel	rbf	100,0	2,112
		poly	100,0	1,830
		linear	98,0	2,326
		sigmoid	95,5	5,304
K Neighbors	n_neighbors	3	100,0	1,458
	n_neighbors	5	100,0	1,409
	n_neighbors	10	100,0	1,351
	n_neighbors weights	3 distance	100,0	1,587
	n_neighbors weights	5 distance	100,0	1,504
	n_neighbors weights	10 distance	100,0	1,380
GaussianNB	var_smoothing	1e-9	97,5	0,892
	var_smoothing	1e-8	97,5	0,868
Decision Tree	criterion	gini	100,0	1,260
	criterion	entropy	100,0	1,086
	criterion splitter	gini random	100,0	0,912
	criterion splitter	entropy random	100,0	0,981
	criterion max_features	gini auto	99,5	0,888
	criterion max_features	entropy auto	99,5	1,653
	criterion splitter max_features	gini random auto	100,0	2,078
	criterion splitter max_features	entropy random auto	100,0	0,953

Окончание таблицы 4

End of table 4

Алгоритм *Algorithm*	Параметр *Parameter*	Значение *Value*	Точность, % *Accuracy, %*	Время обучения, мс *Training time, ms*
MLP	activation	logistic	97,5	922,823
	hidden_layer_sizes	(5, 3)		
	max_iter	500		
	solver	adam		
	activation	logistic	100,0	52,913
	hidden_layer_sizes	(5, 3)		
	max_iter	500		
	solver	lbfgs		
	activation	logistic	98,0	1263,257
	hidden_layer_sizes	(5, 3)		
	max_iter	1000		
	solver	adam		
	activation	logistic	100,0	55,881
	hidden_layer_sizes	(5, 3)		
	max_iter	1000		
	solver	lbfgs		
	activation	relu	97,5	910,756
	hidden_layer_sizes	(5, 3)		
	max_iter	500		
	solver	adam		
	activation	relu	98,0	744,577
	hidden_layer_sizes	(5, 3)		
	max_iter	500		
	solver	sgd		
	activation	relu	100,0	43,871
	hidden_layer_sizes	(5, 3)		
	max_iter	500		
	solver	lbfgs		
	activation	relu	99,0	1812,323
	hidden_layer_sizes	(5, 3)		
	max_iter	1000		
	solver	adam		
	activation	relu	98,0	1026,553
	hidden_layer_sizes	(5, 3)		
	max_iter	500		
	solver	sgd		
	activation	relu	100,0	44,346
	hidden_layer_sizes	(5, 3)		
	max_iter	500		
	solver	lbfgs		
	activation	relu	100,0	45,153
	hidden_layer_sizes	(7,)		
	max_iter	500		
	solver	lbfgs		
	activation	relu	97,0	552,070
	hidden_layer_sizes	(7,)		
	max_iter	1000		
	solver	adam		
	activation	relu	96,5	779,355
	hidden_layer_sizes	(7,)		
	max_iter	1000		
	solver	sgd		
	activation	relu	100,0	21,082
	hidden_layer_sizes	(7,)		
	max_iter	1000		
	solver	lbfgs		

Отдельным важным параметром является скорость работы алгоритма, определяющая возможность работы алгоритма в состоянии потока. Поэтому все алгоритмы с соответствующими параметрами, давшие наилучший результат, дополнительно оцениваются по скорости работы. Наиболее быстрым методом, обучающимся за 0,912 мс, является дерево принятия решения (Decision Tree) с параметрами criterion = gini и splitter = random. Дерево принятия решений было реализовано в виде программного кода на языке Python версии 3 (URL: https://github.com/Moon1705/dissertation). В результате тестирования данного кода значение метрики «точность» составило 100 %. Между тем существуют алгоритмы «наивного» байесовского классификатора (GaussianNB) и дерева принятия решений (Decision Tree) с параметрами criterion = = gini и max_features = auto, время работы которых меньше, чем у дерева принятия решения с параметрами criterion = gini и splitter = random, но с точностью менее 100 %. Было проведено улучшение этих алгоритмов с использованием бэггинга и бустинга, выполнен анализ эффективности и скорости работы после модернизации.

В табл. 5 представлены скорость и точность работы алгоритма до и после применения процедур бустинга и бэггинга, а также скорость работы разработанной функции. Из приведенных данных видно, что использование процедур бустинга и бэггинга не сказалось на эффективности «наивного» байесовского классификатора, это можно объяснить вероятностной природой его решения. Для дерева принятия решений применение процедур бустинга и бэггинга позволило достичь значения точности 100 % при увеличении времени проверки с 0,277 до 0,407 мс (бустинг) и 0,993 мс (бэггинг). Дерево принятия решений с параметрами criterion = gini и splitter = = random также показало точность 100 % при времени проверки 0,333 мс. Указанное время удалось уменьшить с 0,333 до 0,147 мс путем использования программной реализации (вместо обученной модели). Таким образом, метод дерева принятия решений с параметрами criterion = = gini и splitter = random показал лучшие результаты. Дополнительным плюсом использования программной реализации является меньшее количество потребляемых ресурсов.

Таблица 5
Эффективность процедур бустинга и бэггинга

Table 5
Efficiency of boosting and bagging procedures

Алгоритм *Algorithm*	Параметр *Parameter*	Значение *Value*	Метод *Method*	Точность, % *Accuracy, %*	Время обучения, мс *Training time, ms*	Время проверки, мс *Check time, ms*
GaussianNB	var_smoothing	1e-9	normal	97,5	1,245	0,421
			boosting	97,5	4,882	0,766
			bagging	97,5	8,709	1,932
		1e-8	normal	97,5	0,724	,387
			boosting	97,5	6,190	0,916
			bagging	97,5	8,531	1,886
Decision Tree	max_features	auto	normal	99,5	0,721	0,277
	criterion	gini				
	max_features	auto	boosting	100,0	1,606	0,407
	criterion	gini				
	max_features	auto	bagging	100,0	8,114	0,993
	criterion	gini				
	splitter	random	normal	100,0	0,833	0,333
	criterion	gini				
	splitter	random	code	100,0	0	0,147
	criterion	gini				

Результаты анализа практических испытаний разработанного модуля показали факт сработки на всех типах сканирования сетей с помощью утилит Nmap и Masscan. В процессе испытаний были отмечены некоторые особенности. При проведении Idle scan средство мониторинга показывало IP-адрес бота, а не реальный IP-адрес злоумышленника. Тем не менее на практике Idle scan – достаточно редкое явление, так как подобные уязвимости конфигурации отключены

по умолчанию. Замечены ложные сработки (false positive), связанные с широковещательными UDP-запросами DNS- и DHCP-серверов в корпоративной сети, которые приходили на закрытые порты защищаемого сервера. Этот факт стоит принять во внимание при использовании средств защиты в корпоративной сети и добавить серверы, содержащие сетевые службы, которые отправляют широковещательные запросы в список исключенных из анализа IP-адресов (параметр excluded_ips при конфигурации ПО).

Заключение. В результате анализа и отбора наиболее актуальных метрик сформирован датасет, состоящий из 1000 уникальных событий и включающий 250 событий сетевой разведки и 750 легитимных событий. Это выгодно отличает его от датасетов, в которых легитимный трафик преобладает (например, от датасета университета Лафборо) и при использовании которых сетевая разведка будет восприниматься не как аномалия, а как погрешность. Установлено, что для оперативного анализа с применением дерева принятия решений достаточно оперировать примерно половиной определенных ранее метрик. Это позволит ускорить потоковую обработку. Результаты проведенного исследования могут быть полезны при создании актуального и эффективного датасета для выявления признаков сетевой разведки.

Рассмотрены существующие методы классификации, их особенности и способы расчета. Выделены наиболее перспективные методы выявления признаков сетевой разведки, для них проведено обучение и тестирование с использованием различных гиперпара-метров. Наилучшие результаты показал метод дерева принятия решений с параметрами criterion = gini и splitter = = random. Проведена оптимизация отдельных алгоритмов с помощью процедур бэггинга и бустинга для ускорения их работы.

В программном виде реализованы модуль обнаружения признаков сетевой разведки, алгоритм дерева принятия решения и функция для переключения режимов работы модуля (обученное дерево принятия решений или алгоритм с возможностью дообучения модели).

Вклад авторов. *Н. П. Шараев* выполнил сравнительный анализ существующих подходов обнаружения признаков сетевой разведки, сформировал датасет, разработал программный модуль и провел экспериментальные исследования. *С. Н. Петров* определил цели исследования и задачи, которые необходимо было решить для их достижения, принял участие в интерпретации и обобщении полученных результатов.

Список использованных источников

1. Гущин, Р. А. Сетевая разведка / Р. А. Гущин, К. А. Колос ; сост. В. А. Мартинович // Материалы 74-й студ. науч.-техн. конф. – Минск : БНТУ, 2018. – С. 53–54.

2. Караулова, О. А. Оценка аномалий сетевого трафика на основе циклического анализа / О. А. Караулова, Н. В. Киреева // T-comm: телекоммуникации и транспорт. – 2018. – Т. 12, вып. 11. – С. 33.

3. Брюхомицкий, Ю. А. Искусственные иммунные системы в информационной безопасности : учеб. пособие / Ю. А. Брюхомицкий. – Ростов н/Д, Таганрог : Изд-во Южн. федер. ун-та, 2019. – 147 с.

4. Кашницкий, Ю. С. Ансамблевый метод машинного обучения, основанный на рекомендации классификаторов / Ю. С. Кашницкий, Д. И. Игнатов // Интеллектуальные системы. Теория и приложения. – 2015. – Т. 19, вып. 4. – С. 37–55.

5. Халкечев, Р. В. Бустинг – еще один способ машинного обучения [Электронный ресурс] / Р. В. Халкечев // Журнал «Яндекс Практикума». – Режим доступа: https://thecode.media/boosting/. – Дата доступа: 12.06.2021.

6. Detailed analysis of the KDD CUP 99 data set / M. Tavallaee [et al.] // 2009 IEEE Symp. on Computational Intelligence for Security and Defense Applications, Ottawa, Canada, 8–10 July 2009. – Ottawa, 2009. – P. 1–6.

7. Шараев, Н. П. Выявление и анализ признаков сетевой разведки методом машинного обучения / Н. П. Шараев, С. Н. Петров // Управление информационными ресурсами : материалы XVII Междунар. науч.-практ. конф., Минск, 12 мар. 2021 г. – Минск : Академия управления при Президенте Республики Беларусь, 2021. – С. 238–240.

8. Шараев, Н. П. Выявление сетевой разведки методами машинного обучения / Н. П. Шараев, С. Н. Петров // Защита информации : сб. материалов 57-й науч. конф. аспирантов, магистрантов и студентов БГУИР, Минск, Беларусь, 19–23 апр. 2021 г. – Минск : БГУИР, 2021. – С. 34–37.

References

1. Gushchin R. A., Kolos K. A. *Network intelligence*. Materialy 74-j studencheskoj nauchno-tekhnicheskoj konferencii [*Materials of the 74th Student Scientific and Technical Conference*], sostavitel' V. A. Martinovich, Minsk, Belorusskij nacional'nyj tekhnicheskij universitet, 2018, pp. 53–54 (In Russ.).

2. Karaulova O. A., Kireeva N. V. *Estimation of network traffic anomalies based on cyclic analysis*. T-comm: telekommunikacii i transport [*T-comm: Telecommunications and Transport*], 2018, vol. 12, no. 11, p. 33 (In Russ.).

3. Bryuhomickij, Yu. A. Iskusstvennye immunnye sistemy v informacionnoj bezopasnosti. *Artificial Immune Systems in Information Security*. Rostov-on-Don, Taganrog, Izdatel'stvo Yuzhnogo federal'nogo universiteta, 2019, 147 p. (In Russ.).

4. Kashnickij Yu. S., Ignatov D. I. *An ensemble method of machine learning based on the recommendations of classifiers*. Intellektual'nye sistemy. Teoriya i prilozheniya [*Intelligent Systems. Theory and Applications*], 2015, vol. 19, no. 4, pp. 37–55 (In Russ.).

5. Halkechev R. V. *Boosting is another way of machine learning*. Zhurnal "Yandeks Praktikuma" [*Yandex Practicum Magazine*] (In Russ.). Available at: https://thecode.media boosting/ (accessed 12.06.2021).

6. Tavallaee M., Bagheri E., Lu W., Ghorbani A. Detailed analysis of the KDD CUP 99 data set. *2009 IEEE Symposium on Computational Intelligence for Security and Defense Applications, Ottawa, Canada, 8–10 July 2009*. Ottawa, 2009, pp. 1–6.

7. Sharaev N. P., Petrov S. N. *Identification and analysis of signs of network intelligence by machine learning*. Upravlenie informacionnymi resursami : materialy XVII Mezhdunarodnoj nauchno-prakticheskoj konferencii, Minsk, 12 marta 2021 g. [*Information Resource Management: Materials of the* XVII *International Scientific and Practical Conference, Minsk, 12 March 2021*], Minsk, Akademija upravlenija pri Prezidente Respubliki Belarus', 2021, pp. 238–240 (In Russ.).

8. Sharaev N. P., Petrov S. N. *Identification of network intelligence by machine learning methods*. Zashhita informacii : sbornik materialov 57-j nauchnoj konferencii aspirantov, magistrantov i studentov BGUIR, Minsk, Belarus', 19–23 aprelja 2021 g. [*Protection of Information: Collection of Materials of the 57th Scientific Conference of Postgraduates, Undergraduates and Students of BSUIR, Minsk, Belarus, 19–23 April 2021*], Minsk, Belorusskij gosudarstvennyj universitet informatiki i radiojelektroniki, 2021, pp. 34–37 (In Russ.).

Информация об авторах

Шараев Никита Петрович, магистрант кафедры защиты информации, факультет инфокоммуникаций, Белорусский государственный университет информатики и радиоэлектроники.
E-mail: nick.moon.1705@gmail.com

Петров Сергей Николаевич, кандидат технических наук, доцент, доцент кафедры защиты информации, факультет инфокоммуникаций, Белорусский государственный университет информатики и радиоэлектроники.
E-mail: sergpetrov@inbox.ru

Information about the authors

Nikita P. Sharaev, Master Student of the Information Security Department, Faculty of Infocommunications, Belarusian State University of Informatics and Radioelectronics.
E-mail: nick.moon.1705@gmail.com

Sergei N. Petrov, Ph. D. (Eng.), Associate Professor, Associate Professor of the Information Security Department, Faculty of Infocommunications, Belarusian State University of Informatics and Radioelectronics.
E-mail: sergpetrov@inbox.ru

УДК 004.832.32
https://doi.org/10.37661/1816-0301-2021-19-1-32-49

Оригинальная статья
Original Paper

Физически неклонируемые функции с управляемой задержкой распространения сигналов

В. Н. Ярмолик✉, А. А. Иванюк, Н. Н. Шинкевич

Белорусский государственный университет информатики и радиоэлектроники,
ул. П. Бровки, 6, Минск, 220013, Беларусь
✉*E-mail: yarmolik10ru@yahoo.com*

Аннотация

Цели. Решается задача построения нового класса физически неклонируемых функций (ФНФ), обеспечивающих управление задержкой распространения сигнала через элементы, которые расположены на пути его распространения. Актуальность такого исследования связана с активным развитием физической криптографии. В работе преследуются следующие цели: построение базовых элементов ФНФ и их модификаций, разработка методики построения управляемых кольцевых осцилляторов на базе элементов XOR и управляемых кольцевых осцилляторов, основанных на многовходовом переключении сигнала.

Методы. Используются методы синтеза и анализа цифровых устройств, в том числе на программируемых логических интегральных схемах, основы булевой алгебры и схемотехники.

Результаты. Показано, что комбинированные ФНФ, основанные на RS-триггерах, реализуют идею управления задержкой сигнала за счет выбора пути, который представляет собой последовательно подключенные элементы, выбранные в соответствии с запросом ФНФ. Разработана методика построения ФНФ с управляемой задержкой через каждый элемент пути. Исследованы особенности и свойства ФНФ с управляемой задержкой сигналов типа кольцевого осциллятора и показаны возможные решения для случая двухразрядных входных запросов. Предложен базовый элемент и его модификации для построения новых структур ФНФ, основанных на управлении задержкой распространения сигнала. Показано, что задержка сигнала через базовый элемент, представляющий собой многовходовый элемент XOR, зависит не только от количества входов, на которые подается активный входной сигнал, но и от фиксированного значения 0 либо 1 на остальных его входах. Приведена новая структура ФНФ – управляемый кольцевой осциллятор, рассматриваются его реализации для случая управления за счет задания количества входов, на которых изменяется активный входной сигнал.

Заключение. Предложенный подход к построению физически неклонируемых функций, основанный на управлении задержкой сигналов через логические элементы, показал свою работоспособность и перспективность. Экспериментально подтвержден эффект влияния на задержки распространения сигналов через логический элемент количества его входов, на которых изменяются входные сигналы, приводящие к изменению выходного сигнала. Перспективным представляется дальнейшее развитие идей построения управляемых кольцевых осцилляторов и осцилляторов с многовходовым переключением сигнала, а также создания новых структур ФНФ типа арбитр.

Ключевые слова: физическая криптография, физически неклонируемые функции, физические однонаправленные функции, кольцевой осциллятор, физически неклонируемая функция типа арбитр

Благодарности. Авторы выражают искреннюю благодарность резиденту Парка высоких технологий компании SK Hynix Memory Solutions Eastern Europe за предоставленное оборудование для проведения экспериментов в рамках работы совместной учебной лаборатории с Белорусским государственным университетом информатики и радиоэлектроники.

Для цитирования. Ярмолик, В. Н. Физически неклонируемые функции с управляемой задержкой распространения сигналов / В. Н. Ярмолик, А. А. Иванюк, Н. Н. Шинкевич // Информатика. – 2022. − Т. 19, № 1. – С. 32–49. https://doi.org/10.37661/1816-0301-2022-19-1-32-49

Конфликт интересов. Авторы заявляют об отсутствии конфликта интересов.

Поступила в редакцию | Received 17.12.2021
Подписана в печать | Accepted 05.01.2022
Опубликована | Published 29.03.2022

Physically unclonable functions with controlled propagation delay

Vyacheslav N. Yarmolik✉, Alexander A. Ivaniuk, Natallia N. Shynkevich

Belarusian State University of Informatics and Radioelectronics,
st. P. Brovki, 6, Minsk, 220013, Belarus
✉*E-mail: yarmolik10ru@yahoo.com*

Abstract

Objectives. The problem of constructing a new class of physically uncloneable functions (PUF) based on controlling the signal propagation delay through the elements lying on the path of its propagation is being solved. The relevance of this problem is associated with the active development of physical cryptography. For its implementation, the following goals are pursued: the construction of the basic elements of the PUF and their modifications, the development of a methodology for constructing controlled ring oscillators based on XOR elements and controlled ring oscillators based on multi-input signal switching.

Methods. Methods of synthesis and analysis of digital devices were used, including those based on programmable logic integrated circuits (FPGA), the basics of Boolean algebra and circuitry.

Results. It is shown that combined PUFs based on RS-flip-flops implement the idea of controlling the signal delay by choosing a path, which is a series-connected elements selected in accordance with the PUF request. A technique for constructing an PUF with a controlled delay through each element of the path has been developed as a development of the idea of controlling the signal delay along the path. The features and properties of PUF with controlled delay of signals of the ring oscillator type are investigated and possible solutions are shown for the case of two-bit input requests. A basic element and its modifications are proposed for constructing new PUF structures based on the control of the signal propagation delay. It is shown that the signal delay through the basic element, which is a multi-input XOR element, depends not only on the number of inputs to which the active input signal is applied, but also on fixed values of 0 or 1 at its other inputs. A new PUF structure is presented, namely, a controlled ring oscillator, its implementation is considered for the case of control by setting the inputs and their number, by which the active input signal changes.

Conclusion. The proposed new approach to the construction of physically uncloneable functions, based on the control of signal delay through logical elements, has shown its efficiency and promise. The effect of the influence on the delays of signal propagation through the logic element, both the number of its inputs, along which the input signals change, leading to a change in the output signal, and their composition, is experimentally confirmed. It seems promising to further developing the ideas of constructing controlled ring oscillators and oscillators with multi-input switching of input signal, as well as the creation of new PUF structures of arbiter type.

Keywords: physical cryptography, physically unclonable functions, physical one-way functions, ring oscillator, arbiter-based physically unclonable function

Acknowledgements. The authors express their sincere gratitude to the HTP resident of the "SK Hynix Memory Solutions Eastern Europe" company for the equipment provided for carrying out experiments within the framework of the joint laboratory with the Belarusian State University of Informatics and Radioelectronics.

For citation. Yarmolik V. N., Ivaniuk A. A., Shynkevich N. N. *Physically unclonable functions with controlled propagation delay*. Informatika [*Informatics*], 2022, vol. 19, no. 1, pp. 32−49 (In Russ.).
https://doi.org/10.37661/1816-0301-2022-19-1-32-49

Conflict of interest. The authors declare of no conflict of interest.

Введение. Понятие *физически неклонируемых функций* было сформулировано R. Pappu в 2001 г. в работе [1], где была впервые определена концепция физических однонаправленных функций (Physical One-Way Functions). Практически одновременно B. Gassend и др. предложили реализацию кремниевых *физических случайных функций* (Physical Random Functions) [2]. Данные термины были сформулированы исторически первыми, однако в настоящее время в основном употребляется понятие «физически неклонируемые функции» (от англ. Physical Unclonable Functions, PUF). До сих пор отсутствует однозначное определение ФНФ. На практике одним из широко используемых является предложенное U. Rührmair с соавторами определение, согласно которому ФНФ представляют собой физические системы (устройства), обладающие неотъемлемым свойством неклонируемости некоторых их характеристик либо, чаще всего, параметров [3].

Основополагающее свойство неклонируемости ФНФ подразумевает, что в результате производственного процесса не получается создать два идентичных физических устройства, обладающих одинаковыми характеристиками. Изменения возникают из-за несовершенства производственного процесса и варьируются от изготовителя к изготовителю. Вместе с тем внутренние вариации физических устройств предопределены и ограничены фундаментальной физикой материалов. Они присущи структуре цифровых устройств и считаются одним из основных узких мест при тиражировании базовых элементов таких изделий. Собственные вариации цифровых устройств в целом обусловлены случайными колебаниями различного рода примесей используемых материалов, шероховатостью (неравномерностью) кромок линий соединений и компонентов (транзисторов, сопротивлений) элементов, колебаниями толщины оксида и другими причинами, где влияние со стороны изготовителя затруднено либо вообще невозможно. Эти источники вариаций вызывают изменения в значениях параметров элементов устройства и его временных задержек [4–6]. Такое влияние продолжает расти с уменьшением размера элементов цифровых устройств и в связи с изменениями технологических норм (табл. 1) [7].

Таблица 1
Математическое ожидание задержки μ инвертора и его относительная девиация σ/μ в зависимости от технологических норм

Table 1
Delay mean μ *of the inverter and its deviation* σ/μ *with respect to* μ *depending on technological nodes*

Технология, нм *Technology, nm*	Значение задержки (μ), пс *Delay mean* (μ), *ps*	Сигма (σ/μ), % *Sigma* (σ/μ), *%*
12	1,7	21
16	1,8	13
22	2,35	7
32	2,75	1,6
45	3,15	1,4

Данные табл. 1 показывают динамику изменения номинальной задержки μ инвертора, изготовленного по КМОП-технологии, и вариации задержки с изменением технологических норм (масштабированием). Задержки указаны в пикосекундах (пс), а технологические нормы в нанометрах (нм). Сигма (σ) представляет собой среднеквадратическое отклонение задержки. Величина σ/μ оценивает отклонение задержки сигнала от средней величины в процентах. Как видно из табл. 1, задержка уменьшается с увеличением масштабирования (уменьшением технологических норм), но ее отклонение в процентах от среднего быстро растет из-за увеличения влияния различных факторов, которые носят случайную природу. Это означает, что масштабирование

технологии обеспечивает увеличение быстродействия, но из-за роста разбросов параметров и в первую очередь задержек элементов снижает надежность как самих элементов, так и цифровых устройств на их основе.

Увеличение разбросов величин случайных значений задержки сигнала через логический элемент свидетельствует об увеличении ее непредсказуемости. Уникальность задержек и их изменяемость от элемента к элементу являются основой создания множества различных типов ФНФ для цифровых схем [8–14].

В работе [8] впервые было предложено использовать различие в задержке распространения сигнала по симметричным путям для реализации ФНФ типа арбитр (Arbiter PUF). Понятие «путь цифрового устройства» означает последовательное подключение друг к другу логических элементов, каждый из которых характеризуется задержкой распространения сигнала через элемент. В качестве альтернативы ФНФ типа арбитр в работе [9] изучено применение двух раздельных множеств путей, когда первый путь пары выбирается из первого множества элементов, а второй путь – из второго множества. Предложенная схема использует вариации задержек буфера с тремя состояниями и строится путем последовательного подключения пар буферов [9]. Разность частот кольцевых осцилляторов (Ring Oscillator PUF) [2], а также уникальность значений частот (Bistable Ring PUF) [10] были использованы в качестве основы для генерирования пар «запрос – ответ». Много реализаций ФНФ основано на применении состояния элементов памяти после инициализации на базе статического оперативного запоминающего устройства [11] и динамической памяти с произвольным доступом [12]. Рассматривались также другие подходы для создания ФНФ [13, 14], однако методологической основой большинства из них для случая интегральных цифровых схем является создание цифрового устройства, выходное значение которого определяется случайными значениями временных параметров (задержек) сигналов кремниевой подложки. Благодаря технологическим вариациям изготовления цифровых устройств время задержки сигналов по определенному пути (элементу) цифрового устройства будет незначительно отличаться от цифрового устройства к цифровому устройству и от кристалла к кристаллу, несмотря на идентичность их функциональности и топологии.

Все известные решения при создании ФНФ основаны на парадигме, согласно которой задержка по конкретному пути (элементу) имеет случайное, но неизменное и неуправляемое значение, за исключением влияния внешних факторов (температуры, давления, электромагнитного излучения и др.) и временной деградации. Это справедливо только для одновходовых элементов типа инвертора и повторителя, а также для путей, состоящих из последовательно подключенных инверторов и повторителей. Для всех остальных логических элементов с количеством входов не менее двух задержка через элемент отличается для различных входов так же, как и для разных режимов изменения входных его значений. Более того, на уровне элемента оказывается возможным управлять задержкой прохождения сигнала. Задержки через элемент принимают случайные, но в то же время неизменные значения из ограниченного множества при допущении отсутствия влияния внешних факторов и изменения их параметров с течением времени.

В настоящей статье решается задача построения нового класса ФНФ, основанного на управлении задержкой распространения сигнала через элементы, которые лежат на пути его распространения. Базовый элемент ФНФ обеспечивает управляемость задержкой за счет выбора количества входов, влияющих на изменение выходного сигнала, и значений на неактивных входах. Используя базовые элементы и их модификации, предлагаются методики построения управляемого кольцевого осциллятора на элементах XOR и управляемых кольцевых осцилляторов с функцией многовходового переключения сигнала.

Задержки логических элементов. При разработке различных схемотехнических решений по созданию ФНФ ключевым фактором является задержка прохождения сигнала через логический элемент. Считается, что задержка сигнала (как правило, импульсного) принимает случайное непредсказуемое значение в рамках определенного временного интервала. Данный интервал (среднее значение задержки) для каждого логического элемента определяется типом ис-

пользованных в них электронных элементов (например, ТТЛ, ЭСЛ, КМОП), принципиальной схемой логического элемента, технологическими нормами и особенностями процесса его изготовления.

Большинство работ по ФНФ основано на том, что изначально принимается гипотеза о фиксированном значении задержки через элемент без учета многих факторов (см., например, [4]). Эта задержка в процессе изготовления логического элемента принимает случайное значение, которое остается неизменным при функционировании ФНФ. На задержки влияют только внешние и временные факторы, они же рассматриваются разработчиками ФНФ как нежелательные эффекты.

Результатом изготовления простейшего логического элемента наподобие повторителя или инвертора будут являться два произвольных возможных значения задержек. Существует сложная и неоднозначная зависимость между этими двумя переменными, требующая глубокого и детального анализа и понимания на уровне физики процессов функционирования логического элемента. Задержки сигнала определяются на уровне 50 % размаха входного и выходного сигналов (Input and Output Signal) и обозначаются при переходе выходного сигнала из низкого уровня в высокий как Δ_{LH}, а при переходе из высокого уровня в низкий – как Δ_{HL} (рис. 1) [15].

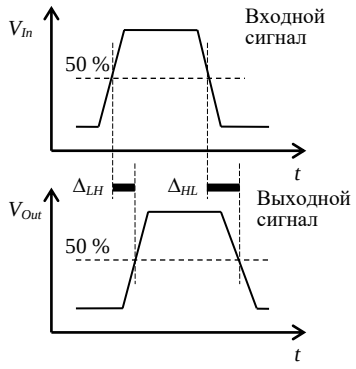


Рис. 1. Задержки Δ_{LH} и Δ_{HL} фронтов выходного импульсного сигнала

Fig. 1. Delays Δ_{LH} and Δ_{HL} of the edges of output pulse signal

Изображенные на рис. 1 временные диаграммы показывают эффект задержки сигнала через повторитель. Значения задержек фронтов (переходов) логического сигнала из 0 в 1 (Low to High, *LH*) либо из 1 в 0 (High to Low, *HL*), как правило, отличаются и в сильной степени зависят от типа электронных элементов, используемых для их изготовления. Например, времена указанных задержек сигнала при переключении КМОП-инвертора можно оценить соотношениями [15]

$$\Delta_{HL} \approx 0{,}7 \cdot R_n \cdot (C_{Oup} + C_{Load}), \quad \Delta_{LH} \approx 0{,}7 \cdot R_p \cdot (C_{Oup} + C_{Load}), \tag{1}$$

где C_{Oup} – эффективная выходная емкость инвертора, а C_{Load} – нагрузочная емкость инвертора, R_n – эффективное сопротивление *n*-канального, а R_p – *p*-канального МОП-транзистора. Разные значения эффективных сопротивлений R_n и R_p свидетельствуют об отличии времен задержек Δ_{LH} и Δ_{HL} инвертора, а технологические девиации при изготовлении МОП-транзисторов ведут к возникновению задержек с произвольными значениями в пределах своего временного интервала.

В справочной литературе для каждого логического элемента в основном приводятся средние (типовые) и (или) максимальные значения задержек Δ_{LH} и Δ_{HL} [16]. Так, для микросхемы К155ЛН1, содержащей шесть инверторов ТТЛ серии К155, указываются максимальное значение Δ_{LH} = 15 нс и максимальное значение Δ_{HL} = 22 нс [16].

Обобщая значения задержек Δ_{LH} и Δ_{HL} для произвольного логического элемента, определяется средняя их величина $\Delta_G = (\Delta_{LH} + \Delta_{HL})/2$, которая интерпретируется как время распространения сигнала через элемент и используется при анализе и синтезе ФНФ. В большинстве рассуж-

дений о случайности величины задержки через элемент либо путь, созданный последовательным подключением элементов, авторы оперируют задержкой Δ_G.

Как уже отмечалось, случайное значение задержки логического элемента зависит от многих факторов. Вместе с тем для конкретной принципиальной схемы элемента и специфики его изготовления у производителя задержки сигналов всегда можно описать стандартными математическими моделями, например законом распределения и численными характеристиками. Весьма важным является то, какие задержки наиболее значимы для целей построения ФНФ.

Рассмотрим задержки сигналов как источники случайности и непредсказуемости на примере простейших логических элементов. Задержки на логических элементах чаще всего исследуются в режиме переключения сигнала по одному входу элемента (Single Input Switching, SIS), изменения которого приводят к изменению выходного значения [15–17].

В работе [7] была изучена зависимость временной задержки от входа, на который подается сигнал, вызывающий изменение на выходе, для элементов, изготовленных по КМОП-технологии. Все факторы, влияющие на задержку (геометрия транзисторов, емкость нагрузки (C_{Load}), скорость нарастания входного сигнала и др.), учитываются в предложенной модели. В случае элемента 2И-НЕ с двумя входами *In*1 и *In*2 как результаты моделирования, так и аналитические расчеты показывают заметное отличие задержек в зависимости от входа, на который подается активный сигнал (табл. 2).

Таблица 2
Вариации математического ожидания задержки μ,
среднеквадратического отклонения σ и его относительная девиация σ/μ
в зависимости от активного входа элемента 2И-НЕ

Table 2
Variations in the mean μ *of the delay, the standard deviation* σ
and its relative deviation σ/μ *depending on active input of the 2NAND gate*

Вход *Input*	Результаты моделирования *Simulation results*			Аналитические результаты *Analytical results*		
	μ, пс	σ, пс	σ/μ, %	μ, пс	σ, пс	σ/μ, %
*In*1	15,60	1,68	10,77	13,86	1,17	8,43
*In*2	16,50	0,91	5,52	16,14	0,97	6,01

Более сложные процессы и, соответственно, зависимости задержек через элемент возникают в случае переключения сигналов одновременно на нескольких входах (Multi Input Switching, MIS) [15, 18]. Это происходит, например, при одновременном переключении сигналов из 1 в 0 и из 0 в 1 на обоих входах *In*1 и *In*2 элемента 2И-НЕ.

В качестве примера можно привести *m*-входовый логический элемент И-НЕ, изготовленный по КМОП-технологии. При нагрузочной емкости C_{Load} задержка Δ_{LH} распространения сигнала через элемент И-НЕ с *m* входами оценивается следующими выражениями [15]:

$$\Delta_{LH} \approx 0,7 \cdot \frac{R_p}{m} \cdot (m \cdot C_{Oup} + C_{Load}),$$
$$\Delta_{LH} \approx 0,7 \cdot R_p \cdot (m \cdot C_{Oup} + C_{Load}). \quad (2)$$

Здесь первое соотношение для Δ_{LH} приведено для случая переключения всех *m* входов элемента И-НЕ одновременно, а второе – только одного входа. Видно, что при переключении логического значения только на одном входе значение задержки Δ_{LH} в *m* раз больше по сравнению со случаем переключения на всех *m* входах [15].

В общем случае задержка сигнала при прохождении через логический элемент зависит от трех аргументов: пути прохождения сигнала, определяемого активным входом (входами); логических значений на неактивных входах (входе) и самих входных сигналов; задержки изменения выходного сигнала из 0 на 1 (*LH*) либо из 1 на 0 (*HL*). Подчеркнем, что временные задержки связаны с задержками переключения входного сигнала на противоположное значение, вызывающее изменение выходного сигнала. В справочной литературе обычно ука-

зываются значения оценок величин Δ_{LH} и Δ_{HL} либо их среднее значение Δ_G. Для серии К155 элементной базы ТТЛ справочными значениями для двухвходовых элементов 2XOR микросхемы К155ЛП5 являются только максимальные значения Δ_{LH} = 30 нс и Δ_{HL} = 22 нс [16].

На примере простейшего двухвходового элемента 2XOR с входами *In*1, *In*2 и выходом *Out* показано многообразие задержек прохождения сигнала, которые в процессе изготовления принимают непредсказуемые случайные значения и могут быть использованы при построении ФНФ (табл. 3).

Таблица 3
Задержки сигнала на элементе 2XOR
Table 3
Signal delays on 2XOR element

Задержка *Delay*	*In*1	*In*2	*Out*	Задержка *Delay*	*In*1	*In*2	*Out*
$\Delta_1(LH)$	*LH*	0	*LH*	$\Delta_5(HL)$	*HL*	0	*HL*
$\Delta_2(HL)$	*LH*	1	*HL*	$\Delta_6(LH)$	*HL*	1	*LH*
$\Delta_3(LH)$	0	*LH*	*LH*	$\Delta_7(HL)$	0	*HL*	*HL*
$\Delta_4(HL)$	1	*LH*	*HL*	$\Delta_8(LH)$	1	*HL*	*LH*

В табл. 3 приведены описания задержек сигнала на элементе 2XOR в режиме SIS. Например, $\Delta_2(HL)$ представляет собой задержку изменения сигнала из 1 в 0 (*HL*) на выходе *Out* при изменении входного сигнала *In*1 из 0 в 1 (*LH*) при удержании на втором входе *In*2 логической единицы. Таким образом, даже в случае простейшего двухвходового элемента XOR можно получить восемь случайных значений задержек выходного сигнала *Out*. Эти задержки зависимы между собой, так как формируются под влиянием общих факторов, присущих данному логическому элементу и его принципиальной схеме, и материалов, примененных для его реализации. Однако на каждую конкретную задержку из восьми значений, присущих элементу 2XOR, оказывают влияние различные случайные факторы в разной комбинации и в разной степени, как это следует, например, из приведенного ранее выражения (1).

Использование элементов с большим числом входов позволяет существенно расширить множество задержек, принимающих случайные значения, в первую очередь за счет режима MIS, когда одновременно изменяются значения на нескольких входах. Элемент 3И в режиме MIS при одновременном переключении сигнала формирует две задержки на трех его входах и шесть задержек на двух входах. Элемент XOR с тремя входами генерирует 24 случайные задержки в режиме SIS и 8 случайных задержек в режиме MIS при переключении значений на всех трех входах, т. е. всего 32 различающихся значения задержек выходного сигнала. Это объясняется спецификой элемента XOR, у которого изменение выходного сигнала возможно только при изменении четности значений на его входах.

Отметим, что каждая из указанных величин задержки может быть использована при построении ФНФ типа арбитр, а их многообразие позволяет управлять задержками – выбирать одну задержку из восьми (табл. 3).

Таким образом, для построения ФНФ с управляемыми задержками распространения сигнала первоначально необходима разработка базовых элементов, которые реализуют функцию управления задержкой, заключающуюся в выборе одного из возможных значений задержки.

Базовый элемент физически неклонируемых функций. Альтернативой построения ФНФ является создание автономных булевых сетей (АБС) [19, 20]. При проектировании АБС выдвигаются условия, противоположные условиям для ФНФ, а при их создании обеспечивается максимально возможная изменчивость и непредсказуемость выходных значений. Повторение измерений для АБС приводит к другим неповторяющимся выходным значениям. В ФНФ повторяемость выходных значений, называемых ответами (Responses, *R*), для одних и тех же значений запроса (Challenge, *C*) обязательна и является неотъемлемым их свойством [1, 8–14]. Вместе с тем основой непредсказуемого поведения АБС так же, как и в случае ФНФ, являются внутренние случайные вариации параметров элементов и их сочетания. Большинство решений

для получения хаотического поведения таких устройств состоит в использовании путей задержки в АБС [19, 20]. Составные компоненты АБС выполняют асимметричные логические операции, для построения которых применяются комбинации элементов логического исключающего ИЛИ (XOR) и исключающего ИЛИ с отрицанием (XNOR) [19]. Элементы XOR и XNOR имеют несколько входов, для каждого из которых искусственно (путем подключения цепочки последовательно соединенных инверторов) обеспечивается различие временных задержек.

Как и в случае с АБС, в качестве основного базового элемента для построения ФНФ используем многовходовый элемент XOR. На рис. 2, *a* изображена обобщенная структура базового элемента, обеспечивающего режим переключения своего выходного значения *Out* при прохождении входного сигнала одновременно по нескольким входам XOR. Элемент XOR базового элемента имеет $m + 1$ вход. На первый вход (*In*) элемента XOR подается входной единичный импульс, при отсутствии входного импульсного сигнала выходное значение *Out* равняется нулю.

Запрос C формируется на входах базового элемента, в качестве которого применяется m-разрядный двоичный вектор $C = c_0\, c_1 \ldots c_{m-1}$, где $c_j \in \{0, 1\}$, $j \in \{0, 1, \ldots, m - 1\}$. В зависимости от данного запроса базовый элемент реализует один из режимов MIS. Количество ненулевых значений c_j запроса C определяет число входов XOR, через которые проходит входной импульс. Отметим, что количество единичных значений c_j для базового элемента, использующего XOR, должно быть четным, что в два раза уменьшает количество возможных значений запроса C.

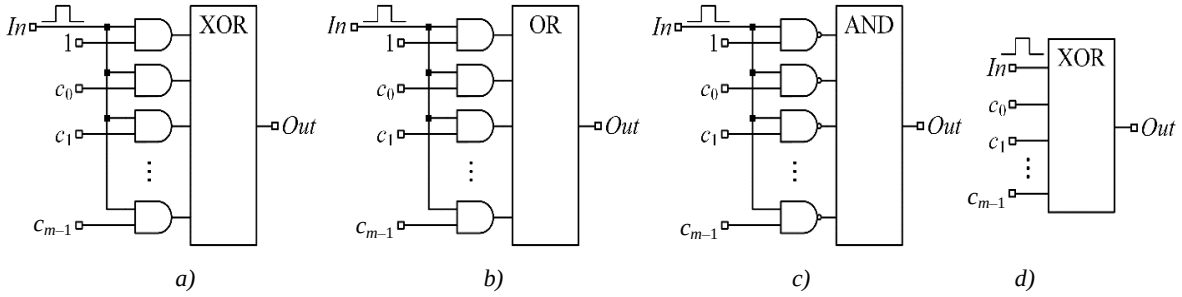


Рис. 2. Базовый элемент: *a)* структура; *b)* и *c)* модификации; *d)* режим прямого управления задержкой
Fig. 2. Basic element: a) structure; b) and c) modification; d) mode of direct control of the delay

Модификации базового элемента, использующие элементы ИЛИ (OR) и И (AND) (рис. 2, *b* и *c*), не накладывают данного ограничения на значения запроса C. В этих случаях вектор значений запроса $C = c_0\, c_1 \ldots c_{m-1}$ может принимать одно из 2^m возможных значений.

Основная идея предлагаемых в настоящей статье решений заключается в том, что для каждого запроса C базовый элемент будет иметь свое уникальное значение задержки прохождения входного импульса, определяемого режимом MIS, который задается значением вектора запроса $C = c_0\, c_1 \ldots c_{m-1}$.

Уникальность элемента XOR позволяет напрямую управлять задержкой прохождения через него входного импульса путем задания произвольных входных значений по остальным его входам (рис. 2, *d*). В данном случае выходное значение изменяется на противоположное в режиме SIS, а уникальность запроса C определяет уникальное значение задержки сигнала через элемент XOR.

Комбинированные физически неклонируемые функции. Как отмечалось ранее [11–13], статические оперативные запоминающие устройства (СОЗУ) широко используются для реализации ФНФ. Запоминающий элемент СОЗУ (ячейка) всегда находится в одном из двух состояний, что, в свою очередь, позволяет использовать его для хранения одного бита информации. Примером такой ячейки может служить RS-триггер, реализованный на двух логических элементах 2И-НЕ (рис. 3, *a*) [13].

Схема RS-триггера построена таким образом, что позволяет комбинационной схеме с положительной обратной связью хранить требуемое значение (0 или 1). Функционирование подобной схемы может быть описано с помощью таблицы переходов, однозначно определяющей функционирование RS-триггера.

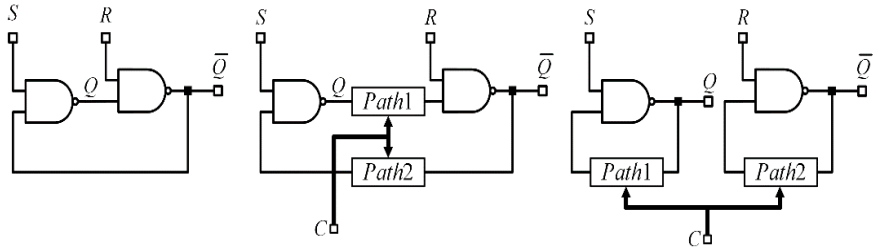


Рис. 3. Схема RS-триггера (*a*), комбинированные ФНФ (*b* и *c*)
Fig. 3. RS-flip-flop circuit (*a*), *combined PUF* (*b and c*)

Входные значения S и R могут принимать любую из четырех возможных комбинаций: 0 0, 0 1, 1 0 или 1 1 [13]. Для обозначения безразличного значения входных сигналов S и R используется набор $X\,X \in \{0\,0, 0\,1, 1\,0, 1\,1\}$, а для выходных значений – набор $Q\ \overline{Q} \in \{0\,1, 1\,0, 1\,1\}$. Отметим, что для RS-триггера (рис. 3, *a*) на выходах Q и $\overline{Q}$ получение комбинаций $Q = 0$ и $\overline{Q} = 0$ невозможно в статическом режиме функционирования.

Высокий уровень (соответствующий логической единице) подаваемого одновременно на входы S и R RS-триггера сигнала позволяет сохранять предыдущее состояние Q, равное 0 либо 1 и определяемое последней операцией записи в данную запоминающую ячейку. Эксперименты показывают, что при включении питающего напряжения все ячейки СОЗУ устанавливаются в одно из двух возможных состояний: $Q = 0$ либо $Q = 1$. В силу симметрии RS-триггера, реализующего ячейку СОЗУ, неизвестно, какое конечное состояние Q примет ячейка: 0 или 1 [13]. В общем случае это состояние является случайным и определяется множеством факторов [11]. Эмуляцией включения питания в случае RS-триггера является последовательная подача на его входы S и R комбинаций 0 0 и 1 1 (табл. 4).

Таблица 4
Таблица переходов RS-триггера
Table 4
RS-trigger transition table

Текущие значения на входах S и R *Current values at inputs S and R*	Следующие значения на входах S и R *Next values at inputs S and R*	Текущее состояние $Q\ \overline{Q}$ *Current values* $Q\ \overline{Q}$	Следующее состояние $Q\ \overline{Q}$ *Next values* $Q\overline{Q}$
$S\,R = X\,X$	0 0	$Q\ \overline{Q}$	1 1
$S\,R = X\,X$	0 1	$Q\ \overline{Q}$	1 0
$S\,R = X\,X$	1 0	$Q\ \overline{Q}$	0 1
$S\,R \neq 0\,0$	1 1	$Q\ \overline{Q}$	$Q\ \overline{Q}$
$S\,R = 0\,0$	1 1	1 1	Случайное состояние Q: $\overline{Q}$ = 0 1 или 1 0

Большинство ячеек СОЗУ при включении питающего напряжения преимущественно переходят в одно из двух возможных состояний, поскольку каждая ячейка, представляющая собой RS-триггер, в силу специфики технологии ее изготовления имеет множество несимметричных элементов и параметров. Это в первую очередь касается длин соединительных проводников, их геометрических размеров, неоднородностей физического состава кремния, его химических свойств и, как результат, различия задержек сигналов.

Для увеличения диапазона изменения случайных значений задержек и, соответственно, стабильности и надежности ФНФ была предложена комбинированная реализация ФНФ [13]. Одним из вариантов подобных ФНФ является объединение ФНФ типа арбитр и ФНФ на базе RS-триггера. Основная идея предложенной схемы – это увеличение пути между выходом одного из двух элементов 2И-НЕ RS-триггера и входом другого элемента 2И-НЕ (см. рис. 3, *б*). Длина пути для двух обратных связей *Path*1 и *Path*2 увеличивается за счет последовательно включенных двухвходовых мультиплексоров ФНФ типа арбитр. Их количество всегда одинаково, а состав перераспределяется между двумя путями. При такой реализации задержки зависят не только от технологических вариаций во время производства логических элементов 2И-НЕ RS-триггера и их задержек, но и от значения запроса *C*. Значение формируемого запроса *C* определяет множество мультиплексоров, вариации задержек которых и влияют на значение ответа.

Необходимо отметить, что запрос *C* определяет не только значение задержек по двум путям *Path*1 и *Path*2, но и структуру комбинированного генератора [13]. При четном количестве единичных значений в векторе запроса $C = c_0\, c_1 \ldots c_{m-1}$ реализуется комбинация ФНФ типа арбитр и ФНФ на базе RS-триггера. Если число единичных значений в запросе нечетное, то в этом случае получим комбинацию ФНФ типа арбитр и кольцевого осциллятора (RO) (см. рис. 3, *c*) [13]. Отличием двух структур комбинированных ФНФ является режим их функционирования, который в первом случае повторяет функционирование ФНФ на базе RS-триггера, а во втором − функционирование кольцевого осциллятора.

В то же время общим для обоих вариантов комбинированных ФНФ является задание значений задержек по цепям обратной связи RS-триггера (см. рис. 3, *b*) и задержек работы двух кольцевых осцилляторов (см. рис. 3, *c*). Значения величин задержек в указанных структурах ФНФ определяются задержками мультиплексоров, возникающими в результате генерирования конкретного запроса *C*. Для каждого нового запроса *C* формируются новые пути *Path*1 и *Path*2, состоящие из других комбинаций тех же мультиплексоров, каждый из которых имеет свое уникальное время задержки.

Появление комбинированных ФНФ, основанных на структурах типа арбитр, повторяет идею задания определенных задержек по двум путям, сформированным из последовательно включенных мультиплексоров. Однако при рассмотрении этих же решений (см. рис. 3, *b* и *c*) с точки зрения ФНФ на базе RS-триггера и RO ФНФ можно сделать совершенно другие выводы. Так, структура, изображенная на рис. 3, *b*, представляет собой ФНФ на базе RS-триггера с управляемыми задержками по цепям обратной связи, а на рис. 3, *c* показаны два RO ФНФ с изменяемыми (управляемыми) частотами функционирования. В обоих случаях изначально управление заключается в выборе определенных задержек сигналов в соответствии с заданным запросом *C*.

Развивая классическую идею управления задержками сигналов через множество последовательно подключенных элементов, рассмотрим ряд решений управления задержками на уровне одного элемента. В качестве основы таких решений используем базовый элемент и его модификации, представленные на рис. 2.

ФНФ с управляемыми задержками на уровне элементов. Как отмечалось ранее, все известные решения при создании ФНФ основаны на том, что задержка по конкретному пути (элементу) имеет случайное, но вместе с тем неизменное и неуправляемое значение. Изменение задержки как результат влияния внешних факторов, таких как температура, давление, электромагнитное излучение, а также деградация физических и химических свойств частей элементов, относятся к негативным и нежелательным эффектам для ФНФ.

В качестве альтернативного подхода для построения ФНФ в настоящей статье обосновывается возможность построения нового класса ФНФ с управляемыми задержками. Первоначально рассмотрим простейший случай базового элемента (см. рис. 2, *d*), представляющего собой двухвходовый элемент XOR, один из входов которого управляющий. При подаче на второй вход этого элемента изменяющегося входного значения выходное значение XOR изменяется на противоположное. Двоичное значение $c_i = 0$ либо $c_i = 1$ на управляющем входе будет определять величину задержки изменения выходного значения *i*-го двухвходового элемента XOR. Ис-

пользуя n последовательно подключенных двухвходовых сумматоров по модулю два (XOR), построим схему управляемого кольцевого осциллятора (Controlled Ring Oscillator, CRO) в упрощенном виде (рис. 4).

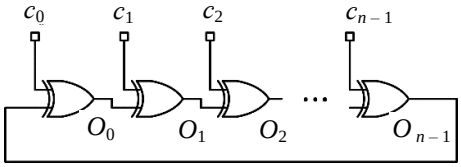


Рис. 4. Управляемый кольцевой осциллятор
Fig. 4. Controlled ring oscillator

Уникальность двухвходового элемента XOR позволяет напрямую управлять задержкой прохождения через него входного импульса путем задания произвольного входного значения 0 либо 1 по его управляющему входу. Предположив, что i-й элемент XOR имеет задержку d_{i0} при $c_i = 0$ и задержку d_{i1} при $c_i = 1$, суммарная задержка D прохождения сигнала через n последовательно подключенных элементов CRO определяется согласно соотношению

$$D = \sum_{i=0}^{n-1} (c_i d_{i1} + \overline{c}_i d_{i0}). \tag{3}$$

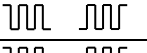
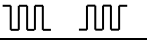
В конечном счете величина задержки D и определяет частоту $f = 1/(2D)$ генерируемых периодических сигналов управляемым кольцевым осциллятором при условии обеспечения отрицательной обратной связи. Необходимым и достаточным условием наличия такой связи является нечетное количество единичных значений c_i запроса $C = c_0\, c_1 \ldots c_{n-1}$.

Таким образом, ФНФ типа CRO, показанная на рис. 4, представляет собой 2^{n-1} классических кольцевых осциллятора RO, каждый из которых имеет свою уникальную частоту формирования выходных сигналов.

Допустив возможность задания любого из 2^m запросов C при функционировании CRO, можно заметить, что его поведение напоминает функционирование комбинированных ФНФ, рассмотренных ранее [13]. Детально исследуем поведение CRO для случая $n = 2$, когда структура, приведенная на рис. 4, представляется двумя последовательно включенными двухвходовыми элементами XOR (табл. 5).

Таблица 5
Описание функционирования CRO для случая $n = 2$

Table 5
Description of CRO *functioning for the case n = 2*

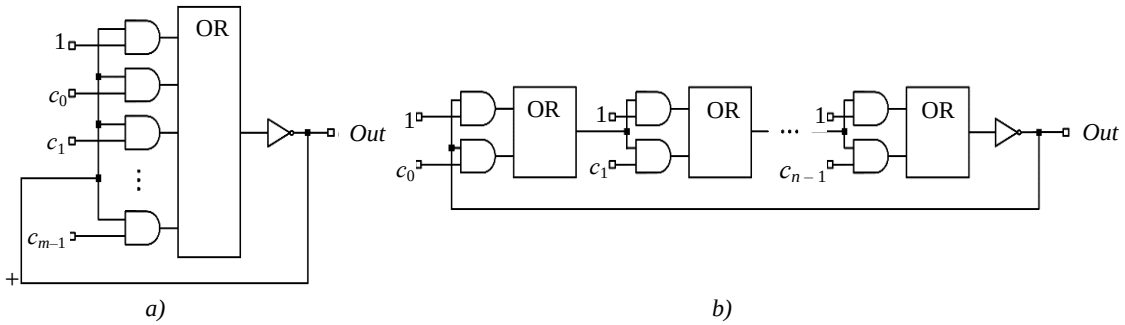
Текущие значения на входах c_0 и c_1 *Current values at inputs* c_0 *and* c_1	Следующие значения на входах c_0 и c_1 *Next values at inputs* c_0 *and* c_1	Текущее состояние $O_0\, O_1$ *Current values* $O_0\, O_1$	Следующее состояние $O_0\, O_1$ *Next values* $O_0\, O_1$
$c_0\, c_1 = 0\, 0$	0 0	$O\, O$	$O\, O$
$c_0\, c_1 = 0\, 1$ $c_0\, c_1 = 1\, 0$ $c_0\, c_1 = 1\, 1$	0 0	Любое, кроме $O\, O$	Случайное состояние $O\, O$
$c_0\, c_1 = X\, X$	0 1	Произвольное	
$c_0\, c_1 = X\, X$	1 0	Произвольное	
$c_0\, c_1 = 1\, 1$	1 1	$O\, \overline{O}$	$O\, \overline{O}$
$c_0\, c_1 = 0\, 0$ $c_0\, c_1 = 0\, 1$ $c_0\, c_1 = 1\, 0$	1 1	Любое, кроме $O\, \overline{O}$	Случайное состояние $O\, \overline{O}$

Из табл. 5 видно, что выходные значения O_0 и O_1 могут принимать и одинаковые ($O \in \{0, 1\}$), и противоположные (O и $\overline{O}$) значения. При подаче на входы c_0 и c_1 нулевых значений на выходах O_0 и O_1 формируются одинаковые значения O O, нулевые либо единичные. Аналогично функционирует CRO и в случае подачи на его входы единичных значений, при этом на выходах O_0 и O_1 формируются инверсные значения O и $\overline{O}$. В обоих случаях обеспечивается положительная обратная связь, определяющая устойчивое состояние CRO, который представляет собой комбинационную схему. Для входных значений 0 1 и 1 0 схема CRO будет генерировать высокочастотные колебания выходного сигнала, так как эти значения входных сигналов обеспечивают отрицательную обратную связь. При этом в первом случае частота сигнала определяется задержкой $d_{00} + d_{11}$, а во втором – суммарной задержкой $d_{01} + d_{10}$. В табл. 5 термин «произвольное» обозначает одно из следующих пяти возможных состояний: 0 0, 0 1, 1 0, 1 1 и состояние кольцевого генератора по выходам O_0 и O_1 CRO.

Как и для классического RS-триггера, для структуры CRO существуют ситуации неопределенного (случайного) поведения. Напомним, что формирование на входах RS-триггера значений 1 1 после входной комбинации 0 0 приводит к случайному состоянию RS-триггера (см. табл. 4). В случае CRO генерирование на входах c_0 и c_1 комбинации 0 0 после любой другой комбинации входных значений так же, как и формирование входных значений 1 1 после отличной от подаваемой ранее входной комбинации, приводит к установке CRO в произвольное (случайное) состояние (см. табл. 5).

Управляемые кольцевые осцилляторы, основанные на многовходовом переключении активного сигнала. Ранее уже отмечалось, что более сложные процессы и, соответственно, зависимости задержек через элемент возникают в случае переключения сигналов одновременно на нескольких входах (MIS) базового элемента и его модификаций, представленных на рис. 2. Под *активным сигналом* понимают значение входного сигнала элемента, изменение которого приводит к изменению выходного значения элемента. Модификации исходного базового элемента, показанные на рис. 2, *b* и *c*, позволяют управлять с помощью запроса *C* количеством переключающихся входных значений, которые изменяют выходное значение базового элемента. Отметим, что не только количество входов, но и их конкретный набор определяются запросом *C*, в результате чего и происходит задание конкретной задержки через базовый элемент.

Для построения CRO с многовходовым переключением входного сигнала (Multi Input Switching Controlled Ring Oscillator, MISCRO) исходными данными будут являться размерность *n* запроса *C*, управляющего величиной задержки, и ее математическое ожидание (среднее значение задержки). Эти два параметра для CRO, изображенного на рис. 4, находятся в полном противоречии. Рост количества 2^n возможных запросов CRO увеличивает значение *n* и, соответственно, среднее значение общей задержки *D* (3), что существенно уменьшает быстродействие CRO. Для возможности нахождения компромисса между размерностью *n* запроса *C* предлагаемых структур MISCRO и его задержками (быстродействием) приведем две его полярные реализации (рис. 5).



a) *b)*

Рис. 5. Управляемый кольцевой осциллятор с многовходовым переключением входного сигнала: *a)* первая реализация; *b)* вторая реализация

Fig. 5. Controlled ring oscillator with multi-input switching of input signal: a) first implementation; b) second implementation

Представленные на рис. 5 схемы основаны на модификации базового элемента (см. рис. 2, *b*). Первая реализация MISCRO (рис. 5, *a*) использует $m + 1$ входовый элемент ИЛИ (OR), на котором задействован режим переключения выходного значения на противоположное состояние при изменении активного сигнала более чем по одному входу. Конкретные входы и их количество определяются ненулевыми компонентами $c_i \in \{0, 1\}$ запроса $C = c_0\, c_1 \ldots c_{n-1}$. Для разных значений C задержка изменения выходного значения, как показывалось ранее, различна. За счет отрицательной обратной связи данная схема генерирует высокочастотный сигнал подобно классическому RO. Не вдаваясь в особенности реализации многовходовых логических элементов, можно констатировать, что первая схема MISCRO характеризуется максимальной частотой формируемого сигнала. Эта частота определяется суммарной задержкой только трех последовательно включенных логических элементов.

Вторая реализация MISCRO (рис. 5, *b*) имеет минимальное быстродействие в силу большого количества последовательно подключенных логических элементов, охваченных отрицательной обратной связью. По сравнению с первой реализацией (рис. 5, *a*) суммарная средняя задержка сигнала может быть больше для второй схемы практически в $(2n + 1)/(n + 2) \div (2n + 1)/3$ раз. Нижняя оценка $(2n + 1)/(n + 2)$ получена при допущении, что задержки сигналов элементов 2И, 2ИЛИ и НЕ одинаковы, а задержка n-входового элемента ИЛИ в n раз больше задержки аналогичного элемента с двумя входами. Верхняя оценка $(2n + 1)/3$ получена при тех же допущениях об одинаковых значениях задержки, но безотносительно количества входов в элементе, т. е. задержки и двухвходового, и n-входового элементов ИЛИ принимались равными. Из приведенных оценок видно, что быстродействие MISCRO, представленного на рис. 5, *a*, практически в два раза больше быстродействия MISCRO на рис. 5, *b*. Возможны компромиссные решения, когда количество входов элемента ИЛИ будет больше двух, но меньше n. Подобные решения возможны и для других модификаций базового элемента, равно как и развитие модификаций на рис. 5. Например, схема на рис. 5, *a* может быть упрощена за счет удаления элемента 2И, на второй вход которого подается значение 1, при одновременном запрете на применение запроса $C = c_0\, c_1 \ldots c_{n-1} = 0\, 0 \ldots 0$.

Еще бóльшие возможности имеет предлагаемый авторами подход при построении реализующих схемы типа арбитр ФНФ, когда сравниваются задержки по двум уникальным путям, построенным за счет реконфигурации этих путей в соответствии со значением запроса C.

Экспериментальные исследования. Практические исследования были направлены на подтверждение главной идеи авторов, заключающейся в том, что простейший логический элемент является источником уникальных значений задержки прохождения входного сигнала. Уникальность и непредсказуемость величины задержки, как уже отмечалось ранее, определяется в основном технологическими нормами и особенностями процесса изготовления логического элемента. Поэтому авторы сконцентрировали свое внимание на реальных цифровых устройствах.

В ходе первого эксперимента было получено подтверждение, что элемент 2XOR является источником восьми значений задержки сигнала (см. табл. 3), каждое из которых может использоваться при построении ФНФ. Исследования проводились для интегральной схемы К155ЛП5 (URL: http://chiplist.ru/chips/K155LP5/, https://pdf1.alldatasheet.com/datasheet-pdf/view/27431/TI/SN7486N.html), которая имеет в своем составе четыре (*A, B, C* и *D*) двухвходовых элемента XOR [17]. Суть эксперимента заключалась в определении основных статистических характеристик, таких как математические ожидания задержки μ, и среднеквадратического отклонения σ для каждой из восьми задержек $\Delta_1(LH)$, $\Delta_2(HL)$, …, Δ_8(LH). Эксперименты проводились для различных экземпляров микросхемы К155ЛП5 и четырех двухвходовых элементов XOR, входящих в их состав. На один из входов двухвходового элемента XOR подавались тестовые импульсы прямоугольной формы при помощи генератора сигналов специальной формы АКИП-3409/1 (URL: https://www.electronpribor.ru/catalog/51/akip-3409_1.htm). Измерение задержек $\Delta_1(LH)$, $\Delta_2(HL)$, …, $\Delta_8(LH)$ распространения фронта тестового сигнала от входа *In*1 либо *In*2 до выхода *Out* (см. табл. 3) осуществлялось при помощи двухканального цифрового осциллографа Rohde & Schwarz RTB2002 (URL: https://www.rohde-schwarz. com/ru/product/rtb2000-productstartpage_63493-266306.html). В качестве примера в табл. 6 даны результаты экспери-

ментов для двух (*A* и *B*) элементов XOR, принадлежащих двум различным микросхемам К155ЛП5#1 и К155ЛП5#2. Каждый эксперимент состоял в проведении 50 измерений задержки с последующим определением ее статистических характеристик. Численные характеристики, приведенные в табл. 6, согласуются со справочными данными микросхемы SN7486N, являющейся аналогом микросхемы К155ЛП5 (URL: https://pdf1.alldatasheet.com/ datasheet-pdf/view/27431/TI/SN7486N.html).

Таблица 6
Статистические характеристики задержек для элементов XOR
микросхем К155ЛП5#1 и К155ЛП5#2

Table 6
Statistical characteristics of time delays for elements XOR
of integrated circuits K155LP5#1 and K155LP5#2

Задержка, нс *Delay, ns*	$\Delta_1(LH)$	$\Delta_2(HL)$	$\Delta_3(LH)$	$\Delta_4(HL)$	$\Delta_5(HL)$	$\Delta_6(LH)$	$\Delta_7(HL)$	$\Delta_8(LH)$
К155ЛП5#1 (A)								
μ	30,48	20,85	31,18	19,31	16,73	38,84	15,35	36,68
σ	0,240	0,109	0,146	0,270	0,266	0,108	0,303	0,269
К155ЛП5#1 (B)								
μ	36,40	23,64	33,51	28,13	17,18	37,21	16,40	38,18
σ	0,168	0,144	0,204	0,123	0,070	0,094	0,129	0,132
К155ЛП5#2 (A)								
μ	35,31	17,75	29,70	16,53	20,35	39,63	16,48	37,88
σ	0,195	0,259	0,249	0,198	0,129	0,163	0,187	0,156
К155ЛП5#2 (B)								
μ	36,05	26,20	37,72	26,70	17,87	38,06	16,69	38,50
σ	0,093	0,155	0,107	0,156	0,108	0,080	0,058	0,104

Анализ результатов, представленных в табл. 6, позволяет сделать вывод о том, что все восемь значений задержек элемента 2XOR различимы и могут быть использованы для управления задержкой через элементы CRO путем выбора одного из них. Отличие значений одной и той же задержки для двух разных (*A* и *B*) элементов XOR одной микросхемы и их различие в зависимости от анализируемой микросхемы свидетельствуют о возможности создания CRO.

Второй эксперимент был нацелен на подтверждение следующей идеи, предлагаемой авторами для схемы CRO: для каждого запроса *C* базовый элемент генератора будет иметь свое уникальное значение задержки прохождения импульса, что, в свою очередь, приведет к уникальности частоты формируемых CRO импульсных последовательностей. Для проведения эксперимента была спроектирована схема CRO (см. рис. 4) для случая $n = 8$ и реализована на программируемой логической интегральной схеме (ПЛИС) Xilinx XC7Z010-1CLG00C (URL: https://www.xilinx.com/content/dam/xilinx/support/documentation/data_sheets/ds190-Zynq-7000-Overview.pdf), входящей в состав платы быстрого прототипирования Digilent ZYBO Z7 (URL: https://digilent.com/reference/programmable-logic/zybo-z7/start). Использованная ПЛИС изготовлена по КМОП-технологии HKMG (High-K Metal Gate) с применением технологической нормы 28 нм.

Спроектированная схема CRO имеет в своем составе дополнительный логический элемент 2И, расположенный в цепи обратной связи осциллятора, для обеспечения стартстопного режима работы. Структурные элементы схемы осциллятора размещены на фиксированных элементах 2XOR (технологическом примитиве XORCY), входящих в состав конфигурируемого блока ускоренного переноса CARRY4 (Carry Chain Block; URL: https://www.xilinx.com/support/ documentation/user_guides/ug474_7Series_CLB.pdf).

Помимо самого осциллятора, были спроектированы и реализованы дополнительные схемы его управления, такие как генератор разрешающего сигнала с программируемой длительностью, 32-разрядные счетчики для подсчета числа сгенерированных импульсов и общая схема управления и передачи данных для микропроцессорной системы ARM, входящей в состав ПЛИС. Кроме этого, было реализовано встраиваемое программное обеспечение, позволя-

ющее управлять экспериментом и передавать полученные данные на рабочую станцию для дальнейшего анализа. Для проектирования и реализации аппаратно-программной системы проведения эксперимента была использована САПР Xilinx Vivado/Vitis 2021 (URL: https://www.xilinx.com/products/design-tools/vivado.html, https://www.xilinx.com/products/design-tools/vitis/vitis-platform. html).

Эксперимент заключался в подаче всех возможных конфигурационных запросов C_i, удовлетворяющих условию

$$w(C_i) = 2j - 1, \ \forall\, i \in \{0, 1, \ldots, 2^n - 1\}, \ \ j \in \{1, 2, \ldots, n/2\},$$

где w – вес Хэмминга вектора запроса C_i. Для каждого из сформированных запросов был произведен подсчет числа сгенерированных импульсов в пределах фиксированного временного окна измерения. Из полученных данных вычислялись значения периодов генерируемых сигналов на выходе кольцевого осциллятора. Значения периодов сигналов, вырабатываемых схемой CRO, в зависимости от подаваемого запроса C_i представлены на рис. 6.

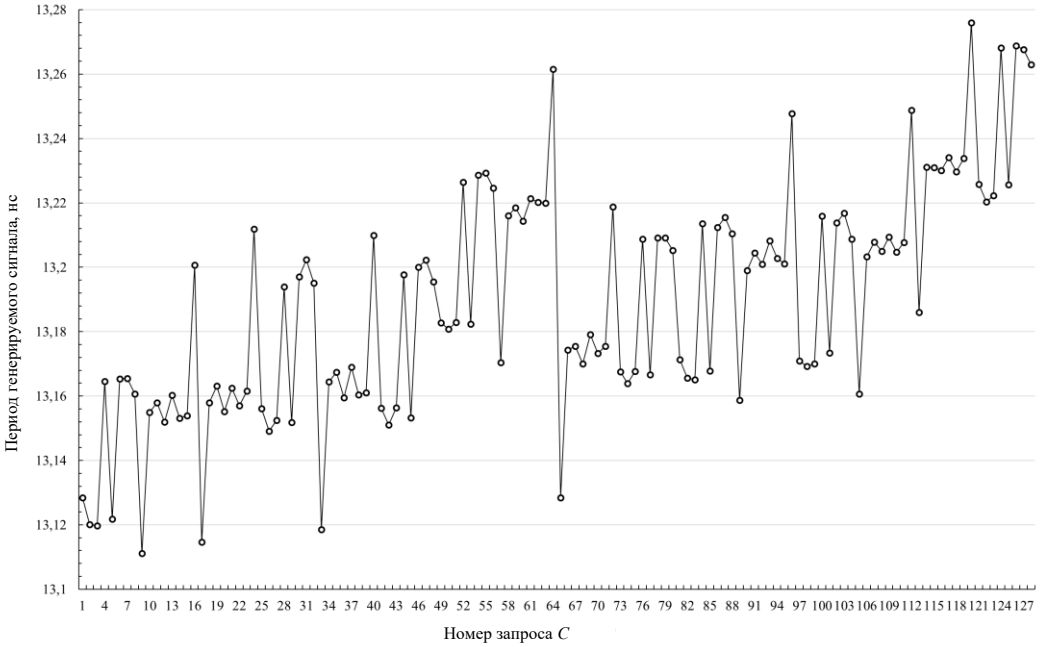


Рис. 6. Зависимость периода генерируемого сигнала от подаваемого запроса

Fig. 6. Dependence of the period of generated signal on submitted challenge

На рис. 7 показана зависимость математического ожидания периодов сигналов от значений запросов для четырех различных ПЛИС.

Согласно формуле (3) все множество уникальных запросов C формирует $2^{n-1} = 128$ уникальных значений $2D$ периодов сигналов, вырабатываемых конфигурируемым осциллятором в диапазоне [13,111; 13,276] нс. Таким образом, можно утверждать, что все структурные элементы 2XOR генератора обладают уникальными значениями задержек d_{i0} и d_{i1}.

Дополнительно были проведены 100 экспериментов, в ходе которых оценивались значения математического ожидания μ и среднеквадратического отклонения σ периодов сигналов, вырабатываемых схемой CRO, в зависимости от всех возможных значений запроса. Так, для многократно повторяемых 128 запросов в 100 экспериментах значения μ принадлежат диапазону [13,11183; 13,27604] нс, а значения σ – диапазону [0,00094; 0,00205] нс.

Для подтверждения факта уникальности эксперимент был повторен на трех других идентичных платах и ПЛИС Xilinx XC7Z010-1CLG00C. При этом использовался такой же BIT-образ конфигурации, что и в эксперименте с первой ПЛИС.

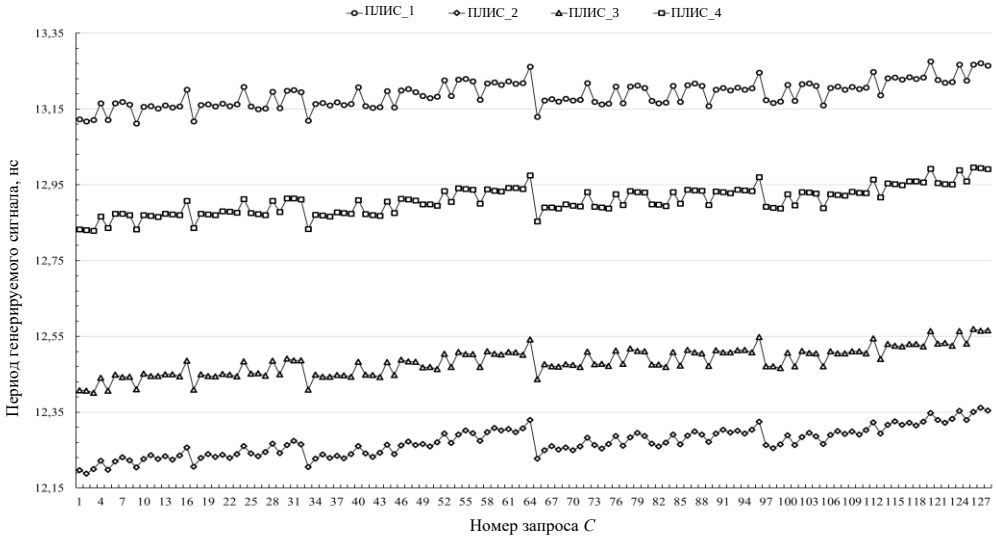


Рис. 7. Зависимость периодов генерируемых сигналов от подаваемых запросов для четырех ПЛИС

Fig. 7. Dependence of the periods of generated signal on the submitted challenges for four FPGAs

Таким образом, предположение о зависимости временных задержек от значений подаваемых запросов для схемы управляемого кольцевого генератора CRO, выдвинутое авторами, подтверждается. Можно утверждать, что временные задержки зависят от управляемых параметров (значения запроса *C*, протяженности *n* конфигурируемого пути) и неуправляемых уникальных характеристик его структурных элементов.

Заключение. Предложенный подход к построению физически неклонируемых функций, основанный на управлении задержкой сигналов через логические элементы, показал свою работоспособность и перспективность. Интересными представляются дальнейшее развитие идей построения управляемых кольцевых осцилляторов CRO и MISCRO, а также создание новых структур ФНФ типа арбитр. Дальнейшие исследования целесообразно расширить в части элементной базы как с технологической, так и функциональной точек зрения. В первую очередь необходимы уточнения особенностей управления задержкой для КМОП-технологии и практической реализации предложенных схем на различного рода программируемых структурах.

Вклад авторов. *В. Н. Ярмолик* предложил идею построения физически неклонируемых функций с управляемой задержкой сигналов. *А. А. Иванюк* принял участие в обобщении и анализе полученных результатов и проведении экспериментальных исследований. *Н. Н. Шинкевич* провела экспериментальные исследования.

Список использованных источников

1. Pappu, R. Physical One-Way Functions: PhD Thesis in Media Arts and Sciences / R. Pappu. – Cambridge : Massachusetts Institute of Technology, 2001. – 154 p.

2. Controlled physical random functions / B. Gassend [et al.] // Proc. of 18th Annual Computer Security Applications Conf. (ACSAC), Las Vegas, Nevada, USA, 9–13 Dec. 2002. – Las Vegas, 2002. – P. 149–160.

3. Rührmair, U. Strong PUFs: models, constructions, and security proofs / U. Rührmair, H. Busch, S. Katzenbeisser // Towards Hardware-Intrinsic Security / eds. A.-R. Sadeghi, D. Naccache. – Berlin, Heidelberg : Springer Berlin Heidelberg, 2010. – P. 79–96.

4. Agarwal, A. Statistical timing analysis for intra-die process variations with spatial correlations / A. Agarwal, D. Blaauw, V. Zolotov // Proc. of Intern. Conf. on Computer Aided Design (ICCAD03), San Jose, CA, USA, 9–13 Nov. 2003. – San Jose, 2003. – P. 900–907.

5. Böhm, C. Physical Unclonable Functions in Theory and Practice / C. Böhm, M. Hofer. – N. Y. : Springer Science + Business Media, 2013. – 270 p.

6. Theoretical analysis of delay-based PUFs and design strategies for improvement / Y. Wang [et al.] // Proc. of the IEEE Intern. Symp. on Circuits and Systems (ISCAS), Sapporo, Japan, 26–29 May 2019. – Sapporo, 2019. – P. 1–5.

7. Gummalla, S. An Analytical Approach to Efficient Circuit Variability Analysis in Scaled CMOS Design: Master Degree Thesis / S. Gummalla. – Arizona : Arizona State University, 2011. – 62 p.

8. A technique to build a secret key in integrated circuits for identification and authentication applications / J. W. Lee [et al.] // Proc. of the Intern. Symp. VLSI Circuits (VLSI'04), Honolulu, Hawaii, USA, 7–19 June 2004. – Honolulu, 2004. – P. 176–179.

9. Ozturk, E. Physical unclonable function with tristate buffers / E. Ozturk, G. Hammouri, B. Sunar // Proc. of the IEEE Intern. Symp. on Circuits and Systems (ISCAS 2008), Seattle, Washington, USA, 18–21 May 2008. – Seattle, 2008. – P. 3194–3197.

10. The bistable ring PUF: A new architecture for strong physical unclonable functions / Q. Chen [et al.] // Proc. of the IEEE Intern. Symp. on Hardware Oriented Security and Trust (HOST'11), San Diego, California, USA, 5–6 June 2011. – San Diego, 2011. – P. 134–141.

11. Holcomb, D. E. Power-up SRAM state as an identifying fingerprint and source of true random numbers / D. E. Holcomb, W. Burleson, K. Fu // IEEE Transactions on Computer. – 2008. – Vol. 58, no. 9. – P. 1198–1210.

12. DRAM-based intrinsic physically unclonable functions for system-level security and authentication / F. Tehranipoor [et al.] // IEEE Transactions on Very Large Scale Integration (VLSI) Systems. – 2016. – No. 99. – P. 1–13.

13. Ярмолик, В. Н. Физически неклонируемые функции / В. Н. Ярмолик, Ю. Г. Вашинко // Информатика. – 2011. – № 2(30). – С. 92–103.

14. Иванюк, А. А. Физическая криптография и защита цифровых устройств / А. А. Иванюк, С. С. Заливако // Доклады БГУИР. – 2019. – № 2(120). – С. 50–58.

15. Верниковский, Е. А. Схемотехника: учебно-методический комплекс / Е. А. Верниковский. – Минск : БГУ, 2012. – 200 с.

16. Jouppi, N. Timing analysis and performance improvement of MOS VLSI designs / N. Jouppi // IEEE Transactions on Computer-Aided Design. – 1987. – Vol. 6, no. 4. – P. 650–665.

17. Богданович, М. И. Цифровые интегральные микросхемы / М. И. Богданович, И. Н. Грель, В. А. Прохоренко. – Минск : Беларусь, 1991. – 493 с.

18. Ram, O. V. S. S. Modeling multiple-input switching in timing analysis using machine learning / O. V. S. S. Ram, S. Saurabh // IEEE Trans. on Computer. – 2021. – Vol. 40, no. 4. – P. 723–734.

19. Experiments on autonomous Boolean networks / D. P. Rosin [et al.] // Chaos: An Interdisciplinary J. of Nonlinear Science. – 2013. – Vol. 23, no. 2. – P. 1–9.

20. Park, M. True random number generation using CMOS Boolean chaotic oscillator / M. Park, J. C. Rodgers, D. P. Lathrop // Microelectronics J. – 2015. – Vol. 46, no. 12. – P. 1364–1370.

References

1. Pappu R. *Physical One-Way Functions: PhD Thesis in Media Arts and Sciences*. Cambridge, Massachusetts Institute of Technology, 2001, 154 p.

2. Gassend B., Clarke D., Dijk M. S., Devadas S. Controlled physical random functions. *Proceedings of the 18th Annual Computer Security Applications Conference (ACSAC), Las Vegas, Nevada, USA, 9–13 December 2002*. Las Vegas, 2002, pp. 149–160.

3. Rührmair U., Busch H., Katzenbeisser S. Strong PUFs: Models, Constructions, and Security Proofs. *Towards Hardware-Intrinsic Security*. In Sadeghi A.-R., Naccache D. (eds.). Berlin, Heidelberg, Springer Berlin Heidelberg, 2010, pp. 79–96.

4. Agarwal A., Blaauw D., Zolotov V. Statistical timing analysis for intra-die process variations with spatial correlations. *Proceedings of the International Conference on Computer Aided Design (ICCAD03), San Jose, CA, USA, 9–13 November 2003.* San Jose, 2003, pp. 900–907.

5. Böhm C., Hofer M. *Physical Unclonable Functions in Theory and Practice*. New York, Springer Science + Business Media, 2013, 270 p.

6. Wang Y., Wang C., Gu C., Cui Y. Theoretical analysis of delay-based PUFs and design strategies for improvement. *Proceedings of the IEEE International Symposium on Circuits and Systems (ISCAS), Sapporo, Japan, 26–29 May 2019*. Sapporo, 2019, pp. 1–5.

7. Gummalla S. *An Analytical Approach to Efficient Circuit Variability Analysis in Scaled CMOS Design: Master Degree Thesis*. Arizona, Arizona State University, 2011, 62 p.

8. Lee J. W., Lim D., Gassend B., Suh G., Dijk M., Devadas S. A technique to build a secret key in integrated circuits for identification and authentication applications. *Proceedings of the International Symposium VLSI Circuits (VLSI'04), Honolulu, Hawaii, USA, 7–19 June 2004.* Honolulu, 2004, pp. 176–179.

9. Ozturk E., Hammouri, G., Sunar B. Physical unclonable function with tristate buffers. *Proceedings of the IEEE International Symposium on Circuits and Systems (ISCAS 2008), Seattle, Washington, USA, 18–21 May 2008*. Seattle, 2008, pp. 3194–3197.

10. Chen Q., Csaba G., Lugli P., Schlichtmann U., Rührmair U. The bistable ring PUF: A new architecture for strong physical unclonable functions. *Proceedings of the IEEE International Symposium on Hardware Oriented Security and Trust (HOST'11), San Diego, California, USA, 5–6 June 2011*. San Diego, 2011, pp. 134–141.

11. Holcomb D. E., Burleson W., Fu K. Power-up SRAM state as an identifying fingerprint and source of true random numbers. *IEEE Transactions on Computers*, 2008, vol. 58, no. 9, pp. 1198–1210.

12. Tehranipoor F., Karimian N., Xiao K., Chandy J. DRAM-based intrinsic physically unclonable functions for system-level security and authentication. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 2016, no. 99, pp. 1–13.

13. Yarmolik V. N., Vashinko Y. G. *Physical unclonable functions.* Informatika [*Informatics*], 2011, no. 2(30), pp. 92–103 (In Russ.).

14. Ivaniuk A. A., Zalivaka S. S. *Physical cryptography and security of digital devices*. Doklady Belorusskogo gosudarstvennogo universiteta informatiki i radioèlektroniki [*Reports of the Belarusian State University of Informatics and Radioelectronics*], 2019, no. 2(120), pp. 50–58 (In Russ.).

15. Vernikovskii E. A. Schemotehnika: uchebno-metodicheskii complex. *Circuitry: Educational-methodical Complex*. Minsk, Belorusskij gosudarstvennyj universitet, 2012, 200 p. (In Russ.).

16. Jouppi N. Timing analysis and performance improvement of MOS VLSI designs. *IEEE Transactions on Computer-Aided Design*, 1987, vol. 6, no. 4, pp. 650–665.

17. Bogdanovich M. I., Grel' I. N., Prohorenko V. A. Tsifrovue integral'nue mikroshemy. *Digital Integrated Circuits*, Minsk, Belarus, 1991, 493 p. (In Russ.).

18. Ram O. V. S. S., Saurabh S. Modeling multiple-input switching in timing analysis using machine learning. *IEEE Transactions on Computer*, 2021, vol. 40, no. 4, pp. 723–734.

19. Rosin D. P., Rontani D., Gauthier D. J. Experiments on autonomous Boolean networks. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 2013, vol. 23, no. 2, pp. 1–9.

20. Park M., Rodgers J. C., Lathrop D. P. True random number generation using CMOS Boolean chaotic oscillator. *Microelectronics Journal*, 2015, vol. 46, no. 12, pp. 1364–1370.

Информация об авторах

Ярмолик Вячеслав Николаевич, доктор технических наук, профессор, Белорусский государственный университет информатики и радиоэлектроники.
E-mail: yarmolik10ru@yahoo.com

Иванюк Александр Александрович, доктор технических наук, доцент, профессор кафедры информатика и заведующий совместной учебной лабораторией SK Hynix Memory Solutions Eastern Europe, Белорусский государственный университет информатики и радиоэлектроники.
E-mail: ivaniuk@bsuir.by

Шинкевич Наталья Николаевна, аспирант, Белорусский государственный университет информатики и радиоэлектроники.
E-mail: nn5h@yahoo.com

Information about the authors

Vyacheslav N. Yarmolik, D. Sc. (Eng.), Professor, Belarusian State University of Informatics and Radioelectronics.
E-mail: yarmolik10ru@yahoo.com

Alexander A. Ivaniuk, D. Sc. (Eng.), Assoc. Prof., Professor of Comp. Sci. Department, Head of the Joint Educational Laboratory "SK Hynix Memory Solutions Eastern Europe", Belarusian State University of Informatics and Radioelectronics.
E-mail: ivaniuk@bsuir.by

Natallia N. Shynkevich, Graduate Student, Belarusian State University of Informatics and Radioelectronics.
E-mail: nn5h@yahoo.com

ОБРАБОТКА СИГНАЛОВ, ИЗОБРАЖЕНИЙ, РЕЧИ, ТЕКСТА И РАСПОЗНАВАНИЕ ОБРАЗОВ

SIGNAL, IMAGE, SPEECH, TEXT PROCESSING AND PATTERN RECOGNITION

УДК 004.942: 621.396.969.3
https://doi.org/10.37661/1816-0301-2022-19-1-50-58

Оригинальная статья
Original Paper

Фильтрация при наличии перерывов информации на основе расширенного метода наименьших квадратов

В. М. Артемьев, А. О. Наумов✉

Институт прикладной физики Национальной академии наук Беларуси,
ул. Академическая, 16, Минск, 220072, Беларусь
✉*E-mail: naumov@iaph.bas-net.by*

Аннотация

Цели. В радиолокационных системах сопровождения движущихся объектов часто возникают перерывы в измерении координат. Наиболее полно в непрерывном времени эта проблема решена в теории систем со случайной структурой в рамках статистической байесовской теории фильтрации при наличии полной априорной статистической информации. Такой подход приводит к сложным алгоритмам, трудно реализуемым на практике. Целью исследования являлась разработка алгоритма фильтрации в условиях перерывов информации на основе применения расширенного метода наименьших квадратов.

Методы. Используются методы теории оценивания, в частности расширенный метод наименьших квадратов, позволяющий находить сравнительно простые алгоритмы при минимальных объемах априорных знаний о характеристиках воздействий.

Результаты. Разработан алгоритм фильтрации радиолокационных сигналов, в основе которого лежат измерения моментов перерывов и экстраполяция измеряемых координат на интервалах отсутствия информации. Полученный алгоритм является нелинейным, и за счет этого в фильтре могут возникать срывы сопровождения. Результаты работы алгоритма продемонстрированы на модельном примере, выполнена оценка точности фильтрации и условий срыва слежения.

Заключение. Разработанный алгоритм фильтрации позволяет определять моменты наступления перерывов и осуществлять экстраполяцию оценок полезной информации. Сравнительная простота алгоритма делает его пригодным для практического использования.

Ключевые слова: радиолокационная станция, цифровая фильтрация, обнаружение движущихся объектов, метод наименьших квадратов, селекция траекторий, срыв слежения

Для цитирования. Артемьев, В. М. Фильтрация при наличии перерывов информации на основе расширенного метода наименьших квадратов / В. М. Артемьев, А. О. Наумов // Информатика. – 2022. – Т. 19, № 1. – С. 50–58. https://doi.org/10.37661/1816-0301-2022-19-1-50-58

Конфликт интересов. Авторы заявляют об отсутствии конфликта интересов.

Поступила в редакцию | Received 21.12.2021
Подписана в печать | Accepted 14.02.2022
Опубликована | Published 29.03.2022

Filtering in the presence of information losses based on the extended least squares method

Valentin M. Artemiev, Alexander O. Naumov✉

Institute of Applied Physics of the National Academy of Sciences of Belarus,
st. Akademicheskaya, 16, Minsk, 220072, Belarus
✉*E-mail: naumov@iaph.bas-net.by*

Abstract

Objectives. In radar systems for moving objects tracking, there are often gaps in the measurement of coordinates. The problem is mostly fully solved in continuous time in the theory of systems with a random structure within the framework of statistical Bayesian theory of filtration in the presence of complete a priori statistical information. This approach leads to complex algorithms that are difficult to implement in practice. The purpose of investigation was to develop a filtering algorithm in conditions of information interruptions based on the use of extended least squares method.

Methods. Methods of estimation theory are used, in particular, the extended least squares method, which makes it possible to find relatively simple algorithms with a minimum amount of a priori knowledge about the characteristics of the impacts.

Results. The algorithm for filtering radar signals has been developed, based on measurements of the moments of breaks and extrapolation of the measured coordinates at intervals of information lack. The resulting algorithm is nonlinear and therefore tracking disruptions may occur in the filter. The results of the algorithm are demonstrated using a model example. The estimation of the filtering accuracy and tracking failure conditions is carried out.

Conclusion. A filtering algorithm has been developed that allows determining the moments of the onset of breaks and extrapolating the estimates of useful information. The comparative simplicity of the algorithm makes it suitable for practical use.

Keywords: radar station, digital filtering, detection of moving objects, the method of least squares, trajectory selection, cycle slip

For citation. Artemiev V. M., Naumov A. O. *Filtering in the presence of information losses based on the extended least squares method*. Informatika [*Informatics*], 2022, vol. 19, no. 1, pp. 50–58 (In Russ.).
https://doi.org/10.37661/1816-0301-2022-19-1-50-58

Conflict of interest. The authors declare of no conflict of interest.

Введение. Построение траекторий движения объектов в радиолокации основано на текущих измерениях их координат и дальнейшей обработке результатов (фильтрации). Для селекции траекторий нескольких объектов производится *стробирование результатов измерений*, т. е. использование данных из ограниченной области измерений [1]. Как правило, из-за наличия помех, замираний сигналов и выхода измерений за пределы строба появляются перерывы в поступлении полезной информации. В модельном представлении перерывы могут рассматриваться как мультипликативная помеха в измерениях, принимающая значение единицы при наличии информации и нуля при ее отсутствии. Решению задачи фильтрации сигналов при таких помехах уделялось внимание в ряде работ. Наиболее полно в непрерывном времени задача решена в теории систем со случайной структурой [2]. Все результаты получены в рамках статистической байесовской теории фильтрации при наличии полной априорной статистической информации. Такой подход приводит к сложным структурам и алгоритмам, трудно реализуемым на практике. Избежать этих ограничений можно путем эмпирического подхода, использующего минимальный объем априорной информации и позволяющего получать сравнительно простые алгоритмы фильтрации. В настоящей работе с этой целью используется расширенный (модифицированный) метод наименьших квадратов [3].

Алгоритм фильтрации с экстраполяцией. Будем считать, что полезный сигнал, подлежащий фильтрации, является случайной последовательностью x_k, задаваемой в дискретные моменты времени $k = 0, 1, 2, \ldots$. Относительно свойств сигнала используем эмпирическое пред-

положение о «гладкости» его огибающей в том смысле, что на протяжении нескольких периодов огибающая может с достаточной точностью быть аппроксимирована полиномом. Измеряемый процесс z_k состоит из смеси полезного сигнала x_k, мультипликативной помехи u_k и аддитивной помехи v_k:

$$z_k = x_k u_k + v_k , \tag{1}$$

где мультипликативная помеха u_k является случайной бинарной последовательностью, т. е. при отсутствии перерывов информации $u_k = 1$, в противном случае $u_k = 0$. Аддитивная помеха v_k считается дискретным, более широкополосным, чем полезный сигнал, шумом с нулевым математическим ожиданием. Процессы x_k, u_k и v_k полагаются статистически независимыми. Алгоритм фильтра должен позволять решать следующие задачи: давать текущую оценку $\hat{x}_k$ полезного сигнала; по результатам измерений давать оценку текущего состояния прерываний сигнала $\hat{u}_k$, равную единице или нулю; определять экстраполированные значения оценок $\tilde{x}_k$ на интервалах прерывания.

В расширенном методе наименьших квадратов [4] используется квадратичная функция потерь $J(\hat{x}_k)$, состоящая из двух слагаемых. Первое слагаемое определяет потери за счет невязки между оценкой $\hat{x}_k$ и измерением z_k в виде квадрата разности $(z_k - \hat{x}_k)^2$, второе – учитывает невязку между оценкой и ее экстраполированным значением $(\tilde{x}_k - \hat{x}_k)^2$. Итоговое выражение функции потерь имеет вид

$$J(\hat{x}_k) = \alpha_k (z_k - \hat{x}_k)^2 + (1-\alpha_k)(\tilde{x}_k - \hat{x}_k)^2 . \tag{2}$$

Выбор весового коэффициента α_k производится исходя из состояния процесса прерывания. При наличии прерываний использование результатов измерений не имеет смысла и следует полагать $\alpha_k = 0$. При отсутствии прерываний коэффициент выбирается в пределах $0 \le \alpha_k \le 1$ и его значение зависит от отношения сигнала к шумам. При большом отношении сигнала к шумам предпочтение следует отдавать результатам измерений $(\alpha_k > 0{,}5)$, а при малом – результатам экстраполяции $(\alpha_k < 0{,}5)$. Поскольку предполагается отсутствие априорных статистических данных о воздействии, целесообразно выбирать значение $\alpha_k = 0{,}5$.

В общем случае, исходя из необходимых условий оптимальности $\partial J(\hat{x}_k)/\partial \hat{x}_k = 0$, имеет место уравнение

$$\alpha_k (z_k - \hat{x}_k) + (1-\alpha_k)(\tilde{x}_k - \hat{x}_k) = 0 ,$$

решение которого приводит к алгоритму фильтрации

$$\hat{x}_k = \alpha_k z_k + (1-\alpha_k)\tilde{x}_k = \tilde{x}_k + \alpha_k (z_k - \tilde{x}_k) . \tag{3}$$

Подставляя формулу (3) в выражение (2), находим величину минимальных потерь

$$\hat{J}(\hat{x}_k) = \alpha_k (1-\alpha_k)(z_k - \tilde{x}_k)^2 \tag{4}$$

с учетом того, что сомножитель $\alpha_k(1-\alpha_k)$ при $\alpha_k = 0{,}5$ максимален и равен 0,25.

Оценка мультипликативной помехи $\hat{u}_k$ должна иметь значение $\hat{u}_k = 1$ в предположении отсутствия перерывов и $\hat{u}_k = 0$ при их наличии. Данные оценки используются для нахождения коэффициентов α_k по формуле

$$\alpha_k = \eta \hat{u}_k , \tag{5}$$

где постоянный множитель η может быть, в частности, равен 0,5.

Значения $\hat{u}_k$ можно находить исходя из величины разности $(z_k - \tilde{x}_k)^2$. Если $u_k = 1$, то $(z_k - \tilde{x}_k)^2 = (x_k - \tilde{x}_k + v_k)^2 = f_{k1}$. Если $u_k = 0$, то $(z_k - \tilde{x}_k)^2 = (\tilde{x}_k - v_k)^2 = f_{k0}$. Эмпирический алгоритм нахождения оценок $\hat{u}_k$ может иметь вид

$$\hat{u}_k = \begin{cases} 0, \text{ если } (z_k - \tilde{x}_k)^2 \geq h; \\ 1, \text{ если } (z_k - \tilde{x}_k)^2 < h, \end{cases} \tag{6}$$

где h – порог обнаружения прерываний.

Сложность нахождения уровня порога h состоит в том, что статистика процесса $(z_k - \tilde{x}_k)^2$ неизвестна и приходится этот уровень на первом этапе находить приближенно, а затем уточнять при моделировании фильтра. Выбор порога можно осуществлять из сравнения средних значений $\overline{f}_{k1}$ и $\overline{f}_{k0}$ с учетом того, что процессы $(x_k - \tilde{x}_k)$ и v_k статистически независимы. При $u_k = 1$ среднее значение $\overline{f}_{k1} = \overline{(x_k - \tilde{x}_k)^2} + \sigma_v^2$, а при $u_k = 0$ $\overline{f}_{k0} = \overline{\tilde{x}_k^2} + \sigma_v^2$. В дальнейшем предполагается, что $\overline{(x_k - \tilde{x}_k)^2} \approx 0$, т. е. можно считать значение $\overline{f}_{k1}$ приблизительно равным σ_v^2. Порог h целесообразно выбирать в диапазоне между σ_v^2 и $(\overline{\tilde{x}_k^2} + \sigma_v^2)$. Используя введенное выше предположение о том, что $\overline{x_k^2} \approx \overline{\tilde{x}_k^2}$, среднее значение уровня порога будет определяться выражением

$$h = 0{,}5\overline{x_k^2} + \sigma_v^2.$$

Если x_k – стационарный случайный процесс с математическим ожиданием m_x и дисперсией σ_x^2, то уровень порога рассчитывается по формуле

$$h = 0{,}5(m_x^2 + \sigma_x^2) + \sigma_v^2. \tag{7}$$

Если x_k – гармонический процесс со средним значением m_x и амплитудой ϑ_x, то верно равенство

$$h = 0{,}5(m_x^2 + 0{,}5\vartheta_x^2) + \sigma_v^2. \tag{8}$$

Экстраполяция $\tilde{x}_k$ является функцией n предыдущих оценок $\hat{x}_{k-i}$, $i = \overline{1,n}$. Вид этой функции задается выбором закона экстраполяции и может быть линейным, полиномиальным, экспоненциальным и т. п. Широко используемым является первый вариант, который может быть представлен в форме полинома n-го порядка

$$\tilde{x}_k = \sum_{i=1}^{n+1} a_i \hat{x}_{k-i}, \tag{9}$$

где коэффициенты a_i выбираются одним из способов полиномиальной экстраполяции. Простейшим вариантом линейной полиномиальной экстраполяции, не требующим знания модели траектории движения объекта, может быть интерполяция конечными разностями [5], являющаяся дискретным аналогом разложения непрерывной функции в ряд Тейлора. В этом случае коэффициенты a_i задаются выражением

$$a_i = (-1)^{i+1}(n+1)!/i!(n+1-i)! \tag{10}$$

Исходя из вышеизложенного, построим блок-схему фильтра с экстраполяцией (рис. 1). Блок-схема фильтра содержит линейную часть, образуемую алгоритмами оценивания (3) и экстраполяции (9), а также нелинейную часть, содержащую алгоритмы оценки прерывания (6) и нахождения коэффициентов фильтра (5). Нелинейный характер фильтра существенно усложняет его анализ и приводит к появлению таких явлений, как срыв фильтрации.

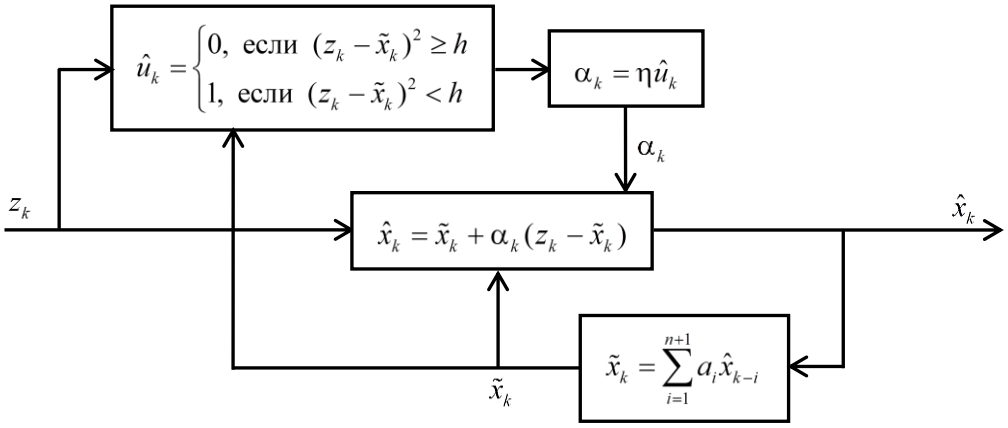


Рис. 1. Блок-схема фильтра с экстраполяцией при перерывах поступления информации

Fig. 1. Block diagram of the filter with extrapolation in conditions of information losses

Срыв фильтрации. Выше отмечалось, что в радиолокации измерение координат и построение на основе полученных данных траекторий объектов (фильтрация) осуществляются путем стробирования, которое производится при каждом измерении относительно экстраполированного значения $\tilde{x}_k$. Если ширина строба равна 2Δ, то при превышении модулем разности $|z_k - \tilde{x}_k|$ величины Δ измерения прерываются и происходит срыв фильтрации или сопровождения. Он может наступить по причинам резкого изменения координат (динамической ошибки фильтрации), наличия аддитивных помех (флуктуационных ошибок) или прерывания измерений за счет замираний отраженного сигнала. Условие срыва можно определить как первое наступление события, соответствующее неравенству

$$(z_k - \tilde{x}_k)^2 \geq \Delta^2. \qquad (11)$$

Индикатор срыва на основе неравенства (11) используется в дополнение к блок-схеме на рис. 1.

Фильтр с экстраполяцией является нелинейным, поэтому срыв фильтрации в нем возможен из-за его динамических свойств даже в том случае, когда $(z_k - \tilde{x}_k)^2 < \Delta^2$. Поскольку при динамическом срыве фильтрации разность $(z_k - \tilde{x}_k)^2$ непрерывно возрастает, в итоге условие срыва (11) будет выполнено. Аналитическое исследование ошибок фильтрации и условий срыва в настоящее время выполнить невозможно. Даже задача первого достижения заданного уровня в линейной системе достаточно сложна [6], поэтому оценка ошибок и условия срыва в фильтре с экстраполяцией могут быть получены лишь путем моделирования.

Моделирование. При нахождении алгоритма фильтрации методом наименьших квадратов априорные статистические данные о случайных процессах x_k, u_k и v_k не использовались. Были наложены лишь эмпирические ограничения в виде предположения о гладкости огибающей дискретного случайного процесса x_k. Поэтому в отличие от статистического подхода, использующего априорные статистические данные, приходится вводить типовые формы процесса x_k и помех, отражающие некоторые реальные свойства воздействий в конкретных ситуациях. Для типовых полезных воздействий могут применяться как случайные, так и регулярные функции полиномиального вида, гармонические и др., а для помех – случайные процессы типа белого шума, точечные и т. п. Качество фильтрации при выбранных типовых воздействиях считается приемлемым и для реальных ситуаций.

При исследовании в качестве типовой огибающей процесса x_k на плоскости использовалась гармоническая функция $x(t) = \vartheta_x \sin\left(\frac{2\pi}{T} t\right) + m_x$, где ϑ_x – амплитуда, T – период следования и m_x – постоянная составляющая. Эти параметры задавались исходя из эмпирических соображений

о допустимых значениях скоростей $\dot{x}$ и ускорений $\ddot{x}$. Так, максимальное значение модуля скорости изменения функции $x(t)$ рассчитывается по формуле $\left|\frac{dx(t)}{dt}\right|_{\max} = \dot{x} = \vartheta_x \frac{2\pi}{T}$, а ускорения – по формуле $\left|\frac{d^2x(t)}{dt}\right|_{\max} = \ddot{x} = \vartheta_x \left(\frac{2\pi}{T}\right)^2$. Задаваясь этими величинами, можно найти параметры функции $x(t)$ из соотношений

$$\vartheta_x = \dot{x}^2/\ddot{x}, \quad T = 2\pi\dot{x}/\ddot{x}. \tag{12}$$

Дискретные значения функции $x(t)$ получаются путем замены непрерывного времени t на дискретное значение $k = 0, 1, 2, \ldots$.

Мультипликативная помеха u_k задается посредством дискретной случайной бинарной последовательности со статистически независимыми значениями $u_k = 1$ или $u_k = 0$ с вероятностями появления p_u и $(1–p_u)$ соответственно. Приведем средние длительности такой последовательности:

– при $u_k = 1$

$$\tau_u = 1/(1 - p_u), \tag{13}$$

– при $u_k = 0$

$$\overline{\tau}_u = 1/p_u. \tag{14}$$

Аддитивная помеха v_k выбирается в форме стационарного дискретного белого шума с нормальным законом распределения вероятностей, нулевым математическим ожиданием и дисперсией σ_v^2.

При использовании в качестве исходных данных максимальных значений скоростей $\dot{x}_{\max}$ и ускорений $\ddot{x}_{\max}$, что обычно имеет место при фильтрации траекторий движущихся объектов, можно получить параметры гармонической функции огибающей входного процесса $x(t)$ по формулам (12). В рассматриваемом примере в качестве функции экстраполяции (9) применяется ряд Тейлора с коэффициентами (10). Так, при экстраполяции нулевого порядка $(n = 0)$

$$\tilde{x}_k = \hat{x}_{k-1}, \tag{15}$$

первого порядка $(n = 1)$

$$\tilde{x}_k = 2\hat{x}_{k-1} - \hat{x}_{k-2}, \tag{16}$$

второго порядка $(n = 2)$

$$\tilde{x}_k = 3\hat{x}_{k-1} - 3\hat{x}_{k-2} + \hat{x}_{k-3} \tag{17}$$

и т. д.

Моделирование проводилось при следующих значениях параметров: $\dot{x}_{\max} = 0{,}5$, $\ddot{x}_{\max} = 0{,}03$, $\vartheta_x = 2$, $T = 25$, $m_x = 5$, $p_u = 0{,}8$, $\tau_u = 5$, $m_x = 5$, $\overline{\tau}_u = 1{,}25$, $\sigma_v^2 = 0{,}1$. Интервал моделирования k выбирался от 0 до 100, а в качестве функций экстраполяции $\tilde{x}_k$ выступали экстраполяторы с уравнениями (15)–(17).

Результаты моделирования при уровне порога $h = 13{,}6$ показаны на рис. 2, где зеленым цветом отмечен график изменения входного процесса x_k, синим – график изменения выходного

процесса $\hat{x}_k$ при экстраполяции нулевого порядка, а красным – график изменения выходного процесса при экстраполяции первого порядка, где при k = 83 наступает срыв фильтрации. При экстраполяции второго порядка срыв наступает сразу и на рисунке не показан. На интервале времени от k = 0 до примерно k = 30 имеет место переходный процесс.

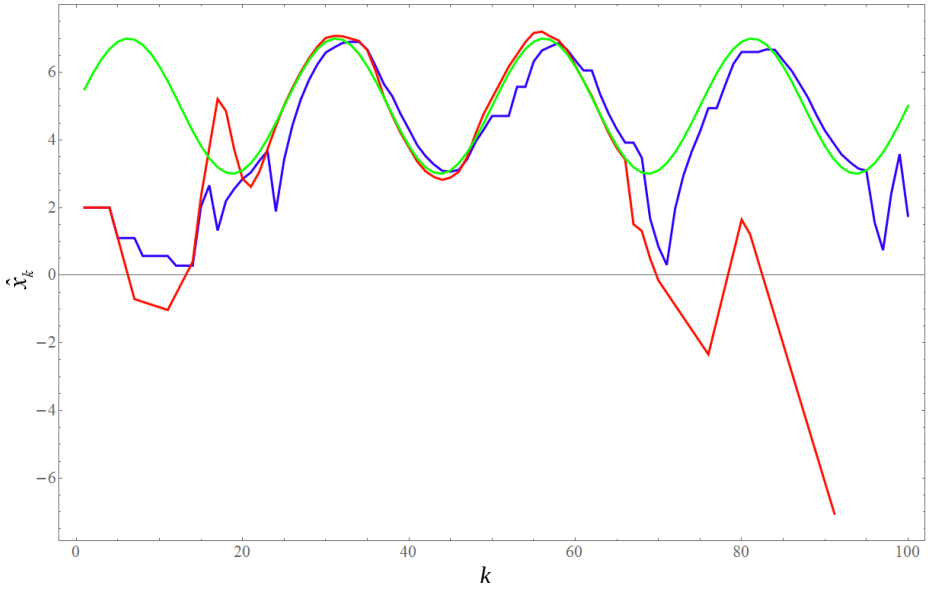


Рис. 2. Изменения входного x_k и выходного $\hat{x}_k$ процессов при экстраполяции нулевого и первого порядков

Fig. 2. Changes of the input x_k and output $\hat{x}_k$ processes at extrapolation of the zero and first orders

Результаты моделирования показывают работоспособность алгоритма фильтрации и то, что с ростом порядка экстраполяции вероятность срыва растет. Методом статистических испытаний по 10 000 реализациям получены средние значения ошибок $m_e = \langle x_k - \hat{x}_k \rangle$ (рис. 3) и их среднеквадратических отклонений σ_e (рис. 4) от величины h в установившемся режиме фильтрации, дополнительно усредненные по времени на участке в один период T.

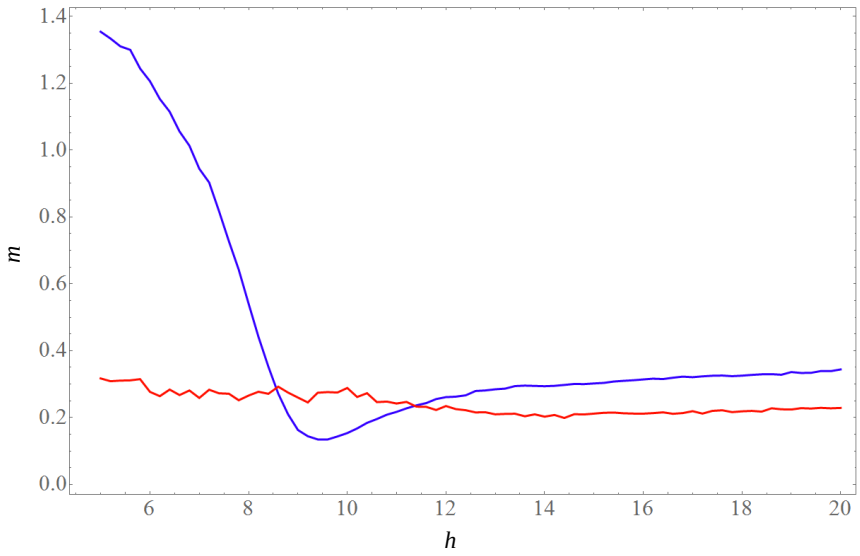


Рис. 3. Средние значения ошибок фильтрации m_e

Fig. 3. Mean values of filtering errors m_e

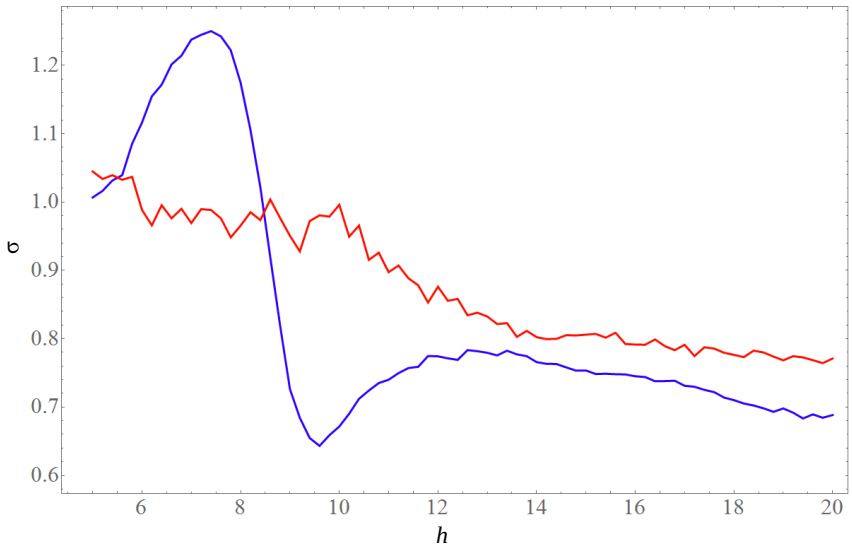


Рис. 4. Среднеквадратические отклонения ошибок фильтрации σ_e

Fig. 4. Standard deviations of filtering errors σ_e

Представленные на рис. 3 и 4 графики для $n = 0$ (синего цвета) и $n = 1$ (красного цвета) дают возможность уточнять величины порога h, исходя из допустимых величин ошибок. Из графиков видно, что установившиеся значения ошибок фильтрации наступают при $h > 12$. При этом средние значения ошибок при $n = 0$ больше, чем при $n = 1$, а среднеквадратические отклонения при $n = 0$ меньше, чем при $n = 1$. Однако при $n = 1$ имели место срывы слежения. Так, на интервале $30 < k < 100$ срывы происходили с вероятностью 0,47. Заметим, что при $m_x = 0$ среднее значение ошибки $m_e = 0$.

Заключение. Для нахождения алгоритма фильтрации при перерывах поступления информации в работе использован расширенный метод наименьших квадратов. Алгоритм позволяет определять моменты наступления перерывов и осуществлять экстраполяцию оценок полезной информации. Полученный фильтр относится к классу нелинейных, что затрудняет его аналитический анализ. Первоначальный выбор параметров фильтра осуществляется приближенно, а затем они уточняются по результатам моделирования. Сравнительная простота алгоритма делает его пригодным для практического использования.

Вклад авторов. *В. М. Артемьев* – постановка задачи, анализ существующих методов и разработка алгоритма фильтрации. *А. О. Наумов* – математическое моделирование, расчет ошибок фильтрации, анализ результатов и оформление статьи.

Список использованных источников

1. Blackman, S. Design and Analysis of Modern Tracking Systems / S. Blackman, R. Popoli. – Norwood : Artech House, 1999. – 1230 p.

2. Казаков, И. Е. Анализ систем случайной структуры / И. Е. Казаков, В. М. Артемьев, В. А. Бухалев. – М. : Физматлит, 1993. – 272 с.

3. Степанов, О. А. Основы теории оценивания с приложениями к задачам обработки навигационной информации : в 2 ч. Ч. 1. Введение в теорию оценивания / О. А. Степанов. – СПб. : Электроприбор, 2010. – 509 с.

4. Артемьев, В. М. Линейная фильтрация многомерных случайных последовательностей расширенным методом наименьших квадратов / В. М. Артемьев, А. О. Наумов, Л. Л. Кохан // Информатика. – 2016. – № 4(52). – С. 51–56.

5. Иванов, В. А. Математические основы теории автоматического регулирования / В. А. Иванов, В. С. Медведев, Б. К. Чемоданов. – М. : Высш. шк., 1971. – 808 с.

6. Тихонов, В. И. Выбросы траекторий случайных процессов / В. И. Тихонов, В. И. Хименко. – М. : Наука, 1987. – 304 с.

Referenses

1. Blackman S., Popoli R. *Design and Analysis of Modern Tracking Systems*. Norwood, Artech House, 1999, 1230 p.

2. Kazakov I. E., Artem'ev V. M., Buhalev V. A. Analiz sistem sluchajnoj struktury. *Analysis of Systems with Random Structure.* Moscow, Fizmatlit, 1993, 272 p. (In Russ.)

3. Stepanov O. A. Osnovy teorii ocenivanija s prilozhenijami k zadacham obrabotki navigacionnoj informacii. Chast' 1. Vvedenie v teoriju ocenivanija. *Basics of the Estimation Theory with Applications to the Problems of Processing of Navigation Information*. *Part 1. Introduction to Estimation Theory*. Saint Petersburg, Jelektropribor, 2010, 509 p. (In Russ.)

4. Artemiev V. M., Naumov A. O., Kokhan L. L. *Linear filtering of random sequences using a least squares method with regularization*. Informatika [*Informatics*], 2016, no. 4(52), pp. 51−56 (In Russ.)

5. Ivanov V. A., Medvedev V. S., Chemodanov B. K. Matematicheskie osnovy teorii avtomaticheskogo regulirovanija. *Mathematical Foundations of the Theory of Automatic Control.* Moscow, Vysshaja shkola, 1971, 808 p. (In Russ.)

6. Tihonov V. I., Himenko V. I. Vybrosy traektorij sluchajnyh processov. *Outliers in Trajectories of Random Processes.* Moscow, Nauka, 1987, 304 p. (In Russ.)

Информация об авторах

Артемьев Валентин Михайлович, член-корреспондент Национальной академии наук Беларуси, доктор технических наук, профессор, главный научный сотрудник, Институт прикладной физики Национальной академии наук Беларуси.
E-mail: artemiev@iaph.bas-net.by

Наумов Александр Олегович, кандидат физико-математических наук, заведующий лабораторией радиотомографии, Институт прикладной физики Национальной академии наук Беларуси.
https://orcid.org/0000-0002-4624-9261
E-mail: naumov@iaph.bas-net.by

Information about the authors

Valentin M. Artemiev, Corresponding Member of the National Academy of Sciences of Belarus, D. Sc. (Eng.), Professor, Chief Researcher, Institute of Applied Physics of the National Academy of Sciences of Belarus.
E-mail: artemiev@iaph.bas-net.by

Alexander O. Naumov, Ph. D. (Phys.-Math.), Head of the Laboratory of Radiotomography, Institute of Applied Physics of the National Academy of Sciences of Belarus.
https://orcid.org/0000-0002-4624-9261
E-mail: naumov@iaph.bas-net.by

БИОИНФОРМАТИКА
BIOINFORMATICS

УДК 004.64, 577.21
https://doi.org/10.37661/1816-0301-2022-19-1-59-71

Оригинальная статья
Original Paper

Разработка базы данных мотивов регуляции транскрипции у бактерий

В. В. Скакун✉, Е. А. Николайчик

Белорусский государственный университет,
пр. Независимости, 4, Минск, 220030, Беларусь
✉*E-mail: skakun@bsu.by*

Аннотация

Цели. Объемы данных, генерируемые современными методами высокопроизводительного секвенирования, таковы, что их анализ выполняется преимущественно в автоматическом режиме. В частности, использование вновь расшифрованных геномных последовательностей возможно только после аннотации функциональных элементов генома, которая, как правило, выполняется автоматическими конвейерами. Такие конвейеры аннотации успешно справляются с идентификацией генов, но ни один из них не аннотирует регуляторные элементы, без которых нельзя понять, когда и как гены могут экспрессироваться. Информация о регуляторных элементах бактерий собрана в нескольких специализированных базах данных (RegulonDB, CollecTF, Prodoric2 и др.), однако только часть этой информации можно использовать для аннотации регуляторных элементов и только у очень ограниченного круга бактерий. Ранее авторами был предложен четкий формальный критерий для применения регуляторной информации к любым бактериальным геномам. Таким критерием стал CR-тег – последовательность аминокислотных остатков транскрипционного регулятора, специфически контактирующих с азотистыми основаниями регуляторного элемента в геномной ДНК. Связанная с CR-тегом математическая модель регуляторного элемента (мотив) может быть корректно применена для аннотации подобных элементов в любых геномах, кодирующих транскрипционный регулятор с идентичным CR-тегом. Накопление связанных с CR-тегами мотивов поставило вопрос об их упорядоченном хранении для удобства последующего применения при аннотации геномных последовательностей. Поскольку ни одна из известных баз данных не использует концепцию CR-тегов, потребовалась разработка новой базы данных. Таким образом, целью работы является создание базы данных с информацией о бактериальных транскрипционных факторах и распознаваемых ими последовательностях ДНК, пригодной для аннотации регуляторных последовательностей в бактериальных геномах.

Методы. Инфологическое моделирование предметной области производилось с помощью методологии IDEF1X. Разработка базы данных выполнялась посредством СУБД Microsoft SQL Server. Кроссплатформенное приложение по импорту данных в базу данных написано на языке C++ с использованием технологии Qt.

Результаты. В результате проведенного исследования предметной области была разработана и реализована в СУБД Microsoft SQL Server реляционная модель данных, позволяющая целостное хранение информации о накопленных мотивах регуляции транскрипции у бактерий, включая и информацию о публикациях, подтверждающих корректность этих мотивов. Для автоматизации процесса ввода накопленных данных разработано кроссплатформенное приложение для импорта структурированных данных о транскрипционных факторах.

Заключение. Основным отличием разработанной базы данных является использование концепции CR-тега. Записи математических моделей регуляторных элементов (мотивов) в базе данных связаны с CR-тегом и поэтому могут быть корректно применены для аннотации подобных элементов в любых геномах, кодирующих транскрипционный регулятор с идентичным CR-тегом. Разработанная база данных обеспечит структурированное и целостное хранение данных, а также их быстрый поиск при использовании в конвейере автоматической аннотации регуляторных элементов в бактериальных геномных последовательностях.

Ключевые слова: регуляция транскрипции, регуляторные мотивы, последовательности ДНК, CR-тег, программа SigmoID, базы данных

Благодарности. Работа выполнялась в рамках задания 1.10.5 ГПНИ «Цифровые и космические технологии, безопасность человека, общества и государства» (2021–2025).

Для цитирования. Скакун, В. В. Разработка базы данных мотивов регуляции транскрипции у бактерий / В. В. Скакун, Е. А. Николайчик // Информатика. – 2022. – Т. 19, № 1. – С. 59–71.
https://doi.org/10.37661/1816-0301-2022-19-1-59-71

Конфликт интересов. Авторы заявляют об отсутствии конфликта интересов.

Поступила в редакцию | Received 01.11.2021
Подписана в печать | Accepted 19.01.2022
Опубликована | Published 29.03.2022

Development of a bacterial regulatory motif database

Victor V. Skakun✉, Yevgeny A. Nikolaichik

Belarusian State University,
av. Nezavisimosti, 4, Minsk, 220030, Belarus
✉*E-mail: skakun@bsu.by*

Abstract

Objectives. The amount of data generated by modern methods of high-throughput sequencing is such that their analysis is performed mainly in automatic mode. In particular, the use of newly decoded genomic sequences is possible only after the annotation of functional elements of the genome, which, as a rule, is performed by automatic pipelines. Such annotation pipelines do a good job to identify the genes, but none of them annotate regulatory elements. Without these elements it is not possible to understand when and how genes can be expressed. Information on the regulatory elements of bacteria is collected in several specialized databases (RegulonDB, CollecTF, Prodoric2, etc.), however, only a part of this information can be used for annotation of regulatory elements, and only for a very limited range of bacteria. Previously, we proposed a clear formal criterion for applying regulatory information to any bacterial genome. Such a criterion is the CR tag, a sequence of amino acid residues of a transcriptional regulator that specifically contacts the nitrogenous bases of regulatory element in genomic DNA. The mathematical model of a regulatory element (motif) associated with a CR tag can be correctly applied to annotate similar elements in any genomes encoding a transcriptional regulator with an identical CR tag. The accumulation of motifs associated with CR tags raised the question of their ordered storage for the convenience of subsequent use in the annotation of genomic sequences. Since no one of well-known databases uses the concept of CR tags, a new database ought to be developed. Thus, the goal of this work is to create a database with information about bacterial transcription factors and DNA sequences recognized by them, suitable for annotation of regulatory sequences in bacterial genomes.

Methods. Infological modeling of the subject area was carried out using the IDEF1X methodology. The database was developed using the Microsoft SQL Server DBMS. A cross-platform application for importing data into a database is written in C++ using Qt technology.

Results. As a result of the study of the subject area, a relational data model was developed and implemented in the Microsoft SQL Server DBMS, which allows holistic storage of information about accumulated transcription regulation motifs in bacteria, including information about the publications confirming their correctness. To automate the process of entering accumulated data, a cross-platform application was developed for importing structured data on transcription factors.

Conclusion. The main difference of the developed database is the use of CR-tag concept. Records of mathematical models of regulatory elements (motifs) in the database are associated with a CR tag and, therefore, can be correctly used to annotate similar elements in any genomes encoding a transcriptional regulator with an identical CR tag. The developed database will provide structured and holistic data storage, as well as their quick search when used in the pipeline for automatic annotation of regulatory elements in bacterial genomic sequences.

Keywords: regulation of transcription, regulatory motifs, DNA sequences, CR tag, SigmoID program, databases

Acknowledgements. The work was carried out within the task 1.10.5 of the State Scientific Research Program "Digital and Space Technologies, Human, Society and State Security" (2021–2025).

For citation. Skakun V. V., Nikolaichik Y. A. *Development of a bacterial regulatory motif database*. Informatika [*Informatics*], 2022, vol. 19, no. 1, pp. 59−71 (In Russ.). https://doi.org/10.37661/1816-0301-2022-19-1-59-71

Conflict of interest. The authors declare no conflict of interest.

Введение. Идентификация регуляторных элементов геномов является одной из наиболее актуальных задач современной геномики. Особую важность этой задаче придает то, что в подавляющем большинстве геномных последовательностей, депонированных в нуклеотидных базах данных (БД), регуляторные элементы вообще не аннотированы. Одной из причин сложившейся ситуации является сложность статистически достоверной идентификации большинства регуляторных элементов в геномных масштабах. Тем не менее решение этой задачи возможно для прокариот, поскольку их геномы компактны, а регуляторные элементы расположены преимущественно в межгенных участках, занимающих порядка 10 % генома. При общем сходстве структур генома две группы прокариот, бактерии и археи, существенно отличаются по механизмам транскрипционной регуляции, поэтому дальнейшее обсуждение касается только бактерий.

Можно выделить три основных типа регуляторных элементов, контролирующих экспрессию генов у бактерий: промоторы, операторы и терминаторы [1]. В отличие от терминаторов промоторы и операторы являются в значительной степени геноспецифическими и очень вариабельными из-за распознавания их большим числом (несколькими сотнями) различных транскрипционных факторов. Бактериальные транскрипционные факторы в подавляющем большинстве случаев являются гомоолигомерами (чаще всего ди- или тетрамерами), в связи с этим типичный регуляторный элемент распознается двумя идентичными ДНК-связывающими доменами и имеет четко выраженную симметрию, что облегчает его идентификацию [1, 2]. Размеры сайтов связывания транскрипционных факторов (операторов) обычно варьируют в пределах 15–25 пар нуклеотидов [1, 2], поэтому с учетом суммарной длины всех регуляторных последовательностей прокариотического генома порядка нескольких сот тысяч пар нуклеотидов статистический анализ позволяет отличать регуляторные элементы от всех прочих геномных последовательностей. Действительно, простейшая математическая модель (весовая матрица) регуляторного элемента, созданная на основе нескольких десятков экспериментально охарактеризованных операторов для конкретного транскрипционного фактора, может быть успешно использована для идентификации операторов в родственных геномах [3, 4].

Вместе с тем проблемой является значительно более высокая скорость эволюции (и, соответственно, вариабельность) регуляторных элементов и генов транскрипционных факторов в сравнении с другими функционально значимыми участками геномов [5–7], из-за чего модель регуляторного элемента, справедливую для одного генома, нельзя применять к другому без доказательства идентичности контактов между белком-регулятором и оператором. Еще одной проблемой является то, что для построения надежной статистической модели регуляторного элемента требуется достаточное число (не менее 10) известных последовательностей, тогда как большинство транскрипционных факторов контролируют небольшие регулоны (например, у бактерий *E. coli* более половины транскрипционных факторов контролируют всего один-три оперона [8]), поэтому для них известно меньшее число мишеней. Сложилась парадоксальная ситуация: опубликовано много экспериментальных работ, характеризующих отдельные транскрипционные факторы, выявлены соответствующие операторные последовательности, но в большинстве случаев эту информацию нельзя непосредственно использовать для поиска со-

ответствующих операторов в других геномных последовательностях из-за отсутствия моделей регуляторных элементов и четких критериев их применения к конкретным геномам.

Существует несколько специализированных БД, обобщающих опубликованную информацию о регуляторных элементах и предлагающих их математические модели. Самой известной БД, собирающей всю доступную информацию об одном организме, является RegulonDB – специализированная БД для *Escherichia coli* [8]. Это наиболее полная БД для отдельно взятого организма, которая содержит информацию о промоторных последовательностях для всех семи сигма-факторов данной бактерии, а также об операторах для 221 из ~300 транскрипционных факторов. Однако только для малого количества представленных в RegulonDB транскрипционных факторов есть пригодные к непосредственному использованию модели операторных последовательностей, что обусловлено двумя основными причинами: для многих транскрипционных факторов число охарактеризованных операторов слишком мало; при достаточном числе операторов, охарактеризованных в разных публикациях, они часто имеют разную ширину или просто не выровнены из-за различий в их описании в разных экспериментальных работах. Такие БД, как CollecTF [9], Prodoric [10], CoryneRegNet [11], собирают экспериментально полученную регуляторную информацию уже для многих видов. Каждая из этих БД имеет определенную специализацию и свои достоинства, но вместе с тем и существенный недостаток – малое число операторных моделей, пригодных для непосредственного использования.

Среди БД с регуляторной информацией самой обширной является RegPrecise [12]. Версия 4.0 этой БД содержит информацию об операторных мотивах для 11 520 транскрипционных факторов, причем в большинстве случаев с моделями, пригодными для непосредственного использования, однако операторные последовательности в RegPrecise в подавляющем числе случаев выявлены методами сравнительной геномики и не имеют экспериментального подтверждения.

Для аннотации известных операторов и промоторов путем применения регуляторной информации из БД RegPrecise, RegulonDB и CollecTF к неохарактеризованным геномам была разработана программа SigmoID [13]. Полученный авторами опыт использования перечисленных выше БД для исследования транскрипционной регуляции у немодельных видов бактерий не имел большого успеха. Первая версия программы SigmoID хотя и позволяла применять регуляторную информацию из RegPrecise, RegulonDB и CollecTF к неохарактеризованным геномам, но не обладала четкими критериями корректности такого переноса регуляторной информации. Фактически пользователю предлагалось самостоятельно установить наличие у исследуемого организма ортологов известных транскрипционных факторов и принять решение о достаточности уровня гомологии между транскрипционными факторами для переноса регуляторной информации. В версии 2 программы SigmoID [14, 15] в качестве критерия возможности применения имеющейся операторной модели к исследуемой геномной последовательности взята высказанная ранее идея [16] о строгом соответствии оператора так называемому CR-тегу – последовательности аминокислотных остатков ДНК-связывающего домена транскрипционного фактора, непосредственно контактирующих с азотистыми основаниями оператора. CR-тег является уникальным идентификатором пары «транскрипционный фактор – операторный мотив», поэтому в версии 2 SigmoID все операторные мотивы связаны с CR-тегами своих транскрипционных факторов и применяются для аннотации операторов только в тех геномах, которые кодируют транскрипционный фактор с идентичным CR-тегом. Благодаря использованию концепции CR-тегов программа SigmoID версии 2.0 способна корректно аннотировать операторные последовательности любых бактериальных геномов в полностью автоматическом режиме с применением имеющейся коллекции калиброванных профилей операторных мотивов.

Результатом процесса анализа регуляторной информации с помощью SigmoID является набор папок (по одной для каждого транскрипционного фактора), содержащих пять текстовых файлов с общим описанием транскрипционного фактора и с данными о CR-теге, найденных операторах, регуляторном мотиве, описываемом профильной скрытой марковской моделью (hidden markov model, HMM) и позиционной весовой матрицей (position weight matrix, PWM), а также о параметрах поиска с использованием этих HMM и PWM. Дополнительно формируются два файла для хранения данных по доказательной базе, подтверждающей справедливость

мотива: публикации, результаты экспериментального подтверждения и сведения о лице, курирующем проведенные исследования. По мере накопления связанных с CR-тегами операторных моделей встает вопрос о хранении этой информации. Ни одна из общеизвестных на сегодня БД не использует концепцию CR-тегов, поэтому потребовалась разработка принципиально новой БД. Ее дизайн предполагает два ключевых отличия от имеющихся решений:

1) каждый транскрипционный фактор имеет математическую модель оператора (скрытую марковскую модель), непосредственно применимую для идентификации операторов в геномных последовательностях;

2) все операторные модели ассоциированы с CR-тегами.

Кроме того, несмотря на использование сравнительной геномики при конструировании операторных моделей, каждая такая модель обязательно должна иметь экспериментальное подтверждение корректности операторного мотива.

Для структурированного и целостного хранения набора взаимосвязанных данных наилучшим решением является применение технологии БД и, в частности, реляционной модели данных. БД может служить ядром системы обработки и анализа геномных данных с целью предсказания и верификации мотивов регуляции транскрипции у бактерий. Хранение в БД множества структурированных и взаимосвязанных данных предоставляет возможность дополнительного статистического анализа, что позволяет повысить детализацию результатов анализа и улучшить качество интерпретации данных. Таким образом, целью настоящей работы является разработка БД мотивов регуляции транскрипции у бактерий. Выполнению этой цели предшествует решение следующих задач: моделирование предметной области для создания схемы БД, создание БД с помощью некоторой системы управления БД СУБД и импорт в БД уже накопленного массива данных, что, в свою очередь, предполагает написание специальной программы, автоматизирующей процесс импорта данных из набора текстовых файлов.

Разработка базы данных. Начальным этапом разработки БД является этап инфологического моделирования, заключающийся в анализе предметной области и создании концептуальной модели данных, цель которой – максимально отразить семантику предметной области в терминах модели данных [17]. Стандартом де-факто здесь выступает технология IDEF1X (URL: https://www.idef.com/idef1-information_modeling_method/). Моделирование предметной области и разработка схемы БД выполнялись с помощью CASE (Computer-aided Software Engineering) системы DBDesigner Fork (URL : https://sourceforge.net/projects/dbdesigner-fork). Моделирование проводилось согласно технологии IDEF1X с отображением связей по нотации “Crow’s foot” («воронья ножка») [17].

При анализе предметной области выделены следующие высокоуровневые сущности: CRTags (CR-теги), TFs (транскрипционные факторы), TF_families (семейства транскрипционных факторов), Motifs (мотивы) и Operators (операторы). Транскрипционные факторы описываются посредством следующих атрибутов: названия, CR-тега, идентификатора в некоторой референсной БД и самой последовательности транскрипционного фактора. Для однозначного определения семейства транскрипционного фактора кроме его названия добавлен и идентификатор Accession (код доступа) соответствующей референсной БД (PFAM или SMART [18, 19]). Регуляторные мотивы представляются посредством профильной HMM и (или) PWM. Сайты связывания транскрипционных факторов (операторы) описываются идентификатором и собственно самой последовательностью сайта связывания. Между сущностями TFs и Motifs установлено отношение «один ко многим», так как для одного транскрипционного фактора может быть описано несколько регуляторных мотивов. Для хранения параметров анализа введена сущность Settings (настройки). Задание параметров в виде пары {Name (название параметра), Value (значение параметра)} позволяет хранить произвольное количество любых параметров. Для хранения детализированной информации, обосновывающей найденные мотивы и операторы транскрипционных факторов, введены сущности Publications (публикации), Curators (кураторы), Evidence_types (типы подтверждения), а также связующие сущности Motifs_curators и Motif_references, которые реализуют отношения

«многие-ко-многим», существующие между сущностями Motifs, Curators и Publications, а также Motifs_references и Evidence_types соответственно. Перечисленные свойства обеспечивают модели данных универсальность и инвариантность к различным видам анализа.

Разработанная схема БД представлена на рис. 1, где пиктограмма ключа – это первичный ключ, красный ромбик слева от названия поля означает, что для данного поля задано ограничение ссылочной целостности. Индексы перечислены внизу каждой таблицы. Схема содержит 12 сущностей и может быть транслирована в реляционную модель, предоставляя уровень нормализации не ниже третьей нормальной формы [17]. В ней учтены необходимые валидаторы значений (поле Email) и ограничения ссылочной целостности. По внешним ключам, осуществляющим связь с сущностями, для которых предполагается много экземпляров, созданы индексы в целях повышения скорости выполнения многотабличных запросов, поиска и фильтрации данных. С помощью индексов реализовано требование уникальности значений полей Accession, CRTag, ProteinID, Email. Прочие поля, для которых предполагается проведение поиска, например Name, Date и др., также проиндексированы.

В распоряжении авторов имеется доступ к серверу под управлением ОС Microsoft Windows Server 2012 с установленной на ней системой управления БД Microsoft SQL Server 2017 (URL: https://www.microsoft.com/en-us/sql-server/sql-server-2017). Данная СУБД относится к разряду промышленных высокопроизводительных и надежных решений и способна эффективно решать задачи хранения и обработки больших наборов данных, включая данные по регуляции транскрипции у бактерий. Соответственно, на следующем этапе разработанная схема БД была транслирована в реляционную модель с учетом требований вышеуказанной СУБД и развернута на сервере. Для создания БД использовалась среда разработки Microsoft SQL Server Management Studio (URL: https://docs.microsoft.com/en-us/sql/ssms/download-sql-server-management-studio-ssms).

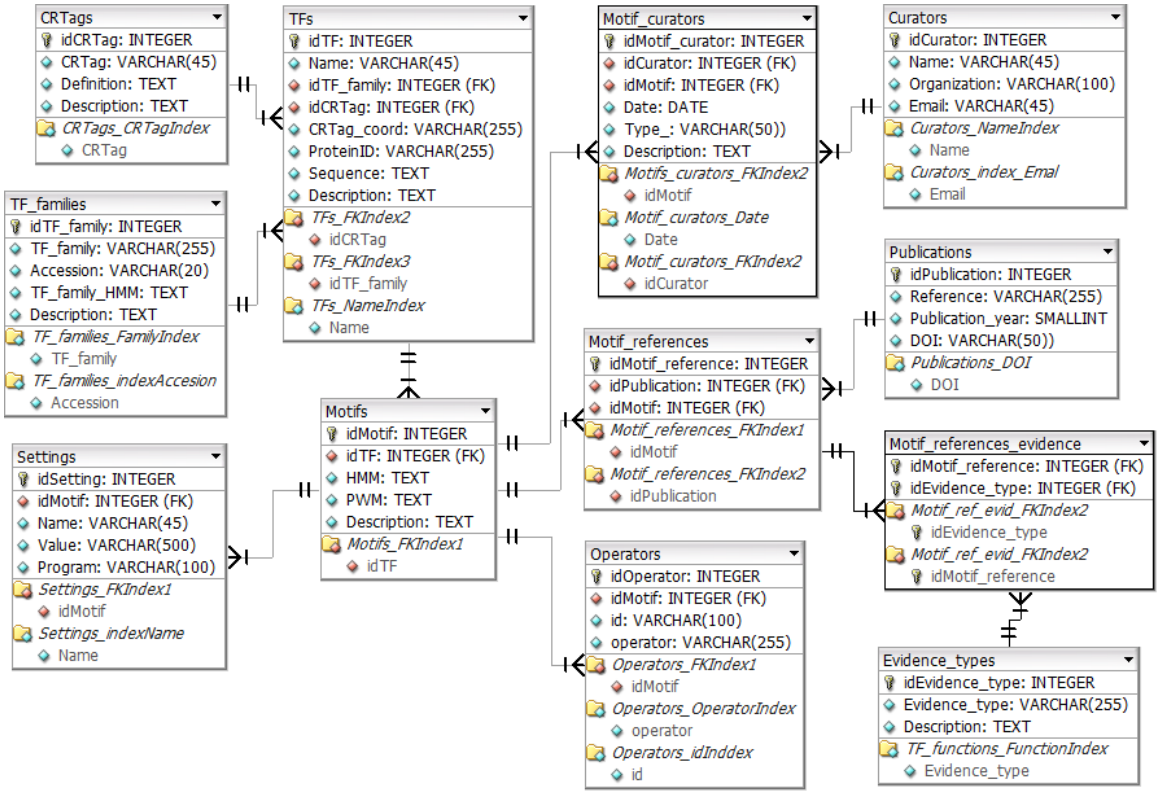


Рис. 1. Схема базы данных мотивов регуляции транскрипции у бактерий в формате IDEF1X

Fig. 1. Schematic of the database of bacterial transcription regulation motifs in the IDEF1X format

Типы и размеры некоторых полей были изменены. Благодаря тому что SQL Server [20] позволяет определять текстовые поля с максимальным размером в 8000 Б, тип полей Definition (отношение CRTags) и Sequence (отношение TFs) был заменен с TEXT на VARCHAR(8000). Такая замена позволяет строить эффективные индексы по вышеуказанным полям в целях ускорения поиска и доступа к данным и дополнительно определять уникальность значений в этих полях.

Учитывая то, что SQL Server не позволяет создавать индексы с требованием уникальности значений для полей с допуском пустых значений, ряд индексов был сформирован с добавлением предиката IS NULL, например:

```
CREATE UNIQUE NONCLUSTERED INDEX idx_TFs_ProteinID_notnull ON TFs(ProteinID) WHERE Pro-
teinID IS NOT NULL;
```

В целях упрощения и стандартизации доступа к данным был разработан набор представлений. Представления по своей сути являются виртуальными таблицами, получающимися в результате выполнения определенного запроса к БД, т. е. представляют собой именованный запрос [17]. Они определяют логическую независимость от данных и интерфейс пользователя для доступа к ним. Приведем SQL-код [21] основных представлений.

Представление TFs_TF_familiesView предназначено для просмотра объединенной информации по транскрипционным факторам и их семействам (основано на таблицах TFs, CRTags и TF_families с правым внешним соединением [21] таблиц TFs и TF_families):

```
CREATE VIEW TFs_TF_familiesView AS
SELECT dbo.TF_families.TF_family, dbo.TF_families.Accession,
dbo.TF_families.TF_family_HMM, dbo.TF_families.Description AS TF_family_description,
dbo.TFs.idTF, dbo.TFs.Name AS TF_name, dbo.CRTags.CRTag, dbo.TFs.CRTag_coord,
dbo.TFs.ProteinID, dbo.TFs.Sequence,  dbo.TFs.Description AS TF_Description FROM
dbo.CRTags INNER JOIN dbo.TFs ON dbo.CRTags.idCRTag = dbo.TFs.idCRTag RIGHT OUTER JOIN
dbo.TF_families ON dbo.TFs.idTF_family = dbo.TF_families.idTF_family;
```

В коде представления dbo есть название схемы, принадлежащей владельцу БД (database owner). Внешнее соединение таблиц позволяет вывести все семейства независимо от того, есть ли в БД хотя бы один транскрипционный фактор данного семейства. Пример вызова представления TFs_TF_familiesView:

```
SELECT TF_family, Accession, TF_family_HMM, TF_family_description, idTF, TF_name, CRTag,
CRTag_coord, ProteinID, Sequence, TF_Description FROM TFs_TF_familiesView;
```

Представление MotifsView предназначено для просмотра объединенной информации по мотивам (основано на таблицах Motifs, TFs, CRTags и TF_families):

```
CREATE VIEW MotifsView AS
SELECT dbo.Motifs.idMotif, dbo.TFs.Name, dbo.CRTags.CRTag, dbo.TF_families.TF_family,
dbo.TF_families.Accession AS TF_family_accession, dbo.TF_families.Description AS
TF_family_description, dbo.TF_families.TF_family_HMM, dbo.Motifs.HMM, dbo.Motifs.PWM,
dbo.TFs.CRTag_coord, dbo.TFs.ProteinID, dbo.TFs.Sequence, dbo.TFs.Description AS
TF_description, dbo.Motifs.Description AS Motif_description FROM dbo.Motifs INNER JOIN
dbo.TFs ON dbo.Motifs.idTF = dbo.TFs.idTF INNER JOIN dbo.CRTags ON dbo.TFs.idCRTag =
dbo.CRTags.idCRTag INNER JOIN dbo.TF_families ON dbo.TFs.idTF_family =
dbo.TF_families.idTF_family;
```

Пример вызова представления MotifsView:

```
SELECT idMotif, Name, CRTag, TF_family, TF_family_accession, TF_family_description,
TF_family_HMM, HMM, PWM, CRTag_coord, ProteinID, Sequence, TF_description, Mo-
tif_description FROM MotifsView;
```

Представление OperatorsView предназначено для просмотра списка операторов определенного мотива (основано на таблицах Motifs, TFs, CRTags, TF_families и Operators):

```
CREATE VIEW OperatorsView AS
SELECT dbo.Operators.idOperator, dbo.Operators.ID, dbo.Operators.Operator, dbo.TFs.Name
AS TF_name, dbo.TF_families.TF_family, dbo.CRTags.CRTag, dbo.Motifs.idMotif,
dbo.TFs.idTF FROM dbo.Motifs INNER JOIN dbo.Operators ON dbo.Motifs.idMotif =
dbo.Operators.idMotif INNER JOIN dbo.TFs ON dbo.Motifs.idTF = dbo.TFs.idTF INNER JOIN
dbo.CRTags ON dbo.TFs.idCRTag = dbo.CRTags.idCRTag INNER JOIN dbo.TF_families ON
dbo.TFs.idTF_family = dbo.TF_families.idTF_family;
```

Пример вызова представления OperatorsView с выборкой для мотива с идентификатором 2:

```
SELECT idOperator, ID, Operator, TF_name, TF_family, CRTag, idMotif, idTF FROM Opera-
torsView WHERE idMotif = 2;
```

Представление MotifSettingsView предназначено для просмотра параметров определенного мотива (основано на таблицах Motifs, TFs, CRTags, TF_families и Settings):

```
CREATE VIEW MotifSettingsView AS
SELECT dbo.Settings.idSetting, dbo.Settings.Name, dbo.Settings.Value,
dbo.Settings.Program, dbo.TFs.Name AS TF_name, dbo.CRTags.CRTag,
dbo.TF_families.TF_family, dbo.Motifs.idMotif, dbo.TFs.idTF FROM dbo.Settings INNER JOIN
dbo.Motifs ON dbo.Settings.idMotif = dbo.Motifs.idMotif INNER JOIN dbo.TFs ON
dbo.Motifs.idTF = dbo.TFs.idTF INNER JOIN dbo.CRTags ON dbo.TFs.idCRTag =
dbo.CRTags.idCRTag INNER JOIN dbo.TF_families ON dbo.TFs.idTF_family =
dbo.TF_families.idTF_family;
```

Пример вызова представления MotifSettingsView с выборкой для транскрипционного фактора с идентификатором 7:

```
SELECT idSetting, Name, Value, Program, TF_name, CRTag, TF_family, idMotif, idTF FROM
MotifSettingsView WHERE idTF = 7;
```

Представление ReferencesView предназначено для просмотра списка публикаций, подтверждающих определенный мотив (основано на таблицах Motifs, TFs, Publications и Motif_references):

```
CREATE VIEW ReferencesView AS
SELECT dbo.TFs.Name AS TF_name, dbo.Publications.Reference,
dbo.Publications.Publication_year, dbo.Publications.DOI, idTF, idMotif FROM
dbo.Motif_references INNER JOIN dbo.Motifs ON dbo.Motif_references.idMotif =
dbo.Motifs.idMotif INNER JOIN dbo.Publications ON dbo.Motif_references.idPublication =
dbo.Publications.idPublication INNER JOIN dbo.TFs ON dbo.Motifs.idTF = dbo.TFs.idTF;
```

Пример вызова представления ReferencesView с выборкой для транскрипционного фактора GVRTDVTRR_WalR:

```
SELECT TF_name, Reference, Publication_year, DOI, idTF, idMotif FROM ReferencesView
WHERE TF_name = 'GVRTDVTRR_WalR';
```

Разработка программы импорта данных. Результатом работы программы SigmoID является набор текстовых файлов (от пяти до семи). Два файла формируются в результате деятельности сторонних программ из пакета MEME Suite (https://meme-suite.org/meme/) [22], используемых SigmoID в своей работе. Остальные файлы экспортируются непосредственно самой SigmoID.

Для выполнения импорта данных разработан алгоритм, основанный на создании промежуточного контейнера данных в виде словаря (map), хранящего параметры анализа и другие данные в виде множества пар «ключ – значение» (URL: https://doc.qt.io/qt-5/qmap.html#details). Удобством словаря является наличие быстрого (индексированного) поиска значений по ключу. Вначале файлы поочередно считываются в оперативную память компьютера и происходит заполнение словаря. Пустые строки и строки с комментариями игнорируются. Как только встречаются требуемые к импорту данные, происходит вставка записи в словарь. Ключ формируется исходя из названия поля в БД, куда впоследствии планируется запись соответствующего значения, или же исходя из контекста

импортируемых данных. На следующем этапе ведущим становится уже БД, вернее ее структура. Поскольку перечень требуемых к импорту данных точно известен, импорт данных в БД происходит согласно этому перечню. Если параметр или другие данные находятся в словаре, происходит вставка записи в БД. Если поиск в словаре завершается возвратом пустого значения, ничего не вставляется. Такой подход позволяет избавиться от множества рутинных проверок во время импорта и гарантирует надежность всего процесса импорта. Недостатками предложенного алгоритма являются более высокие требования к объему оперативной памяти и наличие повторного поиска данных в словаре. Учитывая то, что весь объем импортируемых данных, как правило, не превышает 50 кБ, а поиск по ключу в словаре очень быстрый, указанные недостатки не оказывают существенного влияния на эффективность процесса импорта.

Поскольку программа SigmoID является кроссплатформенной, доступна под лицензией GPL 3.0 и скомпилирована для трех основных десктопных ОС (GNU/Linux, macOS и Windows), для реализации алгоритма импорта данных была выбрана технология Qt (URL: https://www.qt.io/). Языком программирования в Qt является высокоуровневый язык C++ (а также Python). Технология Qt позволяет писать единый исходный код, который, будучи скомпилированным средствами данной технологии, доступными для соответствующей ОС, приводит к получению программного продукта, выполняющегося в нативном режиме в этой ОС. Наличие полноценной графической библиотеки, а также ряда других библиотек (например, библиотеки доступа к БД) делает технологию Qt очень привлекательной для разработки кроссплатформенных графических приложений. Для доступа к данным, хранящимся на сервере под управлением СУБД Microsoft SQL Server, использовались модуль QtSQL и драйвер QODBC3, осуществляющий соединение с SQL Server по технологии ODBC (URL: https://docs.microsoft.com/en-us/sql/odbc).

Для создания словаря применялся класс QMap. Особенностью технологии Qt является эффективная реализация собственной библиотеки стандартных компонентов, в состав которой и входит класс QMap. Импорт данных из текстовых файлов выполняется в потоковом виде с помощью классов QFile, QIODevice и QTextStream. Соединение с БД производилось с помощью класса QSqlDatabase и драйвера QODBC3, осуществляющего соединение с СУБД SQL Server по технологии ODBC. Для вставки записей в БД использовался класс QSqlQuery, позволяющий выполнять запросы к СУБД, написанные на языке SQL [21].

С целью упрощения процесса вставки записей в базовые отношения CRTags, TF_families, Curators, Publications и EvidenceTypes были написаны сохраненные процедуры на языке TransactSQL [18], являющемся языком программирования серверной логики СУБД SQL Server. Сохраненная процедура хранится и осуществляется на сервере, что предоставляет намного более эффективный способ взаимодействия с СУБД, если планируется выполнение нескольких инструкций SQL с проверкой результатов и условий их запуска. Чтобы не дублировать вставку одного и того же CR-тега, семейства транскрипционного фактора, публикации или куратора, производится вначале поиск записи, содержащей соответствующий CR-тег, семейство и т. д. Если запись найдена, возвращается первичный ключ данной записи, в противном случае вставляется запись с последующим возвратом ее первичного ключа. Приведем код сохраненной процедуры для вставки CR-тега:

```
CREATE PROCEDURE [dbo].[pr_addCRTag] @ID int = null OUTPUT, @CRTag varchar(255) AS
 if NOT EXISTS(SELECT * FROM CRTags WHERE CRTag = @CRTag)
   begin
      INSERT INTO CRTags(CRTag) VALUES(@CRTag)
      SET @ID = SCOPE_IDENTITY()
   end
 else
     SELECT @ID = idCRTag FROM CRTags WHERE CRTag = @CRTag
```

Остальные процедуры имеют аналогичный синтаксис. Возврат значения первичного ключа производится через первый параметр процедуры, описанный с ключевым словом OUTPUT.

Интерфейс разработанного приложения SigmoID data importer по импорту данных в БД (скомпилированный для ОС Windows) изображен на рис. 2.

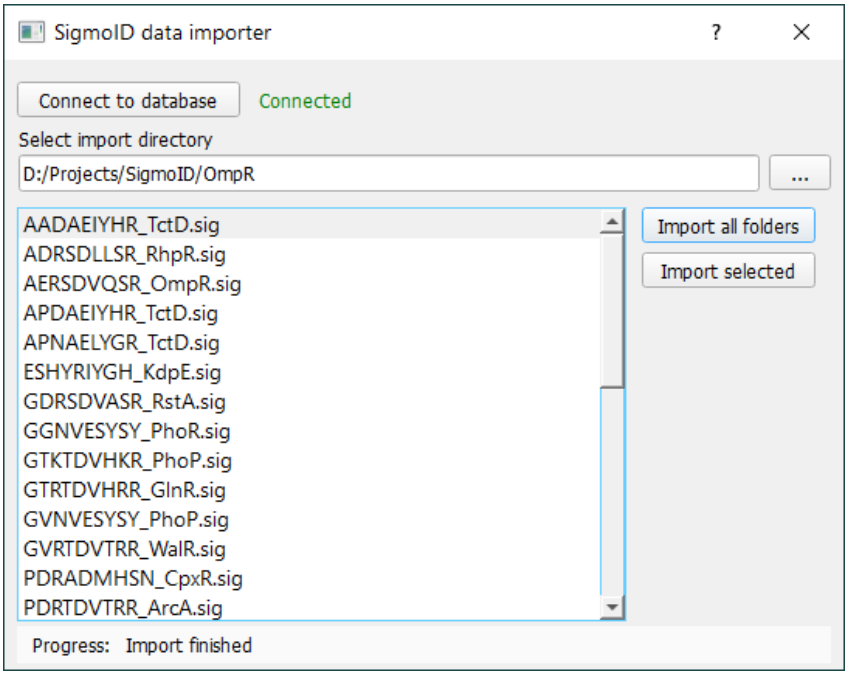


Рис. 2. Внешний вид приложения по импорту данных в БД

Fig. 2. Appearance of the database import application

Заключение. В результате моделирования предметной области была получена инфологическая модель БД мотивов регуляции транскрипции у бактерий. Модель транслирована в реляционную модель и развернута на сервере под управлением СУБД SQL Server. Разработанная БД протестирована путем ввода данных и их обновления с помощью графического клиентского приложения Microsoft SQL Management Studio. Удобство и логичность выполнения действий по вводу данных, полученных из дистрибутива программы SigmoID, доказали соответствие разработанной модели предметной области. Объявленные ограничения на вводимые значения, требования уникальности ввода и правила ссылочной целостности позволяют обеспечить целостность и согласованность данных при их вводе и в процессе дальнейшей работы.

Для облегчения и автоматизации процесса ввода данных в БД разработан и реализован в кроссплатформенном приложении SigmoID data importer алгоритм импорта наборов текстовых файлов, являющихся результатом работы программы SigmoID. Алгоритм основан на промежуточном чтении данных из файлов в индексируемый контейнер типа словаря (map), что позволяет обеспечивать корректное чтение данных из набора файлов, имеющих несколько версий своих структур. Для текущей версии БД разработан набор представлений, предоставляющих удобный интерфейс доступа к хранящимся данным.

В дальнейшем авторы планируют разработать веб-интерфейс к БД и создать полноценную интегрированную систему по обработке и хранению данных мотивов регуляции транскрипции у бактерий.

Вклад авторов. *В. В. Скакун* разработал базу данных и программу импорта данных. *Е. А. Николайчик* сформулировал задачу, обосновал актуальность работы и адаптировал формат файлов программы SigmoID для удобства импорта в базу данных. Инфологическое моделирование предметной области, разработка алгоритма импорта данных и подготовка текста статьи выполнялись совместно обоими авторами.

Список использованных источников

1. Van Hijum, S. A. F. T. Mechanisms and evolution of control logic in prokaryotic transcriptional regulation / S. A. F. T. van Hijum, M. H. Medema, O. P. Kuipers // Microbiology and Molecular Biology Reviews. – 2009. – Vol. 73, no. 3. – P. 481–509. https://doi.org/10.1128/MMBR.00037-08

2. Browning, D. F. Local and global regulation of transcription initiation in bacteria / D. F. Browning, S. J. W. Busby // Nature Reviews Microbiology. – 2016. – Vol. 14, no. 10. – P. 638–650. https://doi.org/10.1038/nrmicro.2016.103

3. Stormo, G. D. DNA binding sites: representation and discovery / G. D. Stormo // Bioinformatics. – 2000. – Vol. 16, no. 1. – P. 16–23. https://doi.org/10.1093/bioinformatics/16.1.16

4. Rodionov, D. A. Comparative genomic reconstruction of transcriptional regulatory networks in bacteria / D. A. Rodionov // Chemical Reviews. – 2007. – Vol. 107, no. 8. – P. 3467–3497. https://doi.org/10.1021/cr068309+

5. Gelfand, M. S. Evolution of transcriptional regulatory networks in microbial genomes / M. S. Gelfand // Current Opinion in Structural Biology. – 2006 – Vol. 16, no. 3. – P. 420–429. https://doi.org/10.1016/j.sbi.2006.04.001

6. Lozada-Chavez, I. Bacterial regulatory networks are extremely flexible in evolution / I. Lozada-Chavez // Nucleic Acids Research. – 2006. – Vol. 34, no. 12. – P. 3434–3445. https://doi.org/10.1093/nar/gkl423

7. Perez, J. C. Evolution of transcriptional regulatory circuits in bacteria / J. C. Perez, E. A. Groisman // Cell. – 2009. – Vol. 138, no. 2. – P. 233–244. https://doi.org/10.1016/j.cell.2009.07.002

8. RegulonDB v 10.5: tackling challenges to unify classic and high throughput knowledge of gene regulation in E. coli K-12 / A. Santos-Zavaleta [et al.] // Nucleic Acids Research. – 2019. – Vol. 47, no. D1. – P. D212–D220. https://doi.org/10.1093/nar/gky1077

9. CollecTF: a database of experimentally validated transcription factor-binding sites in Bacteria / S. Kılıç [et al.] // Nucleic Acids Research. – 2014. – Vol. 42, iss. D1. – P. D156–D160. https://doi.org/10.1093/nar/gkt1123

10. PRODORIC (release 2009): a database and tool platform for the analysis of gene regulation in prokaryotes / A. Grote [et al.] // Nucleic Acids Research. – 2009. – Vol. 37, iss. suppl_1. – P. D61–D65. https://doi.org/10.1093/nar/gkn837

11. CoryneRegNet 7, the reference database and analysis platform for corynebacterial gene regulatory networks / M. T. D. Parise [et al.] // Scientific Data. – 2020. – Vol. 7, no. 1. – P. 142. https://doi.org/10.1038/s41597-020-0484-9

12. RegPrecise 3.0 – A resource for genome-scale exploration of transcriptional regulation in bacteria / P. S. Novichkov [et al.] // BMC Genomics. – 2013. – Vol. 14. – P. 745. https://doi.org/10.1186/1471-2164-14-745

13. Nikolaichik, Y. SigmoID: a user-friendly tool for improving bacterial genome annotation through analysis of transcription control signals / Y. Nikolaichik, A. U. Damienikan // PeerJ. – 2016. – Vol. 4. – P. e2056. https://doi.org/10.7717/peerj.2056

14. Nikolaichik, Y. Genome-wide inference of bacterial transcription factor binding sites: new method and its applications / Y. Nikolaichik, P. Vychik // BMC Bioinformatics. – 2020. – Vol. 21, no. S20. – P. O2. https://doi.org/10.1186/s12859-020-03838-2

15. Nikolaichik, Y. New approach to genome-wide automated inference of bacterial transcription factor binding sites / Y. Nikolaichik, P. Vychik // Abstracts of the XII Intern. Multiconf. "Bioinformatics of Genome Regulation and Structure/Systems Biology". – Novosibirsk, 2020. – P. 75–76. https://doi.org/10.18699/BGRS/SB-2020-046

16. Sahota, G. Novel sequence-based method for identifying transcription factor binding sites in prokaryotic genomes / G. Sahota, G. D. Stormo // Bioinformatics. – 2010. – Vol. 26, no. 21. – P. 2672–2677. https://doi.org/10.1093/bioinformatics/btq501

17. Скакун, В. В. Системы управления базами данных : пособие / В. В. Скакун. – Минск : БГУ, 2020. – 159 с.

18. The Pfam protein families database: towards a more sustainable future / R. D. Finn [et al.] // Nucleic Acids Research. – 2016. – Vol. 44, no. D1. – P. D279–D285. https://doi.org/10.1093/nar/gkv1344

19. Letunic, I. 20 years of the SMART protein domain annotation resource / I. Letunic, P. Bork // Nucleic Acids Research. – 2018. – Vol. 46, no. D1. – P. D493–D496. https://doi.org/10.1093/nar/gkx922

20. Нильсен, П. SQL Server 2005. Библия пользователя : пер с англ. / П. Нильсен. – М. : Вильямс, 2008. – 1232 с.

21. Грофф, Д. П. SQL. Полное руководство : пер. с англ. / Д. П. Грофф, П. Н. Вайнберг, Э. Д. Оппель. – 3-е изд. – М. : Вильямс, 2016. – 960 с.

22. The MEME Suite / T. L. Bailey [et al.] // Nucleic Acids Research. – 2015. – Vol. 43, no. W1. – P. W39–W49. https://doi.org/10.1093/nar/gkv416

References

1. Van Hijum S. A. F. T., Medema M. H., Kuipers O. P. Mechanisms and evolution of control logic in prokaryotic transcriptional regulation. *Microbiology and Molecular Biology Reviews*, 2009, vol. 73, no. 3, pp. 481–509. https://doi.org/10.1128/MMBR.00037-08

2. Browning D. F., Busby S. J. W. Local and global regulation of transcription initiation in bacteria. *Nature Reviews Microbiology*, 2016, vol. 14, no. 10, pp. 638–650. https://doi.org/10.1038/nrmicro.2016.103

3. Stormo G. D. DNA binding sites: representation and discovery. *Bioinformatics*, 2000, vol. 16, no.1, pp. 16–23. https://doi.org/10.1093/bioinformatics/16.1.16

4. Rodionov D. A. Comparative genomic reconstruction of transcriptional regulatory networks in bacteria. *Chemical Reviews*, 2007, vol. 107, no. 8, pp. 3467–3497. https://doi.org/10.1021/cr068309+

5. Gelfand M. S. Evolution of transcriptional regulatory networks in microbial genomes. *Current Opinion in Structural Biology*, 2006, vol. 16, no. 3, pp. 420–429. https://doi.org/10.1016/j.sbi.2006.04.001

6. Lozada-Chavez I. Bacterial regulatory networks are extremely flexible in evolution. *Nucleic Acids Research*, 2006, vol. 34, no. 12, pp. 3434–3445. https://doi.org/10.1093/nar/gkl423

7. Perez J. C., Groisman E. A. Evolution of transcriptional regulatory circuits in bacteria. *Cell*, 2009, vol. 138, no. 2, pp. 233–244. https://doi.org/10.1016/j.cell.2009.07.002

8. Santos-Zavaleta A., Salgado H., Gama-Castro S., Sánchez-Pérez M., Gómez-Romero L., …, Collado-Vides J. RegulonDB v 10.5: tackling challenges to unify classic and high throughput knowledge of gene regulation in E. coli K-12. *Nucleic Acids Research*, 2019, vol. 47, no. D1, pp. D212–D220. https://doi.org/10.1093/nar/gky1077

9. Kılıç S., White E. R., Sagitova D. M., Cornish J. P., Erill I. CollecTF: a database of experimentally validated transcription factor-binding sites in Bacteria. *Nucleic Acids Research*, 2014, vol. 42, iss. D1, pp. D156–D160. https://doi.org/10.1093/nar/gkt1123

10. Grote A., Klein J., Retter I., Haddad I., Behling S., …, Münch R. PRODORIC (release 2009): a database and tool platform for the analysis of gene regulation in prokaryotes. *Nucleic Acids Research*, 2009, vol. 37, iss. suppl_1, pp. D61–D65. https://doi.org/10.1093/nar/gkn837

11. Parise M. T. D., Parise D., Kato R. B., Pauling J. K., Tauch A., …, Baumbach J. CoryneRegNet 7, the reference database and analysis platform for corynebacterial gene regulatory networks. *Scientific Data*, 2020, vol. 7, no. 1, p. 142. https://doi.org/10.1038/s41597-020-0484-9

12. Novichkov P. S., Kazakov A. E., Ravcheev D. A., Leyn S. A., Kovaleva G. Y., …, Rodionov D. A. RegPrecise 3.0 – A resource for genome-scale exploration of transcriptional regulation in bacteria. *BMC Genomics*, 2013, vol. 14, p. 745. https://doi.org/10.1186/1471-2164-14-745

13. Nikolaichik Y., Damienikan A. U. SigmoID: a user-friendly tool for improving bacterial genome annotation through analysis of transcription control signals. *PeerJ*, 2016, vol. 4, p. e2056. https://doi.org/10.7717/peerj.2056

14. Nikolaichik Y., Vychik P. Genome-wide inference of bacterial transcription factor binding sites: new method and its applications. *BMC Bioinformatics*, 2020, vol. 21, no. S20, p. O2. https://doi.org/10.1186/ s12859-020-03838-2

15. Nikolaichik Y., Vychik P. New approach to genome-wide automated inference of bacterial transcription factor binding sites. *Abstracts of the XII International Multiconference "Bioinformatics of Genome Regulation and Structure/ Systems Biology"*. Novosibirsk, 2020, pp. 75–76. https://doi.org/10.18699/BGRS/SB-2020-046

16. Sahota G., Stormo G. D. Novel sequence-based method for identifying transcription factor binding sites in prokaryotic genomes. *Bioinformatics*, 2010, vol. 26, no. 21, pp. 2672–2677. https://doi.org/10.1093/bioinformatics/btq501

17. Skakun V. V. Sistemy upravlenija bazami dannyh. *Database Managements Systems*. Minsk, Belorusskij gosudarstvennyj universitet, 2020, 159 p. (In Russ.).

18. Finn R. D., Coggill P., Eberhardt R. Y., Eddy S. R., Mistry J., …, Bateman A. The Pfam protein families database: towards a more sustainable future. *Nucleic Acids Research*, 2016, vol. 44, no. D1, pp. D279–D285. https://doi.org/10.1093/nar/gkv1344

19. Letunic I., Bork P. 20 years of the SMART protein domain annotation resource. *Nucleic Acids Research*, 2018, vol. 46, no. D1, pp. D493–D496. https://doi.org/10.1093/nar/gkx922

20. Nielsen P. *Microsoft SQL Server 2005 Bible*. 1st ed. Wiley, 2006, 1344 p.

21. Groff J. R., Weinberg P. N., Oppel A. J. *SQL: The Complete Reference*, 3rd ed. McGraw Hill, 2009, 912 p.

22. Bailey T. L., Johnson J., Grant C. E., Noble W. S. The MEME Suite. *Nucleic Acids Research*, 2015, vol. 43, no. W1, pp. W39–W49. https://doi.org/10.1093/nar/gkv416

Информация об авторах

Скакун Виктор Васильевич, кандидат физико-математических наук, доцент, заведующий кафедрой, Белорусский государственный университет.
https://orcid.org/0000-0003-0880-4188
E-mail: skakun@bsu.by

Николайчик Евгений Артурович, кандидат биологических наук, доцент, Белорусский государственный университет.
https://orcid.org/0000-0002-6718-9309
E-mail: nikolaichik@bsu.by

Information about the authors

Victor V. Skakun, Ph. D. (Phys.-Math.), Associate Professor, Head of Department, Belarusian State University.
https://orcid.org/0000-0003-0880-4188
E-mail: skakun@bsu.by

Yevgeny A. Nikolaichik, Ph. D. (Biol.), Associate Professor, Belarusian State University.
https://orcid.org/0000-0002-6718-9309
E-mail: nikolaichik@bsu.by

ПАРАЛЛЕЛЬНЫЕ АРХИТЕКТУРЫ И ВЫЧИСЛЕНИЯ
PARALLEL ARCHITECTURE AND COMPUTING

УДК 519.168+519.873
https://doi.org/10.37661/1816-0301-2022-19-1-72-87

Оригинальная статья
Original Paper

Векторизация итерационных вычислительных процессов и оценки временного ускорения

В. М. Демиденко[1]✉, В. И. Бенедиктович[2]

[1]*Белорусский государственный экономический университет,*
пр. Партизанский, 26, Минск, 220070, Беларусь
✉*E-mail: vmdemidenko@yandex.ru*

[2]*Институт математики НАН Беларуси,*
ул. Сурганова, 11, Минск, 220072, Беларусь
E-mail: benedvi@gmail.com

Аннотация

Цели. Решается задача эффективной организации выполнения последовательных вычислительных процессов в векторном режиме с учетом возможностей современных высокопроизводительных векторно-конвейерных ЭВМ. Актуальность рассматриваемой задачи обусловлена тем, что такие процессы, возникающие при циклической обработке данных и в итерационных алгоритмах, являются наиболее сложными для распараллеливания. При решении задачи ставились три цели: построение математической модели, учитывающей основные архитектурные и вычислительные особенности современных векторно-конвейерных ЭВМ; расчет оптимального суммарного времени выполнения векторных операций; оценка временного выигрыша по сравнению с последовательным режимом обработки данных.

Методы. Для реализации поставленных целей и доказательства основных и вспомогательных утверждений применялся оригинальный метод, включающий установление справедливости индуктивных предположений в рассматриваемых случаях, а также иллюстративный метод теории расписаний, использующий диаграммы Ганта.

Результаты. Предложена векторная модель реализации последовательных вычислений, учитывающая основные особенности векторно-конвейерных ЭВМ. Определено оптимальное суммарное время выполнения последовательных вычислений в векторном режиме и получена нижняя оценка временного выигрыша по сравнению с последовательным режимом их выполнения.

Заключение. Установлено, что при обработке в последовательном режиме скалярных входных данных векторными операциями с длиной конвейера k возможно ускорение не менее чем в $nN/(nk+N)$ раз, где N – размер входа, n – число векторных и соответствующих им скалярных операций. Оценка временного ускорения при векторизации вычислений приводится в сравнении с последовательным режимом их выполнения.

Ключевые слова: конвейеризация, векторизация вычислений, показатель векторизуемости, временное ускорение, векторно-конвейерные вычислительные системы, распараллеливание вычислительных процессов

Благодарности. Исследование выполнено в рамках ГПНИ «Конвергенция-2025».

Для цитирования. Демиденко, В. М. Векторизация итерационных вычислительных процессов и оценки временного ускорения / В. М. Демиденко, В. И. Бенедиктович // Информатика. – 2022. − Т. 19, № 1. – С. 72–87. https://doi.org/10.37661/1816-0301-2022-19-1-72-87

Конфликт интересов. Авторы заявляют об отсутствии конфликта интересов.

Поступила в редакцию | Received 31.12.2021
Подписана в печать | Accepted 28.02.2022
Опубликована | Published 29.03.2022

A vectorization of iterative computational processes and time acceleration estimates

Vitaly M. Demidenko[1]✉, Vladimir I. Benediktovich[2]

[1]Belarusian State Economic University,
av. Partizansky, 26, Minsk, 220070, Belarus,
✉E-mail: vmdemidenko@yandex.ru

[2]Institute of Mathematics of the National Academy of Sciences of Belarus,
st. Surganova, 11, Minsk, 220072, Belarus
E-mail: benedvi@gmail.com

Abstract

Objectives. The problem of efficient organization of the execution of sequential computational processes in the vector mode is solved, taking into account the capabilities of modern high-performance vector-pipeline computers. The relevance of the problem under consideration is due to the fact that processes that occur during cyclic data processing and in iterative algorithms are the most difficult to parallelize. In solving the problem, three main objectives were set. Construction of a mathematical model that takes into account the main architectural and computational features of modern vector-conveyor computers. Calculation of the optimal total execution time of vector operations. Evaluation of the time gain compared to the sequential mode of data processing.

Methods. To achieve the objectives and to prove the main and auxiliary statements, an original method was used, including establishing the validity of the inductive assumptions in the cases under consideration, as well as an illustrative method of scheduling theory using Gantt charts.

Results. A vector model for the implementation of sequential calculations is proposed, which takes into account the main features of vector-conveyor computers. The optimal total execution time of sequential calculations in the vector mode is determined, and a lower estimate of the time gain is obtained in comparison with the sequential mode of their execution.

Conclusion. It has been established that when processing scalar input data in the sequential mode by vector operations with a pipeline length k, acceleration is possible by at least a factor $nN/(nk+N)$, where N is the size of the input, n is the number of vector and corresponding scalar operations. The estimation of the time acceleration in the vectorization of calculations is compared with the sequential mode of their execution.

Keywords: pipelining, vectoring of calculations, vectorizability index, temporal acceleration, vector-pipeline computing systems, calculating processes paralleling

Acknowledgements. The research was carried out within the framework of the state scientific research program "Convergence-2025".

For citation. Demidenko V. M., Benediktovich V. I. *A vectorization of iterative computational processes and time acceleration estimates*. Informatika [*Informatics*], 2022, vol. 19, no. 1, pp. 72−87 (In Russ.).
https://doi.org/10.37661/1816-0301-2022-19-1-72-87

Conflict of interest. The authors declare of no conflict of interest.

Введение. В настоящее время широкое применение высокопроизводительных вычислительных систем, обладающих средствами векторно-конвейерной и параллельной обработки данных, приводит к необходимости адаптации имеющихся и разработки новых методов и алгоритмов

решения научных и практических задач, учитывающих возможности и архитектурные особенности таких систем. Решение обозначенной проблемы возможно либо с помощью имеющихся высокоэффективных вычислительных систем, либо с помощью современных технологий организации вычислительных процессов с учетом архитектурных особенностей ЭВМ.

С момента появления современных вычислительных средств происходит постоянный процесс их совершенствования. При этом одна из тенденций в развитии вычислительного дела неразрывно связана с созданием высокопроизводительных систем векторно-конвейерного типа. Например, суперкомпьютер Cray Titan компании Cray, установленный в Ок-Риджской национальной лаборатории (США), в настоящее время является первым среди векторно-конвейерных вычислительных систем, потеснив экс-лидера Sequoia с пиковой производительностью 16,32 петафлоп/с, который занимал 12-е место среди самых быстрых суперкомпьютеров мира. Пиковая производительность Cray Titan теоретически составляет 27,11 петафлоп/с. Доступ к Cray Titan получают проекты, связанные с процессами сгорания топлива, разработкой передовых технологий в атомной энергетике, новых материалов и исследованиями в области изменения климата.

Несмотря на наличие высокопроизводительных вычислительных систем, их практическое применение широким кругом пользователей затруднено ввиду значительной стоимости и отсутствия доступа к «облачным» вычислениям в Интернете. Поэтому наиболее приемлемым способом повышения эффективности программных и алгоритмических средств является их адаптация к архитектуре векторно-конвейерных ЭВМ.

Известно, что вычислительные процессы, связанные с выполнением операций в последовательном режиме, трудно реализовывать на многопроцессорных системах. Например, такая ситуация возникает при использовании итеративных алгоритмов, в которых на предшествующих шагах вырабатываются входные данные для последующих шагов. В настоящей работе исследуется один из основных способов эффективной организации последовательных вычислительных процессов в векторном режиме с применением макроопераций, позволяющих обрабатывать упорядоченные наборы данных фиксированной длины. По этому направлению зарубежными и отечественными исследователями получены многочисленные результаты [1–5]. В 1980–1990-х гг. в Институте математики НАН Беларуси проводились исследования, связанные с разработкой программно-алгоритмического обеспечения для векторно-конвейерной ЭВМ «Электроника СС БИС», архитектурно сходной с американским аналогом CREY-1. В частности, для введенного в работе [6] класса комбинаторных алгоритмов так называемых неветвящихся вычислительных процессов посредством реконструкции описывающих их графов удалось получить оптимальное расписание их реализации. В статье [7] проиллюстрирована возможность значительного сокращения трудоемкости отдельных комбинаторных алгоритмов не только за счет сегментации скалярных операций, но и за счет применения их векторных аналогов. В работе [8] посредством детального исследования алгоритмов цифровой и лексикографической сортировок реализовано их эффективное отображение на архитектуру векторно-конвейерной ЭВМ «Электроника СС БИС», что позволило получить программные реализации алгоритмов, не улучшаемые имеющимися трансляторами.

Кроме того, в настоящей статье подсчитано суммарное время реализации вычислительного процесса в векторно-конвейерном режиме и получена оценка возможного ускорения в сравнении с применением только скалярных операций.

Постановка задачи и основные определения. Рассмотрим последовательный вычислительный процесс, который состоит в выполнении n операций в заданном порядке над множеством операндов $\Theta = \{\theta_1, \theta_2, \ldots, \theta_\ell, \ldots, \theta_N\}$ при следующих ограничениях:

1) каждая последующая операция начинается после обработки предшествующими операциями всех операндов из Θ;

2) каждая операция осуществляется только над одним операндом и каждый операнд обрабатывается только одной операцией в порядке их нумерации.

Реализуем данный вычислительный процесс в скалярном и векторном режимах и оценим возможный временной выигрыш второго режима по сравнению с первым.

Скалярная модель рассматриваемого вычислительного процесса состоит в следующем. Имеется n скалярных операций $\Omega=\{\omega_1,\omega_2,\ldots,\omega_i,\ldots,\omega_n\}$, которые нужно выполнить в последовательном режиме над N операндами из Θ. Далее операнды из Θ называются скалярными. Время выполнения операции ω_i над произвольным операндом θ_j из Θ, измеряемое в тактах, определено и равно t_{ij}, $i=1, 2, \ldots, n$, $j=1, 2, \ldots, N$ (такт вычислительной системы – временная единица порядка 10^{-6} – 10^{-9} с).

Пусть $T_s(\omega_i,\Theta)$ – время обработки всех операндов из множества Θ операцией ω_i, $T_s(\Omega,\Theta)$ – суммарное время выполнения всех операций из Ω над операндами из Θ. Тогда, очевидно (рис. 1),

$$T_s(\omega_i,\Theta)=\sum_{j=1}^{N} t_{ij}$$

и, следовательно,

$$T_s(\Omega,\Theta)=\sum_{i=1}^{n} T_s(\omega_i,\Theta)=\sum_{i=1}^{n}\sum_{j=1}^{N} t_{ij}\ . \qquad (1)$$

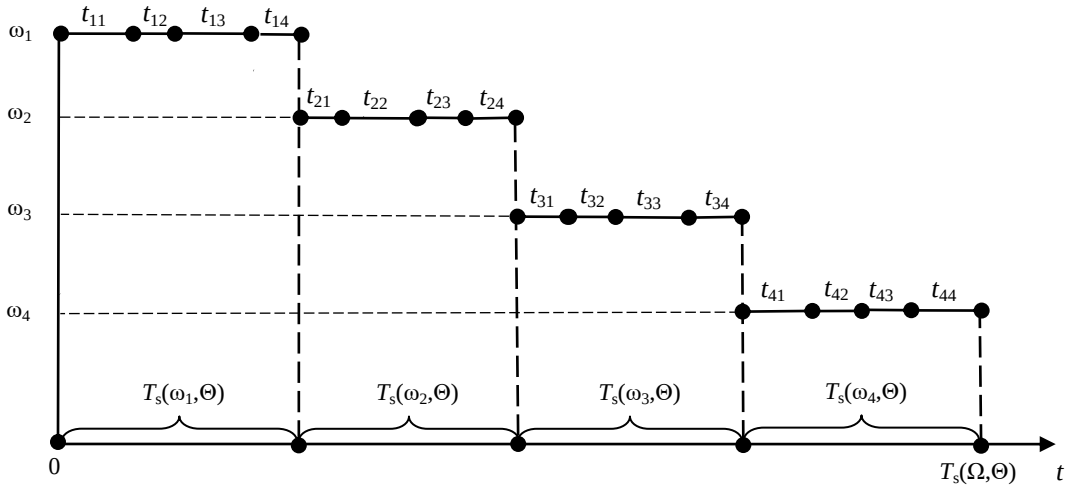


Рис. 1. Временна́я диаграмма скалярной модели при $n = 4$, $N = 4$

Fig. 1. Timing diagram of the scalar model at n = 4, N = 4

В современных векторно-конвейерных ЭВМ наряду со скалярными операциями используются их векторные аналоги – так называемые векторные операции. Специфика последних состоит в том, что они могут в последовательном режиме обрабатывать упорядоченные наборы операндов заданной длины, которые далее называются векторными операндами. Каждая векторная операция может обрабатывать только один векторный операнд, и каждый из них может обрабатываться только одной векторной операцией. Выполнение различных векторных операций над разными векторными операндами может происходить одновременно при их доступности, т. е. в параллельном режиме. Началу обработки операндов векторной операцией предшествует временной интервал ее настройки на обработку операндов, при этом настройка каждой последующей операции начинается в момент начала выполнения предшествующей.

Векторная модель рассматриваемого вычислительного процесса состоит в следующем. Вместо множества скалярных операций $\Omega=\{\omega_1,\omega_2,\ldots,\omega_i,\ldots,\omega_n\}$ используются их векторные аналоги, т. е. множество векторных операций $\bar{\Omega}=\{\bar{\omega}_1,\bar{\omega}_2,\ldots,\bar{\omega}_i,\ldots,\bar{\omega}_n\}$, которые могут выполняться над векторными операндами длиной k в заданном порядке. Множество таких векторных операндов $\bar{\Theta}=\{\bar{\theta}_1,\bar{\theta}_2,\ldots,\bar{\theta}_\ell,\ldots,\bar{\theta}_m\}$ строится посредством разбиения множества скалярных

операндов $\Theta=\{\theta_1,\theta_2,\ldots,\theta_j,\ldots,\theta_N\}$ на $m=N/k$ подмножеств $\bar{\theta}_\ell=\{\theta_{\ell 1},\theta_{\ell 2},\ldots,\theta_{\ell p},\ldots,\theta_{\ell k}\}$, где $\theta_j=\theta_{\ell p}$, $p=j-[j/k]$, $\ell=1, 2, \ldots, m$, $j=1, 2, \ldots, N$. Будем предполагать, что N кратно k, так как в противном случае увеличение числа элементов в Θ на $m(k+1)-N$ операндов при оценке временно́го ускорения приводит к увеличению суммарного времени выполнения векторных операций над векторными операндами по отношению к $T_s(\Omega, \Theta)$.

Любая операция $\bar{\omega}_i$ над операндом $\bar{\theta}_\ell$ в векторном режиме состоит в обработке каждой его компоненты $\theta_{\ell p}$ в последовательном режиме за тактовое время t. Таким образом, можно считать, что верно равенство

$$t_{\bar{\theta}_\ell}=kt \tag{2}$$

для любых $\ell=1, 2, \ldots, m$, где $t_{\bar{\theta}_\ell}$ – время выполнения $\bar{\omega}_i$ над $\bar{\theta}_\ell$.

В реальных векторно-конвейерных ЭВМ операция $\bar{\omega}_i$ над компонентой векторного операнда реализуется за время, которое равно одному машинному такту и значительно меньше времени t_{ij} выполнения скалярной операции $\omega_i\in\Omega$ над скалярным операндом $\theta_j\in\Theta$. Следовательно, будем предполагать, что в векторной модели вычислений $t\ll t_{ij}$, $i=1, 2, \ldots, n$, $j=1, 2, \ldots, N$.

При формулировке вспомогательных и основных утверждений и при их доказательстве используются следующие обозначения:

$\bar{\Omega}_i=\{\bar{\omega}_1,\bar{\omega}_2,\ldots,\bar{\omega}_p,\ldots,\bar{\omega}_i\}$ – подмножество первых i векторных операций из $\bar{\Omega}$, $1\le i\le n$;

$\bar{\Theta}_\ell=\{\bar{\theta}_1,\bar{\theta}_2,\ldots,\bar{\theta}_q,\ldots,\bar{\theta}_\ell\}$ – подмножество первых ℓ векторных операндов из $\bar{\Theta}$, $1\le\ell\le m$;

$T_v(\bar{\omega}_i,\bar{\theta}_\ell)$ – время выполнения операции $\bar{\omega}_i$ над операндом $\bar{\theta}_\ell$, $i=1, 2, \ldots, n$, $\ell=1, 2, \ldots, m$, при условии выполнения всех предшествующих операций $\bar{\omega}_p$, $p=1, 2, \ldots, i-1$;

$T_v(\bar{\Omega}_i,\bar{\theta}_\ell)$ – время выполнения всех операций из $\bar{\Omega}_i$ над операндом $\bar{\theta}_\ell$;

$T_v(\bar{\Omega}_i,\bar{\Theta}_\ell)$ – время выполнения всех операций из $\bar{\Omega}_i$ над операндами из $\bar{\Theta}_\ell$;

$T_v(\bar{\Omega},\bar{\Theta})$ – время выполнения всех операций из $\bar{\Omega}$ над операндами из $\bar{\Theta}$;

$t_\tau(\bar{\omega}_i)$ – время завершения настройки операции $\bar{\omega}_i$, $i=1, 2, \ldots, n$, при условии выполнения всех предшествующих операций $\bar{\omega}_p$, $p=1, 2, \ldots, i-1$;

$t_0(\bar{\omega}_i)$ – время начала операции $\bar{\omega}_i, i=1, 2, \ldots, n$, при условии осуществления всех предшествующих операций $\bar{\omega}_p$, $p=1, 2, \ldots, i-1$.

Введенные временные интервалы далее называются векторным временем выполнения соответствующих векторных операций над операндами.

Непосредственно из определения $t_\tau(\bar{\omega}_i)$, $t_0(\bar{\omega}_i)$ следует справедливость равенств

$$t_\tau(\bar{\omega}_i)=t_0(\bar{\omega}_{i-1})+\tau_i,\quad t_0(\bar{\omega}_i)=T_v(\bar{\omega}_i,\bar{\theta}_1)-t_{\bar{\theta}_1},\quad i=2,\ldots,n. \tag{3}$$

Векторная обработка данных. Для получения нижней оценки коэффициента ускорения при использовании векторной модели вычислений в сравнении со скалярной моделью применяются три вспомогательных и одно основное утверждение, которое определяет формулу для вычисления векторного времени выполнения всех операций из $\bar{\Omega}$ над всеми операндами из $\bar{\Theta}$:

$$T_v(\bar{\Omega},\bar{\Theta})=\sum_{p=2}^{n}\max\{\tau_p,t_{\bar{\theta}_1}\}+T_v(\bar{\omega}_1,\bar{\Theta}). \tag{4}$$

В первых двух вспомогательных утверждениях устанавливается время выполнения всех операций из $\bar{\Omega}_i$ над всеми операндами $\bar{\Theta}_\ell$ для частных случаев, когда $i \in \{1, 2, 3\}$, $1 \le \ell \le m$ и $2 \le i \le n$, $\ell = 2$ соответственно. Третье утверждение расширяет аналогичный результат на общий случай, когда $2 \le i \le n$ и $2 \le \ell \le m$.

Лемма 1. Для $i \in \{1, 2, 3\}$ и любых $1 \le \ell \le m$ справедливы равенства

$$T_v\left(\bar{\omega}_1, \bar{\Theta}_\ell\right) = \tau_1 + \sum_{r=1}^{\ell} t_{\bar{\theta}_\ell}, \tag{5}$$

$$T_v\left(\bar{\Omega}_2, \bar{\Theta}_\ell\right) = T_v\left(\bar{\omega}_2, \bar{\Theta}_\ell\right) = \max\left\{\tau_2, t_{\bar{\theta}_1}\right\} + T_v\left(\bar{\omega}_1, \bar{\Theta}_\ell\right), \tag{6}$$

$$T_v\left(\bar{\Omega}_3, \bar{\Theta}_\ell\right) = T_v\left(\bar{\omega}_3, \bar{\Theta}_\ell\right) = \max\left\{\tau_2, t_{\bar{\theta}_1}\right\} + \max\left\{\tau_3, t_{\bar{\theta}_1}\right\} + T_v\left(\bar{\omega}_1, \bar{\Theta}_\ell\right). \tag{7}$$

Доказательство. При $i = 1$ и любом $1 \le \ell \le m$ справедливость равенства (5) непосредственно следует из равенства (2) и последовательного выполнения операции $\bar{\omega}_1$ над операндами $\bar{\Theta}_\ell$.

Пусть $i = 2$ и $\ell = 1$, тогда возможны два случая: $\tau_2 \ge t_{\bar{\theta}_1}$ либо $\tau_2 < t_{\bar{\theta}_1}$. В обоих случаях операция $\bar{\omega}_2$ над $\bar{\theta}_1$ должна начаться после настройки операции $\bar{\omega}_2$ и выполнения $\bar{\omega}_1$ над $\bar{\theta}_1$. В первом случае в силу $\tau_2 \ge t_{\bar{\theta}_1}$ и (3) имеют место соотношения

$$t_\tau\left(\bar{\omega}_2\right) = \tau_1 + \tau_2 \ge \tau_1 + t_{\bar{\theta}_1} = T_v\left(\bar{\omega}_1, \bar{\theta}_1\right), \tag{8}$$

из которых (рис. 2) следуют равенства

$$T_v\left(\bar{\omega}_2, \bar{\theta}_1\right) = T_v\left(\bar{\Omega}_2, \bar{\theta}_1\right) = t_\tau\left(\bar{\omega}_2\right) + t_{\bar{\theta}_1} = \tau_1 + \tau_2 + t_{\bar{\theta}_1} = \max\{\tau_2, t_{\bar{\theta}_1}\} + T_v\left(\bar{\omega}_1, \bar{\theta}_1\right).$$

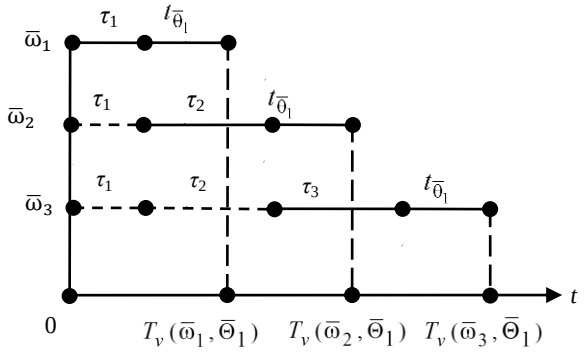


Рис. 2. Временна́я диаграмма векторной модели при $i \in \{1,2,3\}$, $\ell = 1$, $\tau_2 \ge t_{\theta_1}$, $\tau_3 \ge t_{\theta_1}$

Fig. 2. Timing diagram of the vector model at $i \in \{1,2,3\}$, $\ell = 1$, $\tau_2 \ge t_{\theta_1}$, $\tau_3 \ge t_{\theta_1}$

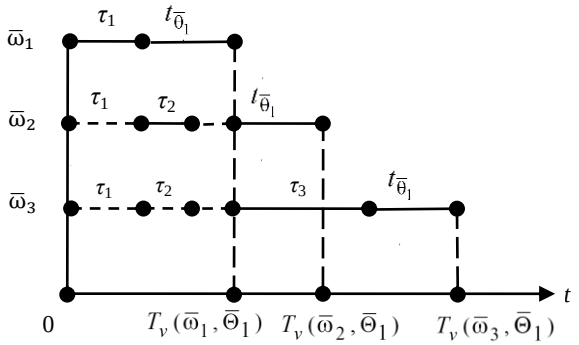


Рис. 3. Временна́я диаграмма векторной модели при $i \in \{1,2,3\}$, $\ell = 1$, $\tau_2 < t_{\theta_1}$, $\tau_3 \ge t_{\theta_1}$

Fig. 3. Timing diagram of the vector model at $i \in \{1,2,3\}$, $\ell = 1$, $\tau_2 < t_{\theta_1}$, $\tau_3 \ge t_{\theta_1}$

Во втором случае в силу $\tau_2 < t_{\bar{\theta}_1}$ выполняется неравенство, противоположное соотношению (8). Следовательно, $t_0\left(\bar{\omega}_2\right) = T_v\left(\bar{\omega}_1, \bar{\theta}_1\right)$, что влечет (рис. 3) справедливость равенств

$$T_v\left(\bar{\omega}_2, \bar{\theta}_1\right) = T_v\left(\bar{\Omega}_2, \bar{\theta}_1\right) = t_0(\bar{\omega}_2) + t_{\bar{\theta}_1} = T_v\left(\bar{\omega}_1, \bar{\theta}_1\right) + t_{\bar{\theta}_1} = \max\{\tau_2, t_{\bar{\theta}_1}\} + T_v\left(\bar{\omega}_1, \bar{\theta}_1\right).$$

Таким образом, при $\ell = 1$ равенство (6) доказано.

Пусть $i=3$ и $\ell=1$, тогда возможен один из четырех случаев:

$$\tau_2 \ge t_{\bar{\theta}_1},\ \tau_3 \ge t_{\bar{\theta}_1};\ \ \tau_2 < t_{\bar{\theta}_1},\ \tau_3 \ge t_{\bar{\theta}_1};\ \ \tau_2 \ge t_{\bar{\theta}_1},\ \tau_3 < t_{\bar{\theta}_1};\ \ \tau_2 < t_{\bar{\theta}_1},\ \tau_3 < t_{\bar{\theta}_1}. \qquad (9)$$

Во всех четырех случаях операция $\bar{\omega}_3$ над $\bar{\theta}_1$ должна начаться после ее настройки и после выполнения $\bar{\omega}_2$ над $\bar{\theta}_1$. Поэтому переменная $t_0(\bar{\omega}_3)$ должна принимать наибольшее значение из $t_\tau(\bar{\omega}_3)$ и $T_v\left(\bar{\omega}_2,\bar{\theta}_1\right)$.

В первом случае из неравенств $\tau_2 \ge t_{\bar{\theta}_1}$, $\tau_3 \ge t_{\bar{\theta}_1}$, равенства 3 и доказанного при $\ell=1$ равенства (6) следует справедливость равенств

$$t_0(\bar{\omega}_2) = T_v\left(\bar{\omega}_2,\bar{\theta}_1\right) - t_{\bar{\theta}_1} = \tau_2 + \tau_1 + t_{\bar{\theta}_1} - t_{\bar{\theta}_1} = \tau_2 + \tau_1,$$

$$t_\tau(\bar{\omega}_3) = t_0(\bar{\omega}_2) + \tau_3 = \tau_1 + \tau_2 + \tau_3,$$

из которых вытекают соотношения

$$T_v\left(\bar{\omega}_2,\bar{\theta}_1\right) = \tau_1 + \tau_2 + t_{\bar{\theta}_1} \le \tau_1 + \tau_2 + \tau_3 = t_\tau(\bar{\omega}_3).$$

Таким образом, доказано, что $t_0(\bar{\omega}_3) = \max\left\{t_\tau(\bar{\omega}_3), T_v\left(\bar{\omega}_2,\bar{\theta}_1\right) = t_\tau(\bar{\omega}_3)\right\}$, откуда с учетом первого равенства (3) следуют равенства (см. рис. 2)

$$T_v\left(\bar{\omega}_3,\bar{\theta}_1\right) = T_v\left(\bar{\Omega}_3,\bar{\theta}_1\right) = t_0(\bar{\omega}_3) + t_{\bar{\theta}_1} = \tau_1 + \tau_2 + \tau_3 + t_{\bar{\theta}_1} = \max\{\tau_2, t_{\bar{\theta}_1}\} + \max\{\tau_3, t_{\bar{\theta}_1}\} + T_v\left(\bar{\omega}_1,\bar{\theta}_1\right),$$

доказывающие справедливость равенства (7) в рассматриваемом случае.

Во втором случае после выполнения равенства (6) при $\ell=1$, неравенств $\tau_2 < t_{\bar{\theta}_1}$, $\tau_3 \ge t_{\bar{\theta}_1}$ и с учетом (3) получаем равенства

$$t_0(\bar{\omega}_2) = T_v\left(\bar{\omega}_2,\bar{\theta}_1\right) - t_{\bar{\theta}_1} = t_{\bar{\theta}_1} + \tau_1 + t_{\bar{\theta}_1} - t_{\bar{\theta}_1} = t_{\bar{\theta}_1} + \tau_1,$$

$$t_\tau(\bar{\omega}_3) = t_0(\bar{\omega}_2) + \tau_3 = t_{\bar{\theta}_1} + \tau_1 + \tau_3.$$

Полученные равенства устанавливают справедливость соотношений (рис. 4)

$$T_v\left(\bar{\Omega}_3,\bar{\Theta}_1\right) = T_v\left(\bar{\Omega}_3,\bar{\theta}_1\right) = t_0(\bar{\omega}_3) + t_{\bar{\theta}_1} = t_{\bar{\theta}_1} + \tau_1 + \tau_3 + t_{\bar{\theta}_1} = \max\left\{\tau_2, t_{\bar{\theta}_1}\right\} + \max\left\{\tau_3, t_{\bar{\theta}_1}\right\} + T_v\left(\bar{\omega}_1,\bar{\theta}_1\right),$$

которые, в свою очередь, доказывают выполнение (7) при $\ell=1$.

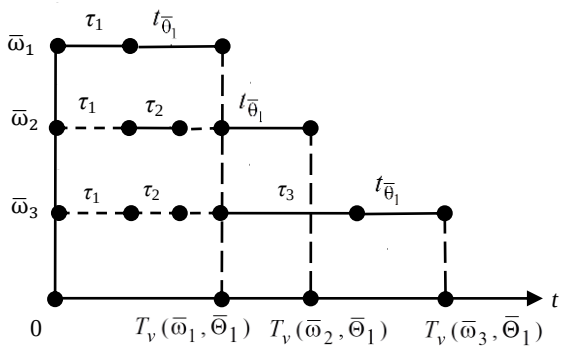


Рис. 4. Временна́я диаграмма векторной модели при $i \in \{1,2,3\}$, $\ell=1$, $\tau_2 < t_{\theta_1}$, $\tau_3 \ge t_{\theta_1}$

Fig. 4. Timing diagram of the vector model at $i \in \{1,2,3\}$, $\ell=1$, $\tau_2 < t_{\theta_1}$, $\tau_3 \ge t_{\theta_1}$

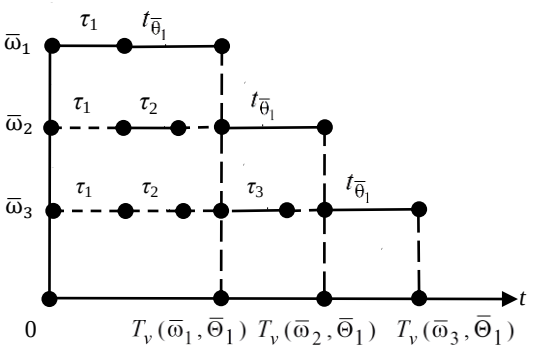


Рис. 5. Временна́я диаграмма векторной модели при $i \in \{1,2,3\}$, $\ell=1$, $\tau_2 < t_{\theta_1}$, $\tau_3 < t_{\theta_1}$

Fig. 5. Timing diagram of the vector model at $i \in \{1,2,3\}$, $\ell=1$, $\tau_2 < t_{\theta_1}$, $\tau_3 < t_{\theta_1}$

В третьем случае, используя неравенства $\tau_2 \geq t_{\bar{\theta}_1}$, $\tau_3 < t_{\bar{\theta}_1}$, равенство (6) при $\ell = 1$ и равенства (3), по аналогии убеждаемся в выполнении равенств

$$t_0(\bar{\omega}_2) = T_v\left(\bar{\omega}_2, \bar{\theta}_1\right) - t_{\bar{\theta}_1} = \tau_2 + \tau_1 + t_{\bar{\theta}_1} - t_{\bar{\theta}_1} = \tau_1 + \tau_2,$$

$$t_\tau(\bar{\omega}_3) = t_0(\bar{\omega}_2) + \tau_3 = \tau_1 + \tau_2 + \tau_3,$$

откуда следует справедливость

$$T_v\left(\bar{\omega}_2, \bar{\theta}_1\right) = \tau_2 + \tau_1 + t_{\bar{\theta}_1} > \tau_1 + \tau_2 + \tau_3 = t_\tau(\bar{\omega}_3),$$

$$t_0(\bar{\omega}_3) = \max\left\{t_\tau(\bar{\omega}_3), T_v\left(\bar{\omega}_2, \bar{\theta}_1\right)\right\} = T_v\left(\bar{\omega}_2, \bar{\theta}_1\right).$$

Из второго соотношения и доказанного при $\ell = 1$ равенства (6) получаем цепочку

$$T_v\left(\bar{\omega}_3, \bar{\theta}_1\right) = T_v\left(\bar{\Omega}_3, \bar{\theta}_1\right) = t_0(\bar{\omega}_3) + t_{\bar{\theta}_1} = T_v\left(\bar{\omega}_2, \bar{\theta}_1\right) + t_{\bar{\theta}} = \max\{\tau_2, t_{\bar{\theta}_1}\} + \max\{\tau_3, t_{\bar{\theta}_1}\} + T_v\left(\bar{\omega}_1, \bar{\theta}_1\right),$$

из которой (см. рис. 4) следует справедливость равенства (7) при $\ell = 1$ в третьем случае.

В четвертом случае по аналогии убеждаемся в том, что из неравенств $\tau_2 < t_{\bar{\theta}_1}$, $\tau_3 < t_{\bar{\theta}_1}$, равенства (6) при $\ell = 1$ и равенств (3) следует справедливость соотношений

$$t_0(\bar{\omega}_2) = T_v\left(\bar{\omega}_2, \bar{\theta}_1\right) - t_{\bar{\theta}_1} = t_{\bar{\theta}_1} + \tau_1 + t_{\bar{\theta}_1} - t_{\bar{\theta}_1} = t_{\bar{\theta}_1} + \tau_1,$$

$$t_\tau(\bar{\omega}_3) = t_0(\bar{\omega}_2) + \tau_3 = t_{\bar{\theta}_1} + \tau_1 + \tau_3,$$

откуда вытекают равенства

$$T_v\left(\bar{\omega}_2, \bar{\theta}_1\right) = t_{\bar{\theta}_1} + \tau_1 + t_{\bar{\theta}_1} > t_{\bar{\theta}_1} + \tau_1 + \tau_3 = t_\tau(\bar{\omega}_3),$$

$$t_0(\bar{\omega}_3) = \max\left\{t_\tau(\bar{\omega}_3),\ T_v\left(\bar{\omega}_2, \bar{\theta}_1\right)\right\} = T_v\left(\bar{\omega}_2, \bar{\theta}_1\right).$$

Из второго равенства и доказанного при $\ell = 1$ равенства (6) получаем соотношения

$$T_v\left(\bar{\omega}_3, \bar{\theta}_1\right) = T_v\left(\bar{\Omega}_3, \bar{\theta}_1\right) = t_0(\bar{\omega}_3) + t_{\bar{\theta}_1} = T_v\left(\bar{\omega}_2, \bar{\theta}_1\right) + t_{\bar{\theta}_1} = \max\{\tau_2, t_{\bar{\theta}_1}\} + \max\{\tau_3, t_{\bar{\theta}_1}\} + T_v\left(\bar{\omega}_1, \bar{\theta}_1\right),$$

из которых (см. рис. 5) следует справедливость равенства (7) при $\ell = 1$ в четвертом случае.

Так как во всех четырех случаях $\bar{\Theta}_1 = \left\{\bar{\theta}_1\right\}$, справедливость леммы 1 доказана при $i \in \{1, 2, 3\}$ и $\ell = 1$.

Выполнение равенств (6) и (7) для $i = 2, 3$ и любых $2 \leq \ell \leq m$ доказывается индукцией по ℓ. Предполагая, что для любых $1 \leq \ell - 1 \leq m - 1$ выполняются равенства

$$T_v\left(\bar{\Omega}_2, \bar{\Theta}_{\ell-1}\right) = T_v\left(\bar{\omega}_2, \bar{\Theta}_{\ell-1}\right) = \max\left\{\tau_2, t_{\bar{\theta}_1}\right\} + T_v\left(\bar{\omega}_1, \bar{\Theta}_{\ell-1}\right), \tag{10}$$

$$T_v\left(\bar{\Omega}_3, \bar{\Theta}_{\ell-1}\right) = T_v\left(\bar{\omega}_3, \bar{\Theta}_{\ell-1}\right) = \max\left\{\tau_2, t_{\bar{\theta}_1}\right\} + \max\left\{\tau_3, t_{\bar{\theta}_1}\right\} + T_v\left(\bar{\omega}_1, \bar{\Theta}_{\ell-1}\right), \tag{11}$$

убедимся в справедливости для $2 \leq \ell \leq m$ равенств

$$T_v\left(\bar{\Omega}_2, \bar{\Theta}_\ell\right) = \max\left\{\tau_2, t_{\bar{\theta}_1}\right\} + T_v\left(\bar{\omega}_1, \bar{\Theta}_\ell\right), \tag{12}$$

$$T_v\left(\bar{\Omega}_3, \bar{\Theta}_\ell\right) = \max\left\{\tau_2, t_{\bar{\theta}_1}\right\} + \max\left\{\tau_3, t_{\bar{\theta}_1}\right\} + T_v\left(\bar{\omega}_1, \bar{\Theta}_\ell\right). \tag{13}$$

Докажем равенство (12). Возможны два случая: $\tau_2 \geq t_{\bar{\theta}_1}$ либо $\tau_2 < t_{\bar{\theta}_1}$. В первом случае в силу (5), (10) и равенства $t_{\bar{\theta}_\ell} = t_{\bar{\theta}_1}$ получаем равенства

$$T_v\left(\bar{\omega}_1, \bar{\Theta}_\ell\right) = T_v\left(\bar{\omega}_1, \bar{\Theta}_{\ell-1}\right) + t_{\bar{\theta}_1}, \quad T_v\left(\bar{\omega}_2, \bar{\Theta}_{\ell-1}\right) = \tau_2 + T_v\left(\bar{\omega}_1, \bar{\Theta}_{\ell-1}\right), \tag{14}$$

из которых и неравенства $\tau_2 \geq t_{\bar{\theta}_1}$ вытекает неравенство $T_v\left(\bar{\omega}_2, \bar{\Theta}_{\ell-1}\right) \geq T_v\left(\bar{\omega}_1, \bar{\Theta}_\ell\right)$. Его справедливость означает, что операция $\bar{\omega}_1$ над операндом $\bar{\theta}_\ell$ заканчивается до момента времени $T_v\left(\bar{\omega}_2, \bar{\Theta}_{\ell-1}\right)$. Поэтому с учетом (10) имеют место равенства

$$T_v\left(\bar{\Omega}_2, \bar{\Theta}_\ell\right) = T_v\left(\bar{\omega}_2, \bar{\Theta}_{\ell-1}\right) + t_{\theta_\ell} = \max\left\{\tau_2, t_{\bar{\theta}_1}\right\} + T_v\left(\bar{\omega}_1, \bar{\Theta}_{\ell-1}\right) + t_{\theta_\ell} = \max\left\{\tau_2, t_{\bar{\theta}_1}\right\} + T_v\left(\bar{\omega}_1, \bar{\Theta}_\ell\right),$$

из которых следует справедливость равенства (12) в первом случае.

Во втором случае из неравенства $\tau_2 < t_{\bar{\theta}_1}$ и равенств (14) получаем соотношение $T_v\left(\bar{\omega}_2, \bar{\Theta}_{\ell-1}\right) < T_v\left(\bar{\omega}_1, \bar{\Theta}_\ell\right)$, которое означает, что операция $\bar{\omega}_2$ над операндом $\bar{\theta}_\ell$ может начаться не ранее момента времени $T_v\left(\bar{\omega}_1, \bar{\Theta}_\ell\right)$ и, следовательно,

$$T_v\left(\bar{\Omega}_2, \bar{\Theta}_\ell\right) = T_v\left(\bar{\omega}_1, \bar{\Theta}_\ell\right) + t_{\theta_\ell} = \max\left\{\tau_2, t_{\bar{\theta}_1}\right\} + T_v\left(\bar{\omega}_1, \bar{\Theta}_\ell\right).$$

Таким образом, равенство (12) доказано для $i = 2$ и любых $2 \leq \ell \leq m$.

Убедимся в справедливости равенства (13). Для этого вначале докажем, что при выполнении первой и второй пар неравенств (9) имеет место соотношение

$$T_v\left(\bar{\omega}_3, \bar{\Theta}_{\ell-1}\right) \geq T_v\left(\bar{\omega}_2, \bar{\Theta}_\ell\right). \tag{15}$$

Действительно, пусть $\tau_2 \geq t_{\bar{\theta}_1}$ и $\tau_3 \geq t_{\bar{\theta}_1}$, тогда в силу равенств (5), (11) и (12) справедливо выражение

$$T_v\left(\bar{\omega}_3, \bar{\Theta}_{\ell-1}\right) - T_v\left(\bar{\omega}_2, \bar{\Theta}_\ell\right) = \tau_2 + \tau_3 + T_v\left(\bar{\omega}_1, \bar{\Theta}_{\ell-1}\right) - \tau_2 - T_v\left(\bar{\omega}_1, \bar{\Theta}_{\ell-1}\right) - t_{\bar{\theta}_\ell} = \tau_3 - t_{\bar{\theta}_\ell} \geq 0,$$

так как $t_{\bar{\theta}_1} = t_{\bar{\theta}_\ell}$. Если $\tau_2 < t_{\bar{\theta}_1}$ и $\tau_3 \geq t_{\bar{\theta}_1}$, то по аналогии убеждаемся в следующем:

$$T_v\left(\bar{\omega}_3, \bar{\Theta}_{\ell-1}\right) - T_v\left(\bar{\omega}_2, \bar{\Theta}_\ell\right) = t_{\bar{\theta}_1} + \tau_3 + T_v\left(\bar{\omega}_1, \bar{\Theta}_{\ell-1}\right) - t_{\bar{\theta}_1} - T_v\left(\bar{\omega}_1, \bar{\Theta}_{\ell-1}\right) - t_{\bar{\theta}_\ell} = \tau_3 - t_{\bar{\theta}_\ell} \geq 0.$$

Таким образом, справедливость неравенства (15) в рассмотренных случаях доказана.

Далее, неравенство (15) гарантирует, что к моменту времени $T_v\left(\bar{\omega}_3, \bar{\Theta}_{\ell-1}\right)$ операция $\bar{\omega}_2$ над $\bar{\theta}_\ell$ завершена и поэтому $T_v\left(\bar{\omega}_3, \bar{\Theta}_\ell\right) = T_v\left(\bar{\omega}_3, \bar{\Theta}_{\ell-1}\right) + t_{\bar{\theta}_\ell}$. Из полученного соотношения и равенства (11) следуют (рис. 6) равенства

$$\begin{aligned} T_v\left(\bar{\omega}_3, \bar{\Theta}_\ell\right) &= \max\left\{\tau_2, t_{\bar{\theta}_1}\right\} + \max\left\{\tau_3, t_{\bar{\theta}_1}\right\} + T_v\left(\bar{\omega}_1, \bar{\Theta}_{\ell-1}\right) + t_{\bar{\theta}_\ell} = \\ &= \max\left\{\tau_2, t_{\bar{\theta}_1}\right\} + \max\left\{\tau_3, t_{\bar{\theta}_1}\right\} + T_v\left(\bar{\omega}_1, \bar{\Theta}_\ell\right). \end{aligned}$$

Таким образом, доказано, что при выполнении первой и второй пар неравенств (9) имеет место равенство (13).

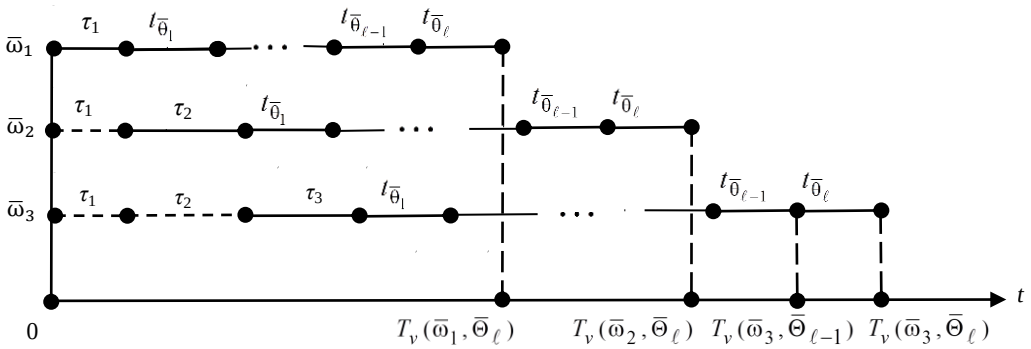


Рис. 6. Временна́я диаграмма векторной модели при $i = 3,\ 2 \le \ell \le m,\ \ \tau_2 \ge t_{\bar{\theta}_1},\ \tau_3 < t_{\bar{\theta}_1}$

Fig. 6. Timing diagram of the vector model at $i = 3,\ 2 \le \ell \le m,\ \ \tau_2 \ge t_{\bar{\theta}_1},\ \tau_3 < t_{\bar{\theta}_1}$

Убедимся в том, что если имеет место третья либо четвертая пара неравенств из (9), то выполняется равенство

$$T_v\left(\bar{\omega}_3, \bar{\Theta}_{\ell-1}\right) = T_v\left(\bar{\omega}_2, \bar{\Theta}_\ell\right). \tag{16}$$

Пусть $\tau_2 \ge t_{\bar{\theta}_1},\ \tau_3 < t_{\bar{\theta}_1}$, тогда следствием (5), (11) и (12) являются равенства

$$T_v\left(\bar{\omega}_3, \bar{\Theta}_{\ell-1}\right) - T_v\left(\bar{\omega}_2, \bar{\Theta}_\ell\right) = \tau_2 + t_{\bar{\theta}_1} + T_v\left(\bar{\omega}_1, \bar{\Theta}_{\ell-1}\right) - \tau_2 - T_v\left(\bar{\omega}_1, \bar{\Theta}_{\ell-1}\right) - t_{\bar{\theta}_\ell} = 0,$$

так как $t_{\bar{\theta}_\ell} = t_{\bar{\theta}_1}$. Если $\tau_2 < t_{\bar{\theta}_1}$ и $\tau_3 < t_{\bar{\theta}_1}$, то по аналогии доказываются равенства

$$T_v\left(\bar{\omega}_3, \bar{\Theta}_{\ell-1}\right) - T_v\left(\bar{\omega}_2, \bar{\Theta}_\ell\right) = t_{\bar{\theta}_1} + t_{\bar{\theta}_1} + T_v\left(\bar{\omega}_1, \bar{\Theta}_{\ell-1}\right) - t_{\bar{\theta}_1} - T_v\left(\bar{\omega}_1, \bar{\Theta}_{\ell-1}\right) - t_{\bar{\theta}_\ell} = 0.$$

Из равенства (16) следует, что при выполнении третьей либо четвертой пары неравенств из (9) к моменту времени $T_v\left(\bar{\omega}_3, \bar{\Theta}_{\ell-1}\right)$ выполнение $\bar{\omega}_2$ над $\bar{\theta}_\ell$ закончено. Следовательно, с учетом равенства (11) (рис. 7) получим выражение

$$T_v\left(\bar{\omega}_3, \bar{\Theta}_\ell\right) = T_v\left(\bar{\omega}_3, \bar{\Theta}_{\ell-1}\right) + t_{\bar{\theta}_\ell} = \max\left\{\tau_2, t_{\bar{\theta}_1}\right\} + \max\left\{\tau_3, t_{\bar{\theta}_1}\right\} + T_v\left(\bar{\omega}_1, \bar{\Theta}_{\ell-1}\right) + t_{\bar{\theta}_\ell} =$$

$$= \max\left\{\tau_2, t_{\bar{\theta}_1}\right\} + \max\left\{\tau_3, t_{\bar{\theta}_1}\right\} + T_v\left(\bar{\omega}_1, \bar{\Theta}_\ell\right),$$

что доказывает справедливость равенства (13) в рассматриваемых случаях.

Таким образом, справедливость леммы 1 доказана.

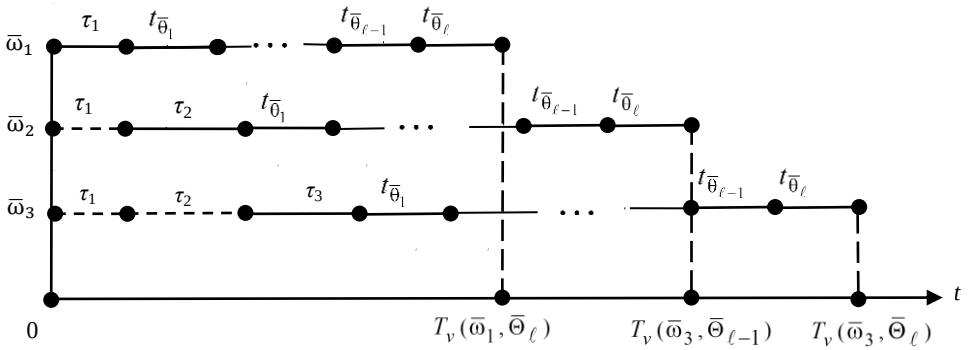


Рис. 7. Временна́я диаграмма векторной модели при $i = 3,\ 2 \le \ell \le m$, $\tau_2 < t_{\theta_1}, \tau_3 < t_{\theta_1}$

Fig. 7. Timing diagram of the vector model at $i = 3,\ 2 \le \ell \le m$, $\tau_2 < t_{\theta_1}, \tau_3 < t_{\theta_1}$

Лемма 2. Если $\ell = 1, 2$, то для любых $1 \le i \le n$ выполняется равенство

$$T_v\left(\bar{\Omega}_i, \bar{\Theta}_\ell\right) = T_v\left(\bar{\omega}_i, \bar{\Theta}_\ell\right) = \sum_{p=2}^{i} \max\left\{\tau_p, t_{\bar{\theta}_1}\right\} + T_v\left(\bar{\omega}_1, \bar{\Theta}_\ell\right). \tag{17}$$

Доказательство. Из леммы 1 следует, что если $\ell = 1, 2$ и $i = 1, 2, 3$, то равенство (17) выполняется. Поэтому доказательство (17) проводится индукцией по *i*.

Предположим справедливость равенства (17) при $\ell = 1, 2$ и $3 \le i-1 \le n-1$ и докажем ее при $\ell = 1, 2$ и $4 \le i \le n$.

При $\ell = 1$ и $i-1$ равенство (17) принимает следующий вид:

$$T_v\left(\bar{\Omega}_{i-1}, \bar{\Theta}_1\right) = T_v\left(\bar{\omega}_{i-1}, \bar{\theta}_1\right) = \sum_{p=2}^{i-1} \max\left\{\tau_p, t_{\bar{\theta}_1}\right\} + T_v\left(\bar{\omega}_1, \bar{\Theta}_1\right). \tag{18}$$

В силу (3) для моментов времени $t_0\left(\bar{\omega}_{i-1}\right)$ и $t_\tau\left(\bar{\omega}_i\right)$ имеют место равенства

$$\begin{aligned} t_0\left(\bar{\omega}_{i-1}\right) &= T_v\left(\bar{\omega}_{i-1}, \bar{\theta}_1\right) - t_{\bar{\theta}_1} = \sum_{p=2}^{i-1} \max\left\{\tau_p, t_{\bar{\theta}_1}\right\} + \tau_1, \\ t_\tau\left(\bar{\omega}_i\right) &= t_0\left(\bar{\omega}_{i-1}\right) + \tau_i = \sum_{p=2}^{i-1} \max\left\{\tau_p, t_{\bar{\theta}_1}\right\} + \tau_1 + \tau_i. \end{aligned} \tag{19}$$

Для τ_i и $t_{\bar{\theta}_1}$ возможны два случая: $\tau_i \ge t_{\bar{\theta}_1}$ либо $\tau_i < t_{\bar{\theta}_1}$. В первом случае в силу $\tau_i \ge t_{\bar{\theta}_1}$ выполняются соотношения

$$t_\tau\left(\bar{\omega}_i\right) = t_0\left(\bar{\omega}_{i-1}\right) + \tau_i \ge t_0\left(\bar{\omega}_{i-1}\right) + t_{\bar{\theta}_1} = T_v\left(\bar{\omega}_{i-1}, \bar{\theta}_1\right),$$

из которых следует, что $t_0\left(\bar{\omega}_i\right) = t_\tau\left(\bar{\omega}_i\right)$, так как к моменту времени $t_\tau\left(\bar{\omega}_i\right)$ закончилась операция $\bar{\omega}_{i-1}$ над операндом $\bar{\theta}_1$ (рис. 8). Следствием равенства $t_0\left(\bar{\omega}_i\right) = t_\tau\left(\bar{\omega}_i\right)$ и второго равенства (19) являются соотношения

$$T_v\left(\bar{\omega}_i, \bar{\theta}_1\right) = t_0\left(\bar{\omega}_i\right) + t_{\bar{\theta}_1} = t_\tau\left(\bar{\omega}_i\right) + t_{\bar{\theta}_1} = \sum_{p=2}^{i} \max\left\{\tau_p, t_{\bar{\theta}_1}\right\} + \tau_1 + t_{\bar{\theta}_1} = \sum_{p=2}^{i} \max\left\{\tau_p, t_{\bar{\theta}_1}\right\} + T_v\left(\bar{\omega}_1, \bar{\theta}_1\right),$$

так как $\tau_i = \max\left\{\tau_i, t_{\bar{\theta}_1}\right\}$. Таким образом, в первом случае при $\ell = 1$ справедливость равенства (17) доказана.

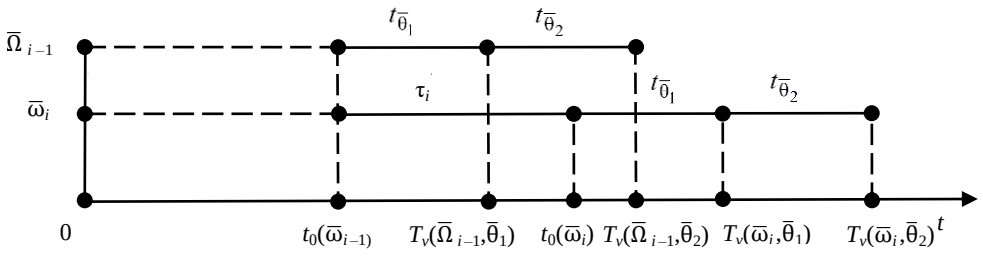


Рис. 8. Временна́я диаграмма при $\tau_i \ge t_{\bar{\theta}_2}$

Fig. 8. Timing diagram at $\tau_i \ge t_{\bar{\theta}_2}$

Пусть выполняется неравенство $\tau_i < t_{\bar{\theta}_1}$. Тогда по аналогии получаем соотношения

$$t_\tau\left(\bar{\omega}_i\right) = t_0\left(\bar{\omega}_{i-1}\right) + \tau_i < t_0\left(\bar{\omega}_{i-1}\right) + t_{\bar{\theta}_1} = T_v\left(\bar{\omega}_{i-1}, \bar{\theta}_1\right),$$

следствием которых является равенство $t_0\left(\bar{\omega}_i\right) = T_v\left(\bar{\omega}_{i-1}, \bar{\theta}_1\right)$. Из данного равенства и выражения (18) вытекает (рис. 9) справедливость соотношений

$$T_v\left(\bar{\omega}_i, \bar{\theta}_1\right) = T_v\left(\bar{\omega}_{i-1}, \bar{\theta}_1\right) + t_{\bar{\theta}_1} = \sum_{p=2}^{i-1} \max\left\{\tau_p, t_{\bar{\theta}_1}\right\} + T_v\left(\bar{\omega}_1, \bar{\theta}_1\right) + t_{\bar{\theta}_1} = \sum_{p=2}^{i} \max\left\{\tau_p, t_{\bar{\theta}_1}\right\} + T_v\left(\bar{\omega}_1, \bar{\theta}_1\right),$$

так как здесь $t_{\bar{\theta}_1} = \max\left\{\tau_i, t_{\bar{\theta}_1}\right\}$. Таким образом, во втором случае справедливость равенства (17) при $\ell = 1$ доказана.

В связи с тем что и в первом, и втором случаях выполняются соотношения

$$\bar{\Theta}_1 = \left\{\bar{\theta}_1\right\}, \quad T_v\left(\bar{\omega}_{i-1}, \bar{\theta}_1\right) = T_v\left(\bar{\Omega}_{i-1}, \bar{\Theta}_1\right) \le T_v\left(\bar{\omega}_i, \bar{\theta}_1\right),$$

очевидно, что $T_v\left(\bar{\Omega}_i, \bar{\Theta}_1\right) = T_v\left(\bar{\omega}_i, \bar{\theta}_1\right)$ (рис. 8 и 9). Таким образом, при $\ell = 1$ справедливость равенства (17) доказана.

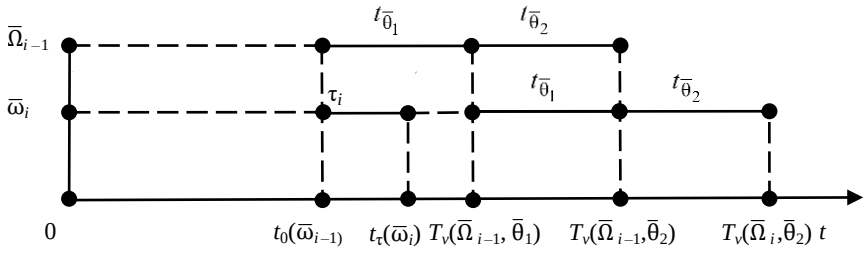


Рис. 9. Временна́я диаграмма при $\tau_i < t_{\bar{\theta}_2}$

Fig. 9. Timing diagram at $\tau_i < t_{\bar{\theta}_2}$

Убедимся в справедливости соотношения (17) при $\ell = 2$. Для этого запишем индуктивное равенство

$$T_v\left(\bar{\Omega}_{i-1}, \bar{\Theta}_2\right) = T_v\left(\bar{\omega}_{i-1}, \bar{\theta}_2\right) = \sum_{p=2}^{i-1} \max\left\{\tau_p, t_{\bar{\theta}_1}\right\} + T_v\left(\bar{\omega}_1, \bar{\Theta}_2\right) \tag{20}$$

и, используя доказанное при $\ell = 1$ равенство (17), вычислим разность

$$\begin{aligned} T_v\left(\bar{\Omega}_i, \bar{\Theta}_1\right) - T_v\left(\bar{\omega}_{i-1}, \bar{\Theta}_2\right) &= \sum_{p=2}^{i} \max\left\{\tau_p, t_{\bar{\theta}_1}\right\} + T_v\left(\bar{\omega}_1, \bar{\Theta}_1\right) - \\ &- \sum_{p=2}^{i-1} \max\left\{\tau_p, t_{\bar{\theta}_1}\right\} - T_v\left(\bar{\omega}_1, \bar{\Theta}_2\right) = \max\left\{\tau_i, t_{\bar{\theta}_1}\right\} - t_{\bar{\theta}_2}. \end{aligned} \tag{21}$$

Возможны два случая: $\tau_i \ge t_{\bar{\theta}_1}$ либо $\tau_i < t_{\bar{\theta}_1}$. В первом случае из разности (21) следует неравенство $\max\left\{\tau_i, t_{\bar{\theta}_1}\right\} - t_{\bar{\theta}_2} \ge 0$, так как верно равенство $t_{\bar{\theta}_2} = t_{\bar{\theta}_1}$, которое влечет $T_v\left(\bar{\Omega}_i, \bar{\Theta}_1\right) \ge T_v\left(\bar{\Omega}_{i-1}, \bar{\Theta}_2\right)$.

Полученное неравенство и доказанное при $\ell = 1$ равенство (17) гарантируют, что к моменту времени $T_v\left(\bar{\Omega}_i, \bar{\Theta}_1\right) = T_v\left(\bar{\omega}_i, \bar{\theta}_1\right)$ операции из $\bar{\Omega}_{i-1}$ закончили обработку операндов из $\bar{\Theta}_2$. Следовательно, выполняются равенства (см. рис. 8)

$$T_v\left(\bar{\Omega}_i, \bar{\Theta}_2\right) = T_v\left(\bar{\omega}_i, \bar{\theta}_1\right) + t_{\bar{\theta}_2} = \sum_{p=2}^{i} \max\left\{\tau_p, t_{\bar{\theta}_1}\right\} + T_v\left(\bar{\omega}_1, \bar{\theta}_1\right) + t_{\bar{\theta}_2} = \sum_{p=2}^{i} \max\left\{\tau_p, t_{\bar{\theta}_1}\right\} + T_v\left(\bar{\omega}_1, \bar{\Theta}_2\right).$$

Таким образом, в первом случае при $\tau_i \geq t_{\bar{\theta}_1}$ справедливость равенства (17) при $\ell = 2$ доказана.

Во втором случае при $\tau_i < t_{\bar{\theta}_1}$ из (21) вытекает равенство $\max\left\{\tau_i, t_{\bar{\theta}_1}\right\} - t_{\bar{\theta}_2} = 0$, так как $t_{\bar{\theta}_1} = \max\left\{\tau_i, t_{\bar{\theta}_1}\right\}$, $t_{\bar{\theta}_2} = t_{\bar{\theta}_1}$, что влечет $T_v\left(\bar{\Omega}_i, \bar{\Theta}_1\right) = T_v\left(\bar{\Omega}_{i-1}, \bar{\Theta}_2\right)$. Поэтому операция $\bar{\omega}_i$ над операндом $\bar{\theta}_2$ начинается в момент времени $T_v\left(\bar{\Omega}_{i-1}, \bar{\Theta}_2\right)$. С учетом равенства (21) это приводит к соотношениям (см. рис. 9)

$$T_v\left(\bar{\Omega}_i, \bar{\Theta}_2\right) = T_v\left(\bar{\Omega}_{i-1}, \bar{\Theta}_2\right) + t_{\bar{\theta}_2} = \sum_{p=2}^{i-1} \max\left\{\tau_p, t_{\bar{\theta}_1}\right\} + T_v\left(\bar{\omega}_1, \bar{\Theta}_2\right) + t_{\bar{\theta}_2} =$$

$$= \sum_{p=2}^{i} \max\left\{\tau_p, t_{\bar{\theta}_1}\right\} + T_v\left(\bar{\omega}_1, \bar{\Theta}_2\right),$$

так как в рассматриваемом случае $t_{\bar{\theta}_2} = t_{\bar{\theta}_1} = \max\left\{\tau_i, t_{\bar{\theta}_1}\right\}$. Следовательно, лемма 2 доказана.

Теорема 1. *Для любых* $2 \leq \ell \leq m$, $2 \leq i \leq n$ *справедливо равенство*

$$T_v\left(\bar{\Omega}_i, \bar{\Theta}_\ell\right) = T_v\left(\bar{\omega}_i, \bar{\Theta}_\ell\right) = \sum_{p=2}^{i} \max\left\{\tau_p, t_{\bar{\theta}_1}\right\} + T_v\left(\bar{\omega}_1, \bar{\Theta}_\ell\right). \tag{22}$$

Доказательство. Равенство (7), доказанное для $i = 3$ и любых $2 \leq \ell \leq m$ в лемме 1, и равенство (17) из леммы 2, справедливость которого доказана для $\ell = 2$ и любых $2 \leq i \leq n$, позволяют сделать два индуктивных предположения о выполнении равенств

$$\begin{aligned} T_v\left(\bar{\Omega}_i, \bar{\Theta}_{\ell-1}\right) &= T_v\left(\bar{\omega}_i, \bar{\Theta}_{\ell-1}\right) = \sum_{p=2}^{i} \max\left\{\tau_p, t_{\bar{\theta}_1}\right\} + T_v\left(\bar{\omega}_1, \bar{\Theta}_{\ell-1}\right), \\ T_v\left(\bar{\Omega}_{i-1}, \bar{\Theta}_\ell\right) &= T_v\left(\bar{\omega}_{i-1}, \bar{\Theta}_\ell\right) = \sum_{p=2}^{i-1} \max\left\{\tau_p, t_{\bar{\theta}_1}\right\} + T_v\left(\bar{\omega}_1, \bar{\Theta}_\ell\right). \end{aligned} \tag{23}$$

Из приведенных равенств вытекает справедливость соотношений

$$T_v\left(\bar{\omega}_i, \bar{\Theta}_{\ell-1}\right) - T_v\left(\bar{\omega}_{i-1}, \bar{\Theta}_\ell\right) = \sum_{p=2}^{i-1} \max\left\{\tau_p, t_{\bar{\theta}_1}\right\} + \max\left\{\tau_i, t_{\bar{\theta}_1}\right\} + T_v\left(\bar{\omega}_1, \bar{\Theta}_{\ell-1}\right) -$$

$$- \sum_{p=2}^{i-1} \max\left\{\tau_p, t_{\bar{\theta}_1}\right\} - T_v\left(\bar{\omega}_1, \bar{\Theta}_{\ell-1}\right) - t_{\bar{\theta}_\ell} = \max\left\{\tau_i, t_{\bar{\theta}_1}\right\} - t_{\bar{\theta}_\ell},$$

следствием которых является равенство

$$T_v\left(\bar{\omega}_i, \bar{\Theta}_{\ell-1}\right) = T_v\left(\bar{\omega}_{i-1}, \bar{\Theta}_\ell\right) + \max\left\{\tau_i, t_{\bar{\theta}_1}\right\} - t_{\bar{\theta}_\ell}. \tag{24}$$

Возможны два случая: $\tau_i \geq t_{\bar{\theta}_1}$ либо $\tau_i < t_{\bar{\theta}_1}$. В первом случае следствием равенств (24) и $t_{\bar{\theta}_\ell} = t_{\bar{\theta}_1}$ является неравенство $T_v\left(\bar{\omega}_i, \bar{\Theta}_{\ell-1}\right) \geq T_v\left(\bar{\omega}_{i-1}, \bar{\Theta}_\ell\right)$, которое означает, что к моменту времени $T_v\left(\bar{\omega}_i, \bar{\Theta}_{\ell-1}\right)$ операция $\bar{\omega}_{i-1}$ над операндом $\bar{\theta}_\ell$ уже закончилась. Поэтому операция $\bar{\omega}_i$ над операндом $\bar{\theta}_\ell$ должно начаться в момент времени $T_v\left(\bar{\omega}_i, \bar{\Theta}_{\ell-1}\right)$. Таким образом, имеет место равенство $T_v\left(\bar{\omega}_i, \bar{\Theta}_\ell\right) = T_v\left(\bar{\omega}_i, \bar{\Theta}_{\ell-1}\right) + t_{\bar{\theta}_\ell}$. Из этого равенства и первого равенства (23) вытекает справедливость соотношений

$$T_v\left(\bar{\omega}_i, \bar{\Theta}_\ell\right) = T_v\left(\bar{\omega}_i, \bar{\Theta}_{\ell-1}\right) + t_{\bar{\theta}_\ell} = \sum_{p=2}^{i} \max\left\{\tau_p, t_{\bar{\theta}_1}\right\} + T_v\left(\bar{\omega}_1, \bar{\Theta}_{\ell-1}\right) + t_{\bar{\theta}_\ell} = \sum_{p=2}^{i} \max\left\{\tau_p, t_{\bar{\theta}_1}\right\} + T_v\left(\bar{\omega}_1, \bar{\Theta}_\ell\right),$$

следствием которых является равенство (22) в первом случае.

Во втором случае при $\tau_i < t_{\bar{\theta}_1}$ выполняется равенство $T_v\left(\bar{\omega}_i, \bar{\Theta}_{\ell-1}\right) = T_v\left(\bar{\omega}_{i-1}, \bar{\Theta}_\ell\right)$, так как из $\tau_i < t_{\bar{\theta}_1}$ и $t_{\bar{\theta}_\ell} = t_{\bar{\theta}_1}$ следует, что $\max\left\{\tau_i, t_{\bar{\theta}_1}\right\} - t_{\bar{\theta}_\ell} = 0$. Таким образом, операция $\bar{\omega}_i$ над операндом $\bar{\theta}_\ell$ начинается в момент времени $T_v\left(\bar{\omega}_{i-1}, \bar{\Theta}_\ell\right)$, что влечет $T_v\left(\bar{\omega}_i, \bar{\Theta}_\ell\right) = T_v\left(\bar{\omega}_{i-1}, \bar{\Theta}_\ell\right) + t_{\bar{\theta}_\ell}$. Из этого равенства и второго равенства (23) вытекает справедливость соотношений

$$T_v\left(\bar{\omega}_i, \bar{\Theta}_\ell\right) = \sum_{p=2}^{i-1} \max\left\{\tau_p, t_{\bar{\theta}_1}\right\} + T_v\left(\bar{\omega}_1, \bar{\Theta}_\ell\right) + t_{\bar{\theta}_\ell} = \sum_{p=2}^{i} \max\left\{\tau_p, t_{\bar{\theta}_1}\right\} + T_v\left(\bar{\omega}_1, \bar{\Theta}_\ell\right),$$

так как $\max\left\{\tau_i, t_{\bar{\theta}_\ell}\right\} = t_{\bar{\theta}_\ell}$ при $\tau_i < t_{\bar{\theta}_1}$. Из полученных соотношений следует справедливость (22) во втором случае. Следовательно, теорема 1 доказана.

Прямым следствием лемм 1, 2 и теоремы 1 является

Теорема 2. *Время выполнения операций из $\bar{\Omega}$ над множеством операндов из $\bar{\Theta}$ в конвейерном режиме определяется равенством*

$$T_v\left(\bar{\Omega}, \bar{\Theta}\right) = T_v\left(\bar{\omega}_n, \bar{\Theta}\right) = \sum_{p=2}^{n} \max\left\{\tau_p, t_{\bar{\theta}_1}\right\} + T_v\left(\bar{\omega}_1, \bar{\Theta}\right). \qquad (25)$$

Оценка возможного ускорения. Для оценки ускорения при конвейерном режиме выполнения последовательных вычислений используется коэффициент

$$K_{eff} = \frac{T_s(\Omega, \Theta)}{T_v\left(\bar{\Omega}, \bar{\Theta}\right)}.$$

В реальных векторно-конвейерных ЭВМ длина векторного операнда составляет от 64 до 128 и более машинных слов, поэтому при оценке коэффициента ускорения можно полагать с учетом равенства (2), что верно соотношение

$$\max\left\{\tau_p, t_{\bar{\theta}_1}\right\} = t_{\bar{\theta}_1} = kt, \quad p = 2, 3, \ldots, n. \qquad (26)$$

Кроме того, время обработки каждой из компонент векторных операндов из $\bar{\Theta}$ векторными операциями из $\bar{\Omega}$ равно одному такту. Следовательно, не ограничивая общности, можно считать, что для времен t_{ij} обработки скалярными операциями $\Omega_i \in \Omega$ скалярных операндов $\Theta_j \in \Theta$ выполняются неравенства $t_{ij} \gg t$ для $i = 1, 2, \ldots, n$, $j = 1, 2, \ldots, N$.

Используя равенства (1), (2), (25), (26) и неравенства $t_{\min} \geq t$ и $\tau_1 \leq kt$, в случае $N = mk$ получим следующую нижнюю оценку коэффициента ускорения:

$$K_{eff} = \frac{T_s(\Omega,\Theta)}{T_v\left(\bar{\Omega},\bar{\Theta}\right)} = \frac{\sum_{i=1}^{n}\sum_{j=1}^{N} t_{ij}}{\sum_{p=2}^{i} \max\left\{\tau_p, t_{\bar{\theta}_1}\right\} + T_v\left(\bar{\omega}_1,\bar{\Theta}\right)} \geq \frac{nNt_{\min}}{\tau_1 + (n-1)kt + mkt} \geq \frac{nmkt}{\tau_1 + (n-1)kt + mkt} \geq \frac{nm}{n+m}.$$

Использование неравенства $\tau_1 \leq kt$ при получении оценки приемлемо, так как настройка векторных операций на выполнение составляет несколько машинных тактов, т. е. $\tau_1 / kt < 1$.

После умножения числителя и знаменателя отношения $\frac{nm}{n+m}$ на k найденную нижнюю оценку для K_{eff} можно записать в равносильном виде:

$$K_{eff} \geq \frac{nN}{nk+N}.$$

Для оценки качества векторизации программ или алгоритмов применяется показатель векторизуемости [1], определяемый равенством $\nu = \tilde{N} / \tilde{T}$, где $\tilde{N}$ – число операций алгоритма, а $\tilde{T}$ – число тактов векторно-конвейерной ЭВМ, требуемых для его выполнения. В зависимости от значения ν предложена следующая схема классификации алгоритмов: $\nu \leq 1/4$ – алгоритм скалярный, $\nu \leq 1/2$ – полувекторный, $\nu \approx 1$ – векторный, $\nu \approx 2$ и более – супервекторный.

Полученная оценка позволяет классифицировать предложенную схему векторизации последовательных вычислений. Так как $\tilde{N} = nN$, $\tilde{T} = nk + N$, простая проверка показывает, что рассмотренный алгоритм векторизации последовательных вычислений является скалярным при $n = 5$, $k = 64$, $N = 20$, $\nu \approx 0{,}29$; векторным при $n = 5$, $k = 64$, $N = 100$, $\nu \approx 1{,}19$; супервекторным при $n = 5$, $k = 64$, $N = 150$, $\nu \approx 2{,}40$.

Заключение. Доказано, что при выполнении последовательных вычислений с использованием векторных операций возможно ускорение не менее чем в $nN/(nk+N)$ раз, где N – общее число обрабатываемых скалярных операндов, n – число скалярных и соответствующих им векторных операций, k – длина конвейера векторных операций, т. е. число машинных слов, одновременно обрабатываемых векторной операцией в последовательном режиме. Оценка коэффициента ускорения при конвейеризации вычислений приводится в сравнении с последовательным режимом их выполнения.

Вклад авторов. *В. М. Демиденко* провел анализ архитектурных и вычислительных особенностей современных векторно-конвейерных вычислительных систем и сделал расчет временных затрат, связанных с конвейеризацией трудных для распараллеливания последовательных вычислений. *В. И. Бенедиктович* получил нижние оценки временного ускорения при выполнении последовательных вычислений в векторном режиме.

Список использованных источников

1. Векторизация программ: теория, методы, реализация : сб. ст. / под ред. Г. Д. Чинина. – М. : Мир, 1991. – 271 с.

2. Воеводин, В. В. Параллельные вычисления / В. В. Воеводин, Вл. В. Воеводин. – СПб. : БХВ-Петербург, 2002. – 608 с.

3. Топорков, В. В. Модели распределенных вычислений / В. В. Топорков. – М. : Физматлит, 2004. – 320 с.

4. Миллер, Р. Последовательные и параллельные алгоритмы: общий подход / Р. Миллер, Л. Боксер ; пер. с англ. – М. : БИНОМ. Лаборатория знаний, 2009. – 406 с.

5. Pllana, S. Programming multi core and many core computing system / S. Pllana, F. Xhafa. – John Wiley & Sons, 2017. – 528 p.

6. Демиденко, В. М. Библиотека базовых программных модулей решения комбинаторных задач упорядочения для векторно-конвейерной ЭВМ / В. М Демиденко, П. С. Кляус, Н. С. Коваленко // Проблемы создания суперЭВМ, суперсистем и эффективность их применения : тез. докл. Первой Всесоюз. конф., Минск, 15–17 сент. 1987. – Минск, 1987. – Ч. 2. – С. 31–32.

7. Демиденко, В. М. О возможности эффективной конвейеризации одного комбинаторного алгоритма / В. М. Демиденко, П. С. Кляус // Вопросы кибернетики. Разработка и использование суперЭВМ : сб. науч. тр. – М. : Институт проблем кибернетики, 1987. – Вып. 7. – С. 96–104.

8. Демиденко, В. М. Базовые процедуры в алгоритмах лексикографической цифровой сортировки и их реализации на векторно-конвейерной ЭВМ / В. М. Демиденко, Л. И. Шевченко // Проблемы создания суперЭВМ, суперсистем и эффективность их применения : тез. докл. Первой Всесоюз. конф., Минск, 15–17 сент. 1987. – Минск, 1987. – Ч. 2. – С. 33–35.

References

1. In Chinin G. D. (ed.) Vektorizatsiya programm: teoriya, metody, realizatsiya : sbornik statej. *Vectorization of Programs: Theory, Methods, Implementation: Collection of Articles.* Moscow, Mir, 1991, 271 p. (In Russ.).

2. Voyevodin V. V., Voyevodin Vl. V. Parallel'nyye vychisleniya. *Parallel Computing*. Saint Petersburg, BHV-Peterburg, 2002, 608 p. (In Russ.).

3. Toporkov V. V. Modeli raspredelennykh vychisleniy. *Distributed Computing Models*. Moscow, Fizmatlit, 2004, 320 p. (In Russ.).

4. Miller R., Boxer L. *Algorithms Sequential and Parallel: A Unified Approach.* 1st ed. Prentice Hall, 1999, 330 p.

5. Pllana S., Xhafa F. *Programming Multi Core and Many Core Computing System*. John Wiley & Sons, 2017, 528 p.

6. Demidenko V. M., Klyaus P. S., Kovalenko N. S. *Library of basic software modules for solving combinatorial ordering problems for a vector-conveyor computer*. Problemy sozdaniya superEVM, supersistem i effektivnost' ikh primeneniya : tezisy dokladov Pervoy Vsesoyuznoy konferentsii, Minsk, 15–17 sentyabrya 1987 [*Problems of Creating supercomputers, Supersystems and the Effectiveness of Their Application: Abstracts of the Report of the First All-Union Conference, Minsk, 15–17 September 1987*]. Minsk, 1987, part 2, pp. 31–32 (In Russ.).

7. Demidenko V. M., Klyaus P. S. *On the possibility of efficient pipelining of one combinatorial algorithm*. Voprosy kibernetiki. Razrabotka i ispol'zovaniye superEVM [*Questions of Cybernetics. Development and use of Supercomputers*]. Moscow, Institute of Problems of Cybernetics, 1987, iss. 7, pp. 96–104 (In Russ.).

8. Demidenko V. M., Shevchenko L. I. *Basic procedures in lexicographic digital sorting algorithms and their implementation on a vector-conveyor computer*. Problemy sozdaniya superEVM, supersistem i effektivnost' ikh primeneniya : tezisy dokladov Pervoy Vsesoyuznoy konferentsii, Minsk, 15–17 sentyabrya 1987 [*Problems of Creating supercomputers, Supersystems and the Effectiveness of Their Application: Abstracts of the Report of the First All-Union Conference, Minsk, 15–17 September 1987*]. Minsk, 1987, part 2, pp. 33–35 (In Russ.).

Информация об авторах

Демиденко Виталий Михайлович, доктор физико-математических наук, профессор, Белорусский государственный экономический университет.
E-mail: vmdemidenko@yandex.ru

Бенедиктович Владимир Иванович, кандидат физико-математических наук, ведущий научный сотрудник, Институт математики Национальной академии наук Беларуси.
E-mail: benedvi@gmail.com

Information about the authors

Vitaly M. Demidenko, D. Sc. (Phys.-Math.), Professor, Belarusian State Economic University.
E-mail: vmdemidenko@yandex.ru

Vladimir I. Benedictovich, Ph. D. (Phys.-Math.), Leading Researcher, Institute of Mathematics of the National Academy of Sciences of Belarus.
E-mail: benedvi@gmail.com

МАТЕМАТИЧЕСКОЕ МОДЕЛИРОВАНИЕ
MATHEMATICAL MODELING

УДК 004.942
https://doi.org/10.37661/1816-0301-2022-19-1-88-95

Обзорная статья
Original Paper

Методология разработки программного обеспечения с использованием модели распределенных объектно-ориентированных стохастических гибридных систем

Р. Е. Шарыкин

Белорусский государственный университет,
пр. Независимости, 4, Минск, 220030, Беларусь
✉*E-mail: sharykin@bsu.edu*

Аннотация
В статье представляется методология разработки программного обеспечения на основе модели распределенных объектно-ориентированных стохастических гибридных систем. Предлагается ориентироваться на создание математической модели для рассматриваемой системы вместе с ее спецификацией на всех этапах разработки целевого программного обеспечения.
Выделяются такие этапы разработки, как построение предварительной математической модели посредством составления ее спецификации, выбор и спецификация метрик системы, статистический анализ модели, апробация в условиях, приближенных к реальным, анализ с применением аналитических методов и реализация. Предлагаются формализм для описания рассматриваемой системы и подход к ее анализу, согласно результатам которого спецификация модели и соответствующая ей математическая модель модифицируются посредством выполнения этапов разработки. Такой подход позволяет получить на выходе не только готовое программное обеспечение, но и математическую модель с изученными свойствами, реализацией которой является данное программное обеспечение.

Ключевые слова: математическое моделирование, переписывающая логика, статистический анализ, стохастические гибридные системы, разработка программного обеспечения

Для цитирования. Шарыкин, Р. Е. Методология разработки программного обеспечения с использованием модели распределенных объектно-ориентированных стохастических гибридных систем / Р. Е. Шарыкин // Информатика. – 2022. − Т. 19, № 1. – С. 88–95. https://doi.org/10.37661/1816-0301-2022-19-1-88-95

Конфликт интересов. Автор заявляет об отсутствии конфликта интересов.

Поступила в редакцию | Received 25.02.2022
Подписана в печать | Accepted 01.03.2022
Опубликована | Published 29.03.2022

Methodology of software development with the use of the model of distributed object-based stochastic hybrid systems

Raman E. Sharykin

Belarusian State University,
av. Nezavisimosti, 4, Minsk, 220030, Belarus
✉*E-mail: sharykin@bsu.edu*

Abstract
Software development methodology based on the model of distributed object-based stochastic hybrid systems is proposed. Such mathematical model is planned to be created for the system being designed along with the system specification at all stages of the software development.
The following stages of the development are considered as building of preliminary mathematical model by designing its specification, choosing and specification of system metrics, statistical analysis of the model, approbation of the mathematical model in conditions close to real, analysis by analytical methods and the implementation. The formalism is proposed for describing the system under consideration and an approach to its analysis. At the steps of the methodology, we adjust the model specification and its corresponding mathematical model in accordance with the results of the analysis. This approach allows to develop not only the software, but also a mathematical model with its properties, which implementation is the resulting software.

Keywords: mathematical modeling, rewriting logic, statistical analysis, stochastic hybrid systems, software development.

For citation. Sharykin R. E. *Methodology of software development with the use of the model of distributed object-based stochastic hybrid systems*. Informatika [*Informatics*], 2022, vol. 19, no. 1, pp. 88–95 (In Russ.).
https://doi.org/10.37661/1816-0301-2022-19-1-88-95

Conflict of interest. The author declare of no conflict of interest.

Введение. Ввиду растущей сложности распределенных систем, разрабатываемых в настоящее время, применение формальных методов для понимания динамики таких систем на этапах их разработки представляется важной задачей. Одно из неудобств применения формальных методов в данной области объясняется трудностью определения подходящей формальной модели, например сети Петри или дискретной марковской цепи. После того как выбор формальной модели сделан, заменить ее будет очень затратно в случае, если обнаружится, что другая формальная модель лучше подходит для описания рассматриваемой системы. В связи с этим важной задачей является разработка подхода, который бы позволял не ограничиваться возможностями отдельного формализма и проводить анализ системы так, чтобы еще до выбора некоторой формальной модели можно было на основе экспериментов учесть аспекты, которые влияют на ее дизайн как можно более полно.

Сложности, возникающие при применении формальных методов в процессе разработки программного обеспечения, описаны в следующих работах.

В статье [1] рассматривается использование мониторов реального времени исполнения для верификации заранее заданных свойств безопасности. Проводится формальное исследование такого подхода, разрабатывается семантика языка спецификации свойств, операционная семантика процесса встраивания мониторов в систему и доказывается корректность этого метода. Ограничения проводимого исследования включают рассмотрение системы в виде ее реализации на конкретном языке программирования, в данном случае Java. При переходе к реализации могут утрачиваться некоторые аспекты системы, например замена дифференциальных уравнений разностными схемами, что влечет за собой появление погрешностей.

В работе [2] предлагается модель параллельных стохастических гибридных систем. Стохастические гибридные системы [3] позволяют моделировать системы, имеющие как дискретную,

так и непрерывную составляющие в условиях вероятностной неопределенности. В данной модели система моделируется множеством агентов, динамика каждого агента следует динамике некоторой стохастической гибридной системы [3] и агенты взаимодействуют друг с другом посредством общих переменных. Показывается, что композиция агентов является корректной операцией. Ограничения данного исследования включают сложность моделирования распределенных систем, сообщающихся путем асинхронных сообщений, а также отсутствие объектно-ориентированного подхода, позволяющего единообразно задавать конструкцию многих объектов в рамках единой структуры.

Для того чтобы учесть аспекты, возникающие при разработке систем (распределенных, асинхронных, стохастических, объектно-ориентированных), необходим формализм, имеющий самые широкие выразительные возможности. Формализм переписывающей логики [4] хорошо удовлетворяет требованию не ограничиваться отдельным «классическим» формализмом. Многие общепринятые виды формализма имеют известное представление в виде «переписывающих теорий» [4, 5]. Вероятностная версия переписывающей логики [6, 7] является еще более подходящим выбором, так как позволяет моделировать стохастические системы. Переписывающие теории дают возможность задавать очень широкий спектр систем, но отсутствие встроенных в математическую модель структур для описания указанных выше аспектов делает их недостаточно удобными при задании систем, состоящих из объектов, которые являются распределенными, объектно-ориентированными, стохастическими и с асинхронной коммуникацией.

Структура модели распределенных объектно-ориентированных стохастических гибридных систем (РООСГС) [8] обусловлена факторами модульности, иерархичности, объектно-ориентированности и распределенной природы таких систем, т. е. позволяет задавать их простым и понятным образом. Преимущество модели РООСГС состоит в возможности задавать систему как набор объединенных в классы объектов, имеющих внутреннее состояние, которое подчиняется динамике, определяемой системой стохастических дифференциальных уравнений, и общающихся посредством асинхронных сообщений. Спецификация модели РООСГС на языке SHYMaude [9] легко транслируется в спецификацию переписывающей логики, тем самым наследуя все ее преимущества, а также может выполняться в системе Maude [10], открывая возможности для непосредственной имитации модели. Такой подход к спецификации системы значительно приближен к реальным задачам и позволяет определять системы формально, но максимально близко к тому, как они устроены на практике.

Методология разработки программного обеспечения. Использование переписывающей логики позволяет начать применение формальных методов на ранних этапах проектирования, в то же время не ограничивая разработчика определенным формализмом. Основанный на переписывающий логике формализм может быть сжатым, интуитивно понятным и хорошо подходить для спецификации распределенных параллельных систем с асинхронной коммуникацией.

Спецификации, выполненные на языке SHYMaude, после трансляции в язык Maude могут дорабатываться с помощью системы Maude [10], что позволяет симулировать и улучшать модель системы в самом начале ее разработки. Как только разработчик остается удовлетворенным полученной спецификацией, он может выбрать наиболее подходящую математическую модель с известным представлением в переписывающей логике для формального доказательства критически важных свойств системы либо нахождения более существенных недостатков, которые невозможно выявить с помощью техники имитаций. Язык Maude также представляет средства, облегчающие проведение формальных доказательств в формализме переписывающей логики [11].

Следующим важным вопросом является анализ системы на ранних этапах ее разработки. Желаемый метод должен быть легким в том смысле, что он может помочь разработчику сконцентрироваться на модели и ее параметрах. В работе [9] предложен метод, с помощью которого можно получить быструю оценку основных поведенческих свойств системы и экспериментально их изучить, позволяя потратить усилия на проведение более сложных формальных процедур валидации на поздних этапах, когда пространство возможностей для дизайна модели уже сокращено.

В рамках методологии, основанной на работе [9], автор предлагает работать со спецификацией системы, выполненной с помощью языка SHYMaude. Для анализа системы рекомендуется использовать:

– инструмент MultiVeStA [12] для выполнения имитаций спецификаций на языке Maude;

– язык MultiQuaTEx [12] для формального задания свойств исследуемой системы. MultiQuaTEx является расширением языка QuaTEx [13], позволяющим более компактно задавать метрики систем;

– клиент-серверную архитектуру для выполнения распределенных имитаций.

В процессе анализа MultiVeStA использует статистический метод Монте-Карло и предоставляет возможность генерации реализаций до достижения предустановленного уровня точности результата. Также система обеспечивает выполнение распределенных вычислений, основанное на клиент-серверном взаимодействии [12, 13].

Опишем поэтапно методологию разработки:

1. Система специфицируется с помощью языка SHYMaude. Спецификация одновременно определяет модель РООСГС. На данном этапе работа по спецификации сродни работе по реализации системы на языке программирования, что значительно облегчает задачу.

2. Выбираются и специфицируются на языке MultiQuaTEx метрики, оценки которых представляются важными разработчику.

3. Используется система MultiVeStA, проводящая статистическую оценку метрик для спецификации системы с помощью метода доверительных интервалов на основе метода Монте-Карло.

4. Полученные результаты анализируются. В случае обнаружения недочетов в спецификации и (или) способов ее «улучшения» спецификация и соответствующая ей модель корректируются. Возвращаемся к этапу 3.

5. Проводится апробация системы. Система реализуется на языке высокого уровня (например, Java), предпочтительно имеющем известное представление в переписывающей логике. Возможно проведение статистического анализа, аналогичного проводимому на этапе 3. В случае нахождения аспектов системы, требующих коррекции и (или) дополнительного анализа (например, возможности реализации некоторого действия несколькими способами), проводятся коррекция и (или) дополнительный анализ, по результатам которого делается выбор предпочтительного способа реализации. Одновременно корректируется исходная модель и, если это видится целесообразным, выполняется переход к этапу 3. Если результаты удовлетворительны и делается вывод, что предварительный анализ модели можно считать законченным, переходим к этапу 6.

6. Имеются две возможности усиления результатов, полученных с помощью предыдущих пунктов:

– *аналитическое исследование свойств.* По имеющейся спецификации подбирается подходящая математическая модель и аналитически исследуется с доказательством интересующих свойств модели;

– *автоматическое доказательство свойств.* Из системы удаляются вероятности и недетерминизм, и используется система автоматического доказательства теорем системы Maude [14] для получения дополнительных доказанных утверждений относительно свойств системы. С помощью данного подхода могут доказываться утверждения, не имеющие отношение к вероятностной природе системы, например утверждение, что ни при каких сценариях поведения одного из участников протокола определенные данные не могут быть получены из доступного ему трафика в незашифрованном виде.

Данный этап в основном может потребоваться для приложений, требующих крайне высокой надежности результатов.

7. Реализация системы на практике. Если природа системы позволяет выделять отдельные прогоны в процессе ее работы, то на данном этапе также можно проводить фоновый статистический анализ для оценки основных метрик системы в реальных условиях.

В предлагаемой методологии формальные методы используются в трех аспектах: для задания модели (модель РООСГС), задания метрик (язык QuaTEx) и получения значимых оценок метрик (статистический анализ).

Апробация методологии. В целях апробации изложенной методологии были выполнены разработки системы стохастической коллаборационной защиты от вирусов, системы одного окна и системы закупок предприятия с использованием обратных аукционов.

Система стохастической коллаборационной защиты от вирусов. В работе [15] методология была использована для разработки предварительной модели стохастической групповой системы защиты от вирусов и ее анализа. Продемонстрировано, как получение оценки основных численных метрик системы, обычно используемых для оценки систем защиты от вирусов, дает возможность оценить систему до фиксации формальной модели и проведения ее углубленного анализа. Из результатов анализа видно, что система защиты позволяет «сохранить» значительный процент узлов незараженными при атаке вируса на каждом шаге распространения, выбирающего следующий атакуемый узел в соответствии с равномерным распределением на множестве всех узлов сети. Также изучено влияние алгоритма выбора групп оповещения на эффективность системы защиты. Анализ показывает, что важным условием эффективности системы является равномерное покрытие всего множества узлов группами оповещения при рассматриваемом типе вируса.

Вместе с тем было обнаружено, что существенное упрощение алгоритма выбора групп оповещения не повлияло на эффективность системы при проведении эксперимента. Это указывает на потенциальную возможность упрощения системы с сохранением ее свойств и может быть более тщательно проверено на более поздних этапах разработки.

Полученные результаты дают возможность утверждать, что применение изложенного подхода статистического анализа систем на этапе проектирования позволяет изучать важные аспекты систем до их анализа аналитическими методами, исправлять какие-либо недостатки и определять некоторые потенциальные «упрощения» отдельных компонентов.

В работе [16] была проведена апробация реализации системы стохастической коллаборационной защиты от вирусов в условиях, приближенных к реальным. Рассматривались две вариации алгоритма с использованием протоколов TCP/IP и UDP в части рассылки оповещений об обнаружении вирусов и назначении групп оповещений. Установлено, что применение протокола UDP приводит к увеличению статистической эффективности системы защиты. Исследования возможного потенциала увеличения эффективности в рамках исходной спецификации на языке переписывающей логики проводились путем снижения задержек в доставке сообщений. Так как протокол UDP не предусматривает создания полноценного соединения, доставка сообщения вызывает значительно меньшие задержки, чем доставка при использовании протокола TCP/IP. Предполагалось, что вирусы для своего распространения используют в основном протокол TCP/IP.

При исследовании влияния размера группы оповещения и общего количества узлов с сохранением в процентном соотношении размера групп было выяснено, что увеличение размера общего количества узлов приводит к заметному росту эффективности системы [16]. Вместе с тем для некоторых сценариев заражения это сопряжено с увеличением нагрузки на сеть ввиду одновременного увеличения количества оповещений в сети. Предложен механизм иерархической организации системы защиты для предотвращения быстрого роста данной нагрузки.

Таким образом, предлагаемая система может использоваться для защиты сетей, состоящих из того количества узлов, которое встречается в реальных условиях.

Система одного окна. В работе [17] проиллюстрировано применение методологии на примере разработки системы одного окна, которая была специфицирована с помощью языка SHYMaude [9]. Сформулированы четыре метрики, измеряющие такие параметры, как длина очереди документов, время их обработки, загруженность сотрудников, и позволяющие оценить эффективность такой системы. Для проведения статистического анализа данных метрик использовался инструмент MultiVeStA. Предложенные метрики были специфицированы с помощью языка MultiQuaTEx. Также определено оптимальное количество сотрудников, задействованных в системе на обработке документов. При оптимальном количестве сотрудников сохраняется баланс в виде приемлемых параметров: максимально наблюдаемой длины очереди документов на рассмотрение, времени обработки документов и загруженности сотрудников.

Для того чтобы максимально приблизить систему к практическому применению, введено управление количеством сотрудников, осуществляемое менеджером. При этом был обнаружен

недостаток такой доработки: на раннем этапе функционирования системы предложенный алгоритм неудачно управляет количеством сотрудников, что приводит к значительному росту значения одной из целевых метрик, а именно максимальной наблюдаемой длины очереди, выводя ее из желаемого интервала. Для исправления этого недостатка было предложено ввести правило, при котором менеджер начинает управлять количеством сотрудников только после того, как длина очереди достигает предварительно установленной нижней границы. После устранения недостатка было обнаружено, что максимальная наблюдаемая длина очереди находится в желаемом интервале.

Данное исследование показывает, что использование предлагаемой методологии дает возможность своевременно исправлять грубые ошибки при проектировании систем одного окна.

Система закупок предприятия. В работе [18] методология использована для предварительной оценки модели системы закупок предприятия на примере модели закупок на основе повторяемого обратного аукциона. Модель системы закупок предприятия была специфицирована для двух типов аукционов с помощью языка SHYMaude. Были сформулированы метрики, представляющие интерес с точки зрения оценки эффективности предложенной системы: цена, по которой будет производиться поставка товара после единичного проведения аукциона, и цена поставки, усредненная по множеству проведенных аукционов. Также были определены параметры, которые могут влиять на значения второй рассматриваемой метрики. Для выполнения статистического анализа метрик использован инструмент MultiVeStA, предложенные метрики специфицированы с помощью языка MultiQuaTEx.

В результате анализа модели на тестовых данных было обнаружено, что применение аукциона первой цены дает более низкую цену поставки товара, но эта разница незначительна.

Увеличение времени проведения аукциона приводит к снижению цены поставки. Вместе с тем основное снижение цены происходит в течение времени, которое может доставлять сложности с проведением аукциона, так как время его проведения может быть ограничено. Можно определить некоторое пороговое значение количества участников, после которого дополнительные участники не приводят к заметному уменьшению цены закупки. При увеличении количества раундов аукциона цена снижается в сторону предельно допустимой минимальной цены продажи, установленной в системе, без возможности четко обозначить некоторую переломную точку. Использование повторных аукционов позволяет достичь более низких значений цены поставки товара, чем простое увеличение длительности отдельно взятого аукциона без повторов.

Таким образом, применение техники повторных аукционов дает альтернативу увеличению длительности аукциона для достижения меньшей средней цены поставки, рассчитываемой за долгосрочный период.

Заключение. В статье представлена методология разработки программного обеспечения на основе модели РООСГС. Для формального моделирования разрабатываемых систем предлагается использование данной модели, а для их спецификации – языка SHYMaude. Анализ систем рекомендуется выполнять с помощью статистического подхода. Для более тщательного изучения свойств системы дополнительно к статистическому анализу могут быть применены аналитический подход [5] и (или) подход, основанный на использовании техники автоматического доказательства теорем [11].

Согласно представленной методологии поэтапный процесс разработки системы включает: спецификацию системы; спецификацию требований; трансляцию спецификации системы в ее исполняемую версию, одновременно являющуюся формальной моделью благодаря свойствам переписывающей логики; исследование статистическими методами требований разработчиков, сформулированных в виде метрик; анализ результатов и доработку исходной спецификации; повторение процесса до получения удовлетворяющего требованиям разработчиков результата.

Процесс доработки может быть повторен на этапе апробации системы в условиях, максимально приближенных к реальным, и продолжен на этапе реализации уже практической версии системы. С переходом на каждый последующий этап цена изменений первоначальной модели возрастает, что подтверждает целесообразность разделения процесса разработки на выделенные в статье этапы.

Список использованных источников

1. Formalizing Java-MaC / U. Sammapun [et al.] // Electronic Notes in Theoretical Computer Science. – 2003. – Vol. 89, iss. 2. – P. 171–190.

2. Bernadsky, M. Structured modeling of concurrent stochastic hybrid systems / M. Bernadsky, R. Sharykin, R. Alur // Lecture Notes in Computer Science. – 2004. – Vol. 3253. – P. 309–324.

3. Bujorianu, M. L. Toward a general theory of stochastic hybrid systems / M. L. Bujorianu, J. Lygeros // Lecture Notes in Control and Information Science. – 2006. – Vol. 337. – P. 3–30.

4. Meseguer, J. Conditional rewriting logic as a unified model of concurrency / J. Meseguer // Theoretical Computer Science. – 1992. – Vol. 96, iss. 1. – P. 73–155.

5. Martí-Oliet, N. Rewriting logic: roadmap and bibliography / N. Martí-Oliet, J. Meseguer // Theoretical Computer Science. – 2002. – Vol. 285, iss. 2. – P. 121–154.

6. Agha, G. A. PMaude: Rewrite-based specification language for probabilistic object systems / G. A. Agha, J. Meseguer, K. Sen // Electronic Notes in Theoretical Computer Science. – 2006. – Vol. 153, iss. 2, no. 2. – P. 213–239.

7. A rewriting based model for probabilistic distributed object systems / N. Kumar [et al.] // Lecture Notes in Computer Science. – 2003. – Vol. 2884. – P. 32–46.

8. Шарыкин, Р. Е. Модель распределенных объектно-ориентированных стохастических гибридных систем / Р. Е. Шарыкин, А. Н. Курбацкий // Журнал Белорусского государственного университета. Математика. Информатика. – 2019. – № 2. – С. 52–61.

9. Шарыкин, Р. Е. Верификация распределенных объектно-ориентированных стохастических гибридных систем / Р. Е. Шарыкин, А. Н. Курбацкий // Вестник Гродненского государственного университета имени Янки Купалы. Сер. 2. Математика. Физика. Информатика, вычислительная техника и управление. – 2019. – Т. 9, № 2. – С. 123–133.

10. Maude: Specification and programming in rewriting logic / M. Clavel [et al.] // Theoretical Computer Science. – 2002. – Vol. 285, iss. 2. – P. 187–243.

11. Building equational proving tools by reflection in rewriting logic / M. Clavel [et al.] // CAFE: An Industrial-Strength Algebraic Formal Method. – Amsterdam, 2000. – P. 1–31.

12. Sebastio, S. MultiVeStA: Statistical model checking for discrete event simulators / S. Sebastio, A. Vandin // Proc. of the 7th Intern. Conf. on Performance Evaluation Methodologies and Tools, Torino, Italy, 10–12 Dec. 2013. – Torino, 2013. – P. 310–315.

13. Sen, K. On statistical model checking of stochastic systems / K. Sen, M. Viswanathan, G. Agha // Lecture Notes in Computer Science. – 2005. – Vol. 3576. – P. 266–280.

14. The maude formal tool environment / M. Clavel [et al.] // Lecture Notes in Computer Science. – 2007. – Vol. 4624. – P. 173–178.

15. Шарыкин, Р. Е. Применение формальных методов при проектировании коллаборационной системы противовирусной защиты / Р. Е. Шарыкин, А. Н. Курбацкий // Журнал Белорусского государственного университета. Математика. Информатика. – 2020. – № 1. – P. 59–69.

16. Шарыкин, Р. Е. Апробация модели стохастической коллаборационной защиты от вирусов / Р. Е. Шарыкин // Системный анализ и прикладная информатика. – 2021. – № 4. – С. 62–70.

17. Шарыкин, Р. Е. Применение формальных методов при проектировании системы одного окна / Р. Е. Шарыкин // Журнал Белорусского государственного университета. Математика. Информатика. – 2021. – № 1. – С. 79–90.

18. Шарыкин, Р. Е. Методика применения формальных методов при проектировании системы закупок предприятия / Р. Е. Шарыкин // Вестник Гродненского государственного университета имени Янки Купалы. Сер. 2. Математика. Физика. Информатика, вычислительная техника и управление. – 2022. – Т. 12, № 1. – С. 134–143.

References

1. Sammapun U., Sharykin R., DeLap M., Kim M., Zdancewic S. Formalizing Java-MaC. *Electronic Notes in Theoretical Computer Science*, 2003, vol. 89, iss. 2, pp. 171–190.

2. Bernadsky M., Sharykin R., Alur R. Structured modeling of concurrent stochastic hybrid systems. *Lecture Notes in Computer Science*, 2004, vol. 3253, pp. 309–324.

3. Bujorianu M. L., Lygeros J. Toward a general theory of stochastic hybrid systems. *Lecture Notes in Control and Information Science*, 2006, vol. 337, pp. 3–30.

4. Meseguer J. Conditional rewriting logic as a unified model of concurrency. *Theoretical Computer Science*, 1992, vol. 96, iss. 1, pp. 73–155.

5. Martí-Oliet N., Meseguer J. Rewriting logic: roadmap and bibliography. *Theoretical Computer Science*, 2002, vol. 285, iss. 2, pp. 121–154.

6. Agha G. A., Meseguer J., Sen K. PMaude: Rewrite-based specification language for probabilistic object systems. *Electronic Notes in Theoretical Computer Science*, 2006, vol. 153, iss. 2, no. 2, pp. 213–239.

7. Kumar N., Sen K., Meseguer J., Agha G. A rewriting based model for probabilistic distributed object systems. *Lecture Notes in Computer Science*, 2003, vol. 2884, pp. 32–46.

8. Sharykin R. E., Kourbatski A. N. *A model of distributed object-based stochastic hybrid systems*. Zhurnal Belorusskogo gosudarstvennogo universiteta. Matematika. Informatika [*Journal of the Belarusian State University. Mathematics. Informatics*], 2019, no. 2, pp. 52–61 (In Russ.).

9. Sharykin R. E., Kourbatski A. N. *Verification of distributed object-oriented stochastic hybrid systems.* Vestnik Grodnenskogo gosudarstvennogo universiteta imeni Yanki Kupaly. Seriya 2. Matematika. Fizika. Informatika, vychislitel'naya tekhnika i upravlenie [*Bulletin of Grodno State University named after Yanka Kupala. Series 2. Mathematics. Physics. Informatics, Computer Technology and Management*], 2019, vol. 9, no. 2, pp. 123–133 (In Russ.).

10. Clavel M., Duran F., Eker S., Lincoln P. D. Maude: Specification and programming in rewriting logic. *Theoretical Computer Science*, 2002, vol. 285, iss. 2, pp. 187–243.

11. Clavel M., Duran F., Eker S., Meseguer J. Building equational proving tools by reflection in rewriting logic. *CAFE: An Industrial-Strength Algebraic Formal Method*. Amsterdam, 2000, pp. 1–31.

12. Sebastio S., Vandin A. MultiVeStA: Statistical model checking for discrete event simulators. *Proceedings of the 7th International Conference on Performance Evaluation Methodologies and Tools, Torino, Italy, 10–12 December 2013*. Torino, 2013, pp. 310–315.

13. Sen K., Viswanathan M., Agha G. On statistical model checking of stochastic systems. *Lecture Notes in Computer Science*, 2005, vol. 3576, pp. 266–280.

14. Clavel M., Duran F., Hendrix J., Lucas S., Meseguer J., Olveczky P. The maude formal tool environment. *Lecture Notes in Computer Science*, 2007, vol. 4624, pp. 173–178.

15. Sharykin R. E., Kourbatski A. N. *Application of formal methods in the design of a collaborative virus defense system.* Zhurnal Belorusskogo gosudarstvennogo universiteta. Matematika. Informatika [*Journal of the Belarusian State University. Mathematics. Informatics*], 2020, no. 1, pp. 59–69 (In Russ.).

16. Sharykin R. E. *Approbation of the stochastic group virus protection model.* Sistemnyi analiz i prikladnaia informatika [*System Analysis and Applied Informatics*], 2021, no. 4, pp. 62–70 (In Russ.).

17. Sharykin R. E. *A method of applying format methods in the design of a single window system*. Zhurnal Belorusskogo gosudarstvennogo universiteta. Matematika. Informatika [*Journal of the Belarusian State University. Mathematics. Informatics*], 2021, no. 1, pp. 79–90 (In Russ.).

18. Sharykin R. E. *A methodology to apply formal methods in the design of an enterprise procurement system*. Vestnik Grodnenskogo gosudarstvennogo universiteta imeni Yanki Kupaly. Seriya 2. Matematika. Fizika. Informatika, vychislitel'naya tekhnika i upravlenie [*Bulletin of Grodno State University named after Yanka Kupala. Series 2. Mathematics. Physics. Informatics, Computer Technology and Management*], 2022, vol. 12, no. 1, pp. 134–143 (In Russ.).

Информация об авторе

Шарыкин Роман Евгеньевич, соискатель кафедры технологий программирования, факультет прикладной математики и информатики, Белорусский государственный университет.
E-mail: sharykin@bsu.edu

Information about the author

Raman E. Sharykin, Aspirant of the Department of Software Engineering, Faculty of Applied Mathematics and Computer Science, Belarusian State University.
E-mail: sharykin@bsu.edu

ИНТЕЛЛЕКТУАЛЬНЫЕ СИСТЕМЫ
INTELLIGENT SYSTEMS

УДК 004.8
https://doi.org/10.37661/1816-0301-2022-19-1-96-110

Обзорная статья
Review Paper

Применение модели освоения языка к решению задачи обработки малых языков

Д. И. Качков

Белорусский государственный университет,
пр. Независимости, 4, Минск, 220030, Беларусь
E-mail: dmitriydikanskiy@gmail.com

Аннотация
Решается задача построения компьютерной модели малого языка. Ее актуальность обусловлена необходимостью устранения информационного неравенства между носителями различных языков, востребованностью новых инструментов для исследования малоизученных языков и инновационных подходов к моделированию языка в условиях дефицита ресурсов, необходимостью поддержки и развития языков малых народов.
При решении задачи обработки малых языков на этапе описания проблемной ситуации преследуются три основные цели: обоснование проблемы моделирования языка в условиях дефицита ресурсов как особой задачи в сфере моделирования естественных языков, обзор литературы по соответствующей теме и разработка концепции модели усвоения языка с относительно малым числом доступных ресурсов.
Используются методы компьютерного моделирования с применением нейронных сетей, обучение с частичным привлечением учителя и обучение с подкреплением.
В работе приведен обзор литературы, посвященной моделированию изучения лексики, морфологии и грамматики родного языка ребенком. На основании современных представлений о ходе изучения языка предложена архитектура системы обработки малого языка, которая при обучении опирается на компьютерное моделирование онтогенеза. Выделены основные компоненты системы и принципы их взаимодействия. В основе системы лежит модуль, построенный на базе современных диалоговых языковых моделей и обученный на каком-либо крупном языке, например английском. При обучении используется промежуточный слой, который представляет высказывания в некотором абстрактном виде, например, в символах формальной семантики. Соотношение между формальной записью высказываний и их переводом на целевой малый язык изучается методом моделирования процесса усвоения лексики и грамматики языка ребенком. Отдельный компонент имитирует неязыковой контекст, в котором происходит изучение языка.
В статье исследуется задача моделирования малых языков. Дано подробное обоснование актуальности моделирования малых языков: показана социальная значимость этой проблемы, польза ее решения для лингвистики, этнографии, этнологии и культурной антропологии. Отмечена неэффективность подходов, применяемых к крупным языкам, в условиях дефицита ресурсов. Предложена модель изучения языка с помощью имитации онтогенеза, которая опирается как на полученные результаты в области компьютерного моделирования, так и на данные психолингвистики.

Ключевые слова: информационные технологии, языковые модели, обработка малого языка, усвоение языка, обучение с подкреплением, нейронные сети, архитектура Transformer

Для цитирования. Качков, Д. И. Применение модели освоения языка к решению задачи обработки малых языков / Д. И. Качков // Информатика. – 2022. − Т. 19, № 1. – С. 96–110.
https://doi.org/10.37661/1816-0301-2022-19-1-96-110

Конфликт интересов. Автор заявляет об отсутствии конфликта интересов.

Поступила в редакцию | Received 18.11.2021
Подписана в печать | Accepted 08.12.2021
Опубликована | Published 29.03.2022

Applying the language acquisition model to the solution small language processing tasks

Dzmitry I. Kachkou

Belarusian State University,
av. Nezavisimosti, 4, Minsk, 220030, Belarus
E-mail: dmitriydikanskiy@gmail.com

Abstract
The problem of building a computer model of a small language was under solution. The relevance of this task is due to the following considerations: the need to eliminate the information inequality between speakers of different languages; the need for new tools for the study of poorly understood languages, as well as innovative approaches to language modeling in the low-resource context; the problem of supporting and developing small languages.
There are three main objectives in solving the problem of small natural language processing at the stage of describing the problem situation: to justify the problem of modeling language in the context of resource scarcity as a special task in the field of natural languages processing, to review the literature on the relevant topic, to develop the concept of language acquisition model with a relatively small number of available resources.
Computer modeling techniques using neural networks, semi-supervised learning and reinforcement learning were involved.
The paper provides a review of the literature on modeling the learning of vocabulary, morphology, and grammar of a child's native language. Based on the current understanding of the language acquisition and existing computer models of this process, the architecture of the system of small language processing, which is taught through modeling of ontogenesis, is proposed. The main components of the system and the principles of their interaction are highlighted. The system is based on a module built on the basis of modern dialogical language models and taught in some rich-resources language (e.g., English). During training, an intermediate layer is used which represents statements in some abstract form, for example, in the symbols of formal semantics. The relationship between the formal recording of utterances and their translation into the target low-resource language is learned by modeling the child's acquisition of vocabulary and grammar of the language. One of components stands for the non-linguistic context in which language learning takes place.
This article explores the problem of modeling small languages. A detailed substantiation of the relevance of modeling small languages is given: the social significance of the problem is noted, the benefits for linguistics, ethnography, ethnology and cultural anthropology are shown. The ineffectiveness of approaches applied to large languages in conditions of a lack of resources is noted. A model of language learning by means of ontogenesis simulation is proposed, which is based both on the results obtained in the field of computer modeling and on the data of psycholinguistics.

Keywords: information technology, language models, low-resource language processing, language acquisition, reinforcement learning, neural networks, Transformer architecture

For citation. Kachkou D. I. *Applying the language acquisition model to the solution small language processing tasks*. Informatika [*Informatics*], 2022, vol. 19, no. 1, pp. 96−110 (In Russ.).
https://doi.org/10.37661/1816-0301-2022-19-1-96-110

Conflict of interest. The author declare of no conflict of interest.

Введение. Решение проблемы автоматической обработки естественного языка является одним из наиболее актуальных направлений в области искусственного интеллекта. Современные языковые модели, как правило, основанные на архитектуре Transformer, позволяют достичь впечатляющих результатов в решении разнообразных задач, связанных с обработкой и пониманием текстов. Обучение таких моделей происходит в два этапа: первый требует огромного корпуса неразмеченных данных, второй – небольшой выборки размеченных данных, связанных с конкретной задачей.

Как показывают исследования, для раскрытия всего потенциала языковых моделей корпус неразмеченных текстов должен насчитывать миллиарды слов. Очевидно, что подобные обучающие выборки можно построить только для крупных языков: английского, русского, китайского, которые имеют развитую литературную традицию и широко используются в Интернете. Между тем автоматическая обработка малых языков – не менее важная задача для современной науки, решение которой позволит внести весомый вклад как в сохранение исчезающих языков, так и в компьютерную лингвистику.

Основные подходы, выработанные в рамках проблемы обработки малых языков, адаптируют крупные языковые модели к условиям дефицита доступных ресурсов. В первую очередь это методы искусственной генерации дополнительных текстов на целевом языке и трансфер знаний от модели крупного языка к модели малого языка.

Другая сфера компьютерной лингвистики – моделирование онтогенеза языка, т. е. процесса усвоения языка ребенком. Как правило, подобные модели используются для исследования гипотез, связанных с процессом освоения родного языка.

В настоящей работе выдвигается гипотеза о применимости метода моделирования онтогенеза к задаче моделирования малых языков.

1. Малые языки. Автоматическая обработка языков с дефицитом ресурсов (англ. low-resource languages) сформировалась как отдельная проблема в области компьютерной лингвистики. Исследователи активно ищут технологии и подходы, которые позволят повысить эффективность решения языковых задач в условиях малого объема доступной обучающей выборки.

Остановимся подробнее на термине «языки с дефицитом ресурсов». Можно выделить следующие категории ресурсов [1]:

1. Данные, размеченные для решаемой задачи. Размеченные данные необходимы для проведения обучения с учителем, поэтому отсутствие их в достаточном количестве становится значительной проблемой при решении поставленной задачи. Стоит отметить, что разметка данных нередко проводится вручную специалистами в данном языке и (или) в данной предметной области. Такие специалисты могут отсутствовать, в частности, по причине экзотичности рассматриваемого языка. Отсутствие соответствующего эксперта дополнительно осложняет решение задачи обработки языка.

2. Неразмеченные тексты на целевом языке или на целевую тему. Во многих современных моделях применяется «обучение с частичным привлечением учителя» (semi-supervised learning) [2]: на первом этапе происходит обучение без учителя на большом корпусе неразмеченных данных, на втором – дообучение под конкретную задачу на небольшом количестве размеченных примеров. Первый этап обучения требует больших корпусов текстов. Если корпус достаточного объема оказывается недоступен, построение эффективных языковых моделей на базе Transformer значительно затрудняется.

3. Вспомогательные ресурсы – любые другие ресурсы, не упомянутые ранее, которые могут быть использованы для решения поставленной задачи обработки языка. К ним можно отнести, например, корпусы параллельных текстов на целевом и некотором другом языке, автоматический переводчик на целевой язык, базы знаний, справочники и т. д.

Здесь и далее под языком может пониматься не только любой естественный или искусственный язык, но и некоторое подмножество естественного языка, например профессиональный жаргон моряков или русский язык опубликованных в Интернете текстов.

Обозначим основные ситуации, в которых можно столкнуться с дефицитом ресурсов [1]:

1. Целевой язык – малоизученный язык с малым числом носителей, например один из коренных языков жителей Южной Америки. Такому языку свойственен жесткий дефицит всех ресурсов.

2. Целевой язык – язык, число носителей которого относительно велико, но который не попадал во внимание компьютерных лингвистов. В качестве примера можно привести распространенные на полуострове Сомали кушитские языки: сидамо, хатия, камбата, каждым из которых владеет более миллиона человек. Для такого языка может наблюдаться недостаток оцифрованных текстов, корпусов размеченных и неразмеченных данных.

3. Целевой язык подробно исследован с точки зрения автоматической обработки (например, английский, русский), однако поставлена нестандартная задача или выбрана нетривиальная предметная область.

В настоящей работе исследован первый сценарий, касающийся обработки находящихся под угрозой исчезновения языков с малым числом носителей. В дальнейшем будем именовать их «малыми языками» по аналогии с расхожим термином «малые народы». Подчеркнем, что многие из изложенных ниже соображений будут применимы и к прочим сценариям, особенно к случаю крупных, но малоизученных языков.

Рассмотрим конкретный малый язык – язык южноамериканских индейцев туюка, относящийся к туканской языковой семье. Для оценки количества доступных ресурсов на этом языке был составлен корпус неразмеченных текстов. Анализ библиографической базы данных Glottolog[1], архива общественной организации SIL International[2] и базы языковых ресурсов Open Language Archives Community[3] показал, что в настоящее время существует менее 100 материалов, относящихся к языку туюка, причем некоторые из материалов представляют собой переиздание ранних публикаций. Многие из книг не были оцифрованы и недоступны в электронном виде. Важной составляющей корпуса неразмеченных текстов является перевод Нового Завета (URL: http://gospelgo.com/s/tuyuca_nt.htm). Вместе с ним в корпус вошли ряд опубликованных текстов на туюка, фразы из испанско-туюка и туюка-испанского словарей, а также предложения, использованные в качестве примеров в грамматических очерках. Суммарный объем корпуса составил около 180 000 слов. Для языковых моделей с современной архитектурой такого объема недостаточно, для их обучения используются корпуса на миллиард слов. Таким образом, моделирование языка туюка производится в условиях дефицита ресурсов.

2. Актуальность проблемы моделирования малых языков. В поддержку актуальности проблемы обработки языков с малым числом ресурсов можно высказать следующие соображения.

Во-первых, для носителей малого языка система обработки родного языка имеет такую же ценность, как для носителей английского, русского или любого другого крупного языка. Ценность представляют, например, механизмы рекомендаций в социальных сетях, способные анализировать интересы пользователя и предлагать ему другие публикации на схожие темы, автоматическим образом определяя их тему и тональность. Другой пример – интерфейсы взаимодействия с искусственными интеллектуальными системами на естественном языке, в частности чат-боты, способные поддержать диалог с пользователем, распознать некоторую типовую проблему и подсказать ее решение. Наконец, что, может быть, наиболее значимо, подобные системы могут стать шагом в направлении устранения проблемы информационного неравенства, от которого могут страдать носители редких и малоизученных языков. В настоящее время подавляющее большинство материалов в глобальной сети Интернет доступно на крупных языках, в первую очередь на английском, а также на русском, китайском, испанском и т. д. Эта информация включает в себя статьи и рекомендации по медицине, экономике, бытовым вопросам, разнообразные обучающие материалы и инструкции. Носители малых языков, не владеющие в достаточной степени крупными языками, не могут потреблять эту информацию. Отсюда возникает существенная социальная проблема информационного неравенства: не все жители Земли в равной мере обеспечены доступом к накопленной человечеством информа-

[1]Spoken L1 Language: Tuyuca [Electronic resource] / eds.: H. Hammarström, R. Forkel, M. Haspelmath ; Max Planck Institute for the Science of Human History // Glottolog. – Mode of access: https://glottolog.org/resource/languoid/id/tuyu1244. – Date of access: 13.10.2021.

[2]Language & Culture Archives: Tuyuca [Electronic resource] // SIL International. – Mode of access: https://www.sil.org/resources/search/language/tue. – Date of access: 13.10.2021.

[3]OLAC resources in and about the Tuyuca language [Electronic resource] // OLAC: Open Language Archives Community. – Mode of access: http://www.language-archives.org/language.php/tue. – Date of access: 13.10.2021.

ции. В некоторых случаях эта проблема может стать критической. Так, например, вызовом человечеству стала пандемия коронавируса, начавшаяся в 2020 г. Возникла экстренная необходимость проинформировать буквально каждого жители планеты о тех мерах предосторожности, которые следует предпринять в целях противостояния вирусу. Ответом на этот вызов стали решения, позволяющие автоматически переводить медицинские материалы на малые языки [3, 4]. В целом разработка эффективных автоматических переводчиков на малые языки является существенным шагом в направлении устранения информационного неравенства.

Во-вторых, ограничение на ресурсы мотивирует компьютерных лингвистов на поиск новых подходов к проблеме моделирования языков. В 2019 г. большое распространение получили модели, основанные на архитектуре нейронных сетей Transformer [5], например BERT [6]. Они оказались очень эффективными при решении разнообразных задач обработки естественного языка. Для подобных моделей применяется обучение с частичным привлечением учителя (semi-supervised learning) [2]: на первом этапе происходит обучение без учителя на большом корпусе неразмеченных данных, на втором – дообучение под конкретную задачу на небольшом количестве размеченных примеров, так называемый fine-tuning. Такой подход реализует идею переноса знаний (transfer learning): представление, полученное о языке в ходе первой стадии обучения, переиспользуется при решении конкретной задачи. Один из наиболее существенных недостатков указанного подхода заключается в том, что первая стадия обучения требует больших корпусов текстов. Например, в работах [7, 8] было показано, что корпус на несколько миллиардов слов не раскрывает весь потенциал модели BERT. Очевидно, что подобные корпуса недоступны для малых языков, не имеющих развитой литературной традиции и широкого представления в Интернете. Возникает отдельная задача моделирования малых языков, требующая собственных подходов. Наиболее популярное решение в этой области – адаптация имеющихся решений к работе в условиях дефицита ресурсов. Среди используемых подходов можно отметить две категории [1]:

– автоматическое расширение обучающей выборки (перенос аннотаций с текста на крупном языке на параллельный текст на моделируемом языке; автоматическое порождение данных, например, путем замены синонимов в имеющейся выборке);

– переиспользование знаний, полученных в ходе моделирования крупного языка (параллельное обучение модели многим языкам, сопоставление пространств векторных представлений слов на разных языках).

Обозначенные приемы имеют очевидные недостатки. Искусственная генерация похожих данных обедняет выборку. Совместное изучение нескольких языков эффективно, когда они родственны или типологически близки, но вызывает трудности при работе со своеобразным языком, например языком-изолятом. Представляет интерес разработка принципиально иных подходов, спроектированных специально для низкоресурсных языков.

В-третьих, автоматические модели могут сыграть важную роль в качестве инструмента сохранения языка, находящегося под угрозой исчезновения. Невозможно сохранить язык, на котором не желают разговаривать люди. Мотивация изучать малый язык имеет различную природу. Например, это может быть желание сохранить собственную идентичность и культуру. Существенным аргументом в пользу изучения языка может стать наличие доступных обучающих материалов, а также контента на разнообразные темы, в том числе художественных произведений. Соответствующие технологии позволяют упростить процесс изучения языка и облегчить создание текстов на целевом языке.

Актуальность проблемы сохранения малых языков, в свою очередь, неоднократно обсуждалась в литературе (см., например [9]). Можно отметить следующие соображения:

1. Для представителей соответствующих этносов родной язык может выступать одним из средств самоидентификации, возможно, основным, что становится особенно актуальным в эпоху глобализации. Более того, даже если в настоящее время общество отказывается от родного языка в пользу более крупного и развитого, со временем вновь может возникнуть спрос на культуру и язык предков.

2. Каждый язык является самоценным историко-культурным феноменом, в силу тесной связи языка и мышления являющимся своеобразным отражением уникальной философии народа,

сформировавшего этот язык, что, безусловно, значимо для таких наук, как этнография, этнология, культурная антропология.

3. Каждый язык служит дополнительным источником информации для лингвистических учений, причем каждая черта естественного языка дает дополнительную информацию об устройстве языка и речи вообще. Эти сведения используются в исследованиях человеческого мозга и мышления.

3. Освоение языка ребенком. Как было сказано выше, подходы к обработке малых языков в основном сводятся к адаптации идей, заложенных в построении крупных языковых моделей, подобных BERT, к языкам, располагающим малым числом доступных ресурсов для обучения. Речь идет либо об искусственном расширении обучающей выборки, либо о переносе знаний с модели крупного языка на модель низкоресурсного языка.

В этой связи любопытно задаться вопросом, как происходит освоение языка младенцем. Данный процесс универсален: вне зависимости от родного языка человек, не имеющий существенных особенностей в развитии, оказывается способен овладеть своим родным языком без каких-либо вспомогательных средств и методик [10, с. 2]. Можно ли в некотором приближении смоделировать этот процесс для обучения машины языку? Данный вопрос актуален не только с точки зрения психолингвистики, но и с точки зрения естественной обработки низкоресурсных языков: ребенок изучает родной язык, не имея каких-либо лингвистических сведений о нем, а также не опираясь на огромную обучающую выборку, т. е. использует тот объем ресурсов, который соизмерим с ресурсами, доступными для малоисследованного языка.

Вопрос языкового онтогенеза является открытым. Какие механизмы позволяют ребенку эффективно изучить родной язык, а также почему эти механизмы перестают функционировать после определенного возраста [10, с. 2], до сих пор неизвестно. Имеет смысл рассмотреть основные гипотезы, принятые в онтолингвистике, а также объективные факты, обнаруженные в ходе воспроизводимых экспериментов.

Стадии становления языка не раз описаны в литературе [11, с. 14–15; 12, с. 46–51; 13, с. 75–85]. Исследования показывают, что уже при рождении ребенок имеет определенную предрасположенность к языку, в частности способен узнавать по звучанию родной язык. Начиная с трех месяцев, у ребенка активируется функция запоминания слов, а в возрасте около пяти-семи месяцев ребенок начинает лепетать. В восемь месяцев у ребенка отдельные слова увязываются с объектами действительности – в первую очередь теми, которые движутся, т. е. выделяются на фоне неподвижной картинки. В возрасте около года человек начинает произносить полноценные слова, однако чаще всего они характеризуют ситуацию в целом и, возможно, его эмоциональное состояние. Такие слова получили название «голофразы». При игре с лошадкой высказанное «тпру» может обозначать и «лошадь», и «сани», и «садись», и «поедем», и «остановись». С возникновением представления о грамматике языка происходит разделение слов. На примере лошадки участники ситуации могут получить название «тпрунька», тогда как для действий будет выбрано другое слово. На более позднем этапе происходит выделение из контекста концепта лошадки, который связывается с ярлыком «лошадь». Когда размер словаря достигает приблизительно 100 слов, ребенок начинает их комбинировать. В возрасте около полутора лет у ребенка начинается интенсивное наращивание активного словарного запаса. Между 2 и 2,5 годами в речи увеличивается число используемых аффиксов и служебных слов. Примерно в три года происходит резкое становление грамматики: ребенок за короткий промежуток времени овладевает синтаксисом и морфологией языка в значительном объеме. Этот процесс осуществляется параллельно с пониманием маленьким человеком иерархической структуры мира вообще: как вещи состоят из деталей, так и речь состоит из отдельных слов и компонентов. На этом этапе ребенок формулирует для себя определенные правила языка, которые последовательно уточняются путем проб и ошибок.

Безусловно, дети не рождаются со знанием языка и навык владения языком не вырабатывается автоматически. Что именно является движущей силой, заставляющей ребенка учить язык, достоверно неизвестно. На этот счет существуют различные теории.

Бихевиористическая теория научения предполагает, что основными механизмами освоения языка являются подражание и подкрепление. Ребенок пробует подражать речи своих опекунов

и ориентируется на их реакцию («подкрепляя» ту или иную гипотезу). В целом данная гипотеза не позволяет полностью объяснить процесс становления речи в полной мере [14, с. 74–75].

Противоположная теория, базирующаяся на идеях Н. Хомского, заключается в том, что ребенок рождается сразу с сильными предпосылками к изучению языка: в его мозге наличествует специальная программа, которая обеспечивает усвоение грамматики любого языка, – так называемая универсальная грамматика. Ориентируясь на звучащую речь, ребенок адаптирует правила универсальной грамматики к своему родному языку, осуществляя тем самым обучение [14, с. 75–77]. Данная теория также подвергается критике: опыт детей, лишенных возможности изучать язык (выросших в лесу или подвергнутых жестокому обращению со стороны опекунов), показывает, что у них универсальная грамматика не реализуется никаким образом, хотя при восприятии языка как инстинкта более естественно было бы ожидать случайной реализации [15, с. 100]. Кроме того, в универсальной грамматике должны были быть некоторые универсальные элементы, свойственные всем языкам. Исследования показывают, что обнаружить подобные универсалии не представляется возможным [15, с. 93–94].

Когнитивная теория происхождения речи человека предполагает, что развитие речи обусловлено присущей ребенку с рождения способностью получать и обрабатывать информацию. В отличие от теории научения когнитивная теория предполагает, что движущим механизмом изучения языка является не подражание, а социальное взаимодействие [15, с. 74–75]. Согласно этой теории развитие языка не отличается от развития восприятия, памяти или мыслительных процессов.

Ребенок учится участвовать в разговоре, больше узнавая о языке и о том, как им пользоваться, тренируется планировать разговор на языке [11, с. 5–6]. Участие взрослого в изучении языка ребенком очень важно. Эксперименты показывают, что ребенок не в состоянии выучить язык, регулярно слушая телевизор. Взрослый, участвуя в разговоре с ребенком, следит за проявлениями его внимания и имеет возможность в зависимости от обстоятельств корректировать свое речевое поведение. Кроме того, речь взрослых при общении с ребенком имеет целый ряд специфических черт, призванных способствовать овладению языком: она медленнее, в ней строже соблюдаются правила грамматики, она больше нацелена на происходящее здесь и сейчас, в ней встречается больше повторов [13, с. 79].

4. Освоение языка и обучение с подкреплением. Изучение языка не происходит в отрыве от изучения мира – это два взаимно обусловленных процесса. Советский психолог Л. С. Выготский сближал факт развития значения слова с фактом развития сознания. Для него слово – это средство, которое отражает внешний мир в его связях и отношениях [12, с. 42]. На вышеуказанном материале наблюдаются параллели между познанием мира и изучением языка: ребенок одновременно приходит к пониманию иерархичности мира и иерархичности языка [13, с. 82].

Тесная связь между языком и мышлением, между изучением языка и изучением окружающего мира выводит задачу моделирования онтогенеза языка за рамки компьютерной лингвистики и приближает ее к задаче моделирования искусственного интеллекта.

Если рассматривать язык и мир с точки зрения семиотики Чарльза Пирса, то эти два сложных компонента могут быть представлены двумя моделями: моделью языка как многоуровневой системы знаков и моделью мира как системы произвольного вида. Процесс изучения в этом случае будет сводиться к изучению правил функционирования обеих моделей, а также выработке соотношения между компонентами двух моделей – «значениями» [16, с. 76].

Подобной структурой обладала, например, «программа, понимающая естественный язык» Терри Винограда [17]. Предложенный Виноградом агент SHRDLU действовал в «мире», состоящем из блоков различных форм и цветов, выполнял инструкции по переносу блоков, сохранял историю инструкций, мог изучать и оперировать новыми понятиями. Стоит отметить, что существенный пласт лексики был задан аксиоматически, как такового изучения естественного языка решение не предполагало. Проблему сопоставления информации, получаемой по двум каналам, один из которых – текст на естественном языке, затрагивает задача Visual Question Answering [18] и ее развитие – задача Embodied Question Answering [19]. Задача Visual Question Answering предполагает разработку программы, способной отвечать на вопросы, сформулированные на естественном языке и относящиеся к изображению, также поступающему на вход

системы: «Сколько лошадей видно?» – «Две лошади». В задаче Embodied Question Answering агент действует внутри своеобразного 3D-мира и для поиска ответа ему требуется совершать дополнительные действия (в частности, перемещаться в пространстве, чтобы увидеть предмет, к которому относится вопрос).

И бихевиористическая, и когнитивная гипотезы усвоения языка предполагают, что ребенок, пытаясь использовать речь, преследует некоторые цели (подражание или социальное взаимодействие). Эффективность коммуникации некоторым образом подкрепляется: была достигнута цель или нет. Этот процесс во многом напоминает процесс обучения с подкреплением – одно из направлений машинного обучения.

Подробный обзор работ, находящихся на стыке обучения с подкреплением и обработки естественного языка, рассмотрен в статье [20]. Многие из исследований затрагивают задачу следования инструкциям, представленным на некотором языке. Например, агент, разработанный в публикации [21], учится ориентироваться в 2D-мирах определенного вида и перемещаться в точку, заданную с помощью команды на английском языке (Reach the cell above the westernmost rock). Текстовые квесты представляют особый интерес, так как в этом случае на естественном языке показаны не только инструкции, но и текущее состояние окружения. Для построения подобных текстовых миров был разработан генератор TextWorld [22].

Необходимо отметить, что эффективность агента в отмеченных работах определяется по его способности достигать цель, а не по уровню овладения языком. Если ставить целью освоение естественного языка, следует продумать функцию награды как неотъемлемого компонента обучения с подкреплением. Эта функция может быть построена с помощью обратного обучения с подкреплением – метода, при котором заданы стратегии действия, рассматриваемые как «экспертные», «оптимальные», а цель обучения – построение корректной функции награды, которая в некотором роде объяснит действия «эксперта» [23].

Обучение с подкреплением продемонстрировало хорошие результаты при моделировании автоматических игроков в настольные игры. Например, нейронная сеть AlphaZero, разработанная компанией DeepMind, обучилась побеждать гроссмейстеров в шахматы, сеги и го [24]. При некотором допущении коммуникацию тоже можно рассматривать как игру, цель которой – получить от собеседника ожидаемый отклик. Текущий диалог может быть рассмотрен как состояние среды, возможные реплики агента – как потенциальные действия, а соответствие реплики собеседника ожидаемой – как функция оценки. Эта конфигурация соответствует теоретическому описанию обучения с подкреплением. Таким образом сформулирована задача, в которой агент методом проб и ошибок ищет стратегию оптимального использования естественного языка, что определенным образом перекликается с тем, как овладевает речью ребенок.

5. Существующие модели освоения языка. Вопрос моделирования онтогенеза языка исследовался в литературе. Как правило, целью таких исследований преимущественно является проверка психолингвистических гипотез о процессе изучения языка [25, с. 92].

Если говорить о процессе освоения словаря, то используемые подходы можно разбить на две категории. В первую категорию попадает изучение связи между словами и референтами, во вторую – постепенное уточнение смысла слов на базе полных предложений и некоторого, зачастую зашумленного, визуального представления [25, с. 94]. Второй подход применен, например, в работе [26], где в качестве «фона» используется множество меток, а результат обучения системы заключается в распределении вероятностей, что данное слово соотносится с данной меткой. Отдельного внимания удостаиваются синонимы и омонимы, которые нарушают отношение «один к одному». Авторы делают вывод, что для изучения смыслов слов в таком окружении не требуется особых предпосылок и специальных механизмов изучения, достаточно общих алгоритмов.

Изучение морфологии языка и обнаружение закономерностей словоизменения представляют собой задачи, которые могут быть решены с помощью нейронной сети прямого распространения [27]. Характер обучения соотносится с наблюдаемым процессом изучения морфологии детьми – так называемой U-образной кривой обучения: вначале ребенок запоминает конкретные формы, затем совершает обобщение, которое может приводить к ошибкам (например,

форма Past Simple в английском языке обычно образуется с помощью суффикса -ed, однако для ряда неправильных глаголов это утверждение неверно), и, наконец, выучивает исключения.

Моделирование освоения грамматики является более сложной задачей. В терминах теории универсальной грамматики решить ее можно только с помощью моделирования параметризованной грамматики, способной реализоваться в виде различных естественных языков. Публикация [25] – один из примеров такой модели: в ее основе лежит алгоритм, способный изучить категориальную грамматику.

Если же исходить из бихевиористической или когнитивной теорий, грамматику можно «извлечь» из конкретных примеров с помощью общих алгоритмов. Были разработаны модели, которые показали, что с помощью статистического анализа из массива высказываний, адресованных ребенку, можно извлечь информацию о структурах и закономерностях языка [25, с. 94; 28, с. 48]. В качестве материала исследования используются корпуса детской речи, например база CHILDES [29], содержащая коллекцию экспериментальных и наблюдательных данных об изучении языка детьми, а также о взаимодействии детей и взрослых.

Интерес представляет, например, модель MOSAIC (Model of Syntax Acquisition in Children) [30], обучающаяся на неразмеченном тексте и способная порождать высказывания, похожие на детские. Модель MOSAIC обучается в два этапа: на первом этапе строится иерархическая структура связанных ячеек, где каждая ячейка символизирует слово, а каждая связь представляет собой разницу между связанными словами. На втором этапе порождаются новые связи между словами со схожими контекстами. Этот шаг позволяет обнаруживать в речи закономерности и делать обобщения [31, с. 52]. Отмечается, что после обучения системы на корпусе CHILDES она достаточно эффективно имитировала особенности детской речи.

6. Архитектура модели. Безусловно, имитация онтогенеза осложнена тем, что само это явление изучено недостаточно. В литературе представлены опыты, имитирующие отдельные аспекты изучения языка ребенком. Обучающая выборка в этих экспериментах строится с ориентиром на объем реплик, доступных ребенку к некоторому возрасту, а результаты, в свою очередь, соотносятся с тем, насколько хорошо дети владеют речью. Таким образом удается сымитировать изучение лексики, словоизменения и грамматики.

Построение системы имитации онтогенеза – очень трудоемкая задача, особенно с учетом обозначенной выше взаимосвязи между изучением языка и изучением мира. Прежде всего следует определить те аспекты этого процесса, которыми можно пренебречь. В частности, педагоги отмечают, что процесс изучения жестового языка у глухих детей идентичен изучению звукового языка у слышащих детей. В частности, дети родителей, использующих жестовый язык, лепечут руками [13, с. 76]. Известно, что неслышащие люди, у которых повреждена зона Брока, имеют проблемы с высказываниями на языке жестов – точно так же, как люди, использующие звуковой язык, испытывают значительные затруднения на морфологическом и синтаксическом уровнях [13, с. 51]. У слабовидящих и слепых детей усвоение речи также осуществляется в процессе общения и по тем же закономерностям, что и у зрячих, хотя и с особенностями, вызванными объективными причинами [32]. Это позволяет предположить, что природа каналов связи с внешним миром не является существенной. Следовательно, при изучении принципиальной эффективности предлагаемого подхода допустимо выбрать произвольную модальность речи. В дальнейшем в настоящей работе будем ориентироваться на текстовое представление языка. Предполагается, что если предложенная система окажется эффективной, ее можно будет адаптировать для работы со звуковым представлением языка без существенных изменений в ядре архитектуры: либо непосредственно с помощью подготовленной обучающей выборки звуковых материалов, либо посредством обучаемого компонента, конвертирующего произнесенные фразы в текст.

В полноценную систему, результат работы которой будет использоваться как языковая модель, можно включить перечисленные выше разработки. Отметим, что они требуют значительной доработки. Так, например, в эксперименте с изучением связей между словами и «объектами» между двумя множествами строилось биективное отображение. Доступное ребенку предположение, что каждый объект имеет какое-то свое название, нередко рассматривается в психолингвистической литературе как один из двигателей процесса усвоения языка. Обрат-

ное предположение в корне неверно: многие слова выражают значительно более сложные концепции – от плохо формализуемых абстрактных понятий вроде цвета, запаха или ощущения до логических операторов, например кванторных местоимений «все», «каждый». Следует предусмотреть возможность изучения названий характеристик, численных мер или действий.

Модель нужно дополнить вспомогательными компонентами, в первую очередь имитацией «мира», объекты которого будут соотноситься со словами. В описанном выше эксперименте с изучением лексики «мир» (или «сцена» в терминах публикации) представлял собой неупорядоченный набор меток. В этом случае, однако, связи между объектами примитивны. Выбор более сложной конфигурации «мира» (например, блочного мира Терри Винограда или шахматной доски) позволит также моделировать лексику, соответствующую положению объектов, их атрибутов и взаимодействию между собой. Между тем это усложняет эксперимент, поскольку требует выработки схемы представления упомянутых концепций.

Другой модуль, который может быть добавлен, – компонент, оценивающий эффективность коммуникации. Как отмечают психологи, желание успешно взаимодействовать с окружающими также является одним из факторов, обусловливающих эффективное освоение языка ребенком. Этот фактор может некоторым образом порождать стремление совершить коммуникативный акт, а затем сопоставить воспринятый ответ с ожиданиями и соответствующим образом оценить состоявшийся диалог. Полученная оценка может быть использована как функция награды в терминах обучения с подкреплением.

Можно предложить и другие дополнительные компоненты, в частности средство поддержания контекста, позволяющее поддерживать тему диалога на протяжении нескольких реплик, или базу знаний, хранящую освоенную ранее информацию.

Попробуем соединить все описанные концепты воедино. Инициатором коммуникации выступает своеобразный «мозг» системы. Этот модуль в первую очередь имеет доступ ко всем компонентам, представляющим собой источник какой-либо информации: «миру», или «сцене», средству сохранения контекста, базе знаний. Выше упоминалось о взаимосвязи изучения языка и познания мира ребенком, однако, безусловно, следует понимать, что разработка модели «познания мира» слишком сложна. На этом этапе необходимо внести ряд упрощений. Рассматриваемый модуль можно представить как чат-бот-систему, которая принимает на вход реплику собеседника (в том числе нулевую, что будет соответствовать самостоятельной инициации беседы) и данные «окружающего мира», а затем генерирует реплику-ответ. Поскольку имитации сознания не требуется, можно воспользоваться одним из готовых алгоритмов, в частности нейронной сетью на базе архитектуры Transformer [33].

Появление нейронной сети сложной архитектуры вновь требует обучения на огромном массиве данных, недоступном для малого языка. Обойти эту проблему предполагается с помощью промежуточного метаязыка. В данном случае реплики будут проходить три стадии: реплика на исходном языке, представление реплики на метаязыке и векторное представление реплики с помощью Transformer. Аналогично идет обратный процесс: векторное представление реплики порождает представление на метаязыке, которое в свою очередь становится основой реплики на естественном языке. Transformer может быть использован как чат-бот, оперирующий метаязыком. Такого чат-бота можно обучить на материалах крупного языка (английского или русского) с помощью промежуточного слоя, ответственного за перевод с естественного языка на метаязык и обратно.

Структура метаязыка – предмет дальнейшего изучения. В целом он должен позволять выразить основные компоненты высказывания, такие как субъект, объект, предикат, атрибут. Подобные вопросы формального представления смысла проработаны в рамках формальной семантики [34].

Интерес вызывает представление лексики. В работе [26] всем словам, используемым в эксперименте, соответствовали метки на «сцене». В предлагаемой модели количество объектов «мира» (блочного мира Терри Винограда, шахматной доски) ограничено. Ожидается, что для лексики, соответствующей объектам «мира», будет найдено предметное соответствие.

Что касается остальных слов, можно рассмотреть следующий подход. По умолчанию предполагается, что неизвестным словам будут соответствовать некоторые нумерованные невиди-

мые объекты. Отношения между словами будут изучаться на примерах их употребления. В первую очередь речь идет о таком соотношении, как тождественность и различие. Более сложный подход предполагает изучение смыслов слов как результат применения логических функций к другим ранее изученным словам. Поиск подобного семантического представления для лексем широко представлен в работах А. Вежбицка и К. Годдард: авторы стремятся построить метаязык, который позволит определить все лексемы языка посредством малого набора «семантических примитивов» [35]. Важно, чтобы система имела возможность изучать не только существительные. Поэтому следует продумать способ представления «мира». Если хранить шахматную доску как двойной массив, элементы которого – фигуры, у системы не будет возможности изучить понятие «находится». Вместо этого следует использовать предикаты с аргументами, например «Находится(Конь, Доска)», «Соседствует(Конь, Слон)», «ОбладаетКачеством(Конь, белый)».

Следует понимать, что при переводе с метаязыка на естественный язык могут возникнуть непредвиденные трудности. Например, во многих языках коренных народов Южной Америки грамматикализирована эвиденциальность – указание на источник информации. В языке туюка туканской языковой семьи (его носители проживают на границе Бразилии и Колумбии) каждый глагол обязательно получает один из пяти суффиксов: визуального свидетельства (díiga apéwi – он играл в футбол, говорящий это видел), невизуального свидетельства (díiga apéti – он играл в футбол, говорящий слышал, как это происходило), вещественного свидетельства (díiga apéji – он играл в футбол, и говорящий обнаружил тому явные подтверждения), пересказывательности (díiga apéjigɨ – он играл в футбол, и говорящему об этом кто-то сообщил) и умозаключения (díiga apéhĩji – он играл в футбол, ибо у говорящего есть причины так считать) [36]. Если метаязык будет проектироваться на основе европейских языков, он не будет обязательно хранить значение эвиденциальности. Как результат, сформулированное на метаязыке высказывание «он играл в футбол», вероятнее всего, не получится корректно перевести на туюка: придется искусственно делать предположение об источнике информации и выбор суффикса окажется произвольным.

Конвертация метаязыка в малый естественный язык и обратно должна осуществляться с помощью компонентов, описанных в разд. 5. Грамматический модуль обучается строить структуру фразы на языке, лексический модуль тренируется подбирать подходящие лексемы, морфологический модуль совершенствуется в подборе верной словоформы. Обучение компонентов может происходить как на основе успешности коммуникации, так и по прецедентам с использованием в качестве обучающей выборки ответных реплик.

Заключение. В работе подробно обоснована актуальность моделирования малых языков, показана социальная значимость проблемы, польза ее решения для лингвистики, этнографии, этнологии и культурной антропологии, отмечено потенциальное развитие сферы обработки естественного языка.

В отличие от крупных языков, эффективно моделируемых с помощью решений на базе Transformer, малые языки не обладают достаточным количеством доступных ресурсов и потому не могут быть обработаны столь же успешно. В статье рассмотрена возможность моделирования малых языков с помощью имитации онтогенеза. Для этого приведен краткий обзор современного научного представления об онтогенезе языка. В полной мере данное явление и его движущие механизмы не изучены. Тем не менее накопленные сведения позволяют моделировать отдельные аспекты процесса освоения языка, в том числе опираясь на психолингвистические гипотезы. Можно говорить о связанном развитии психолингвистики и языкового моделирования: гипотезы о ходе процесса усвоения языка позволяют проектировать те или иные модели, а эффективность моделей в свою очередь позволяет характеризовать правдоподобность гипотез. Определенные результаты в этой сфере представлены в научной литературе: были разработаны системы, которые изучали лексический, морфологический и грамматический уровни языка, опираясь на корпус детской речи CHILDES и во многом имитируя усвоение соответствующих аспектов языка ребенком.

В контексте проблемы моделирования познания языка, которая сопряжена с познанием мира, можно вспомнить алгоритмы машинного обучения с подкреплением, мотивированные пси-

хическими моделями. Основная тенденция в современных работах, находящихся на стыке обработки языка и обучения с подкреплением, – обучение агента исполнению команд, отданных на естественном языке. Изучение языка в этих экспериментах является вспомогательной задачей. Однако успешную коммуникацию можно рассматривать и как цель, к которой стремится агент. В этом случае система обучения с подкреплением может преследовать цель изучения языка как основную.

На основании приведенных соображений в работе представлен концепт системы обработки языка, обучение которой происходит посредством моделирования онтогенеза. Выделены основные компоненты системы и принципы их взаимодействия.

Список использованных источников

1. A Survey on Recent Approaches for Natural Language Processing in Low-Resource Scenarios [Electronic resource] / M. A. Hedderich [et al.]. – 2020. – Mode of access: https://arxiv.org/abs/2010.12309. – Date of access: 12.10.2021.

2. Dai, A. M. Semi-supervised sequence learning [Electronic resource] / A. M. Dai, Q. V. Le // Proc. of the 28th Intern. Conf. on Neural Information Processing Systems. – 2015. – Vol. 2. – P. 3079–3087. https://doi.org/10.18653/v1/P17-1161

3. TICO-19: the translation initiative for Covid-19 [Electronic resource] / A. Anastasopoulos [et al.] // Proc. of the 1st Workshop on NLP for COVID-19 (Part 2) at EMNLP 2020. – Dec. 2020. – Mode of access: https://aclanthology.org/2020.nlpcovid19-2.5/. – Date of access: 12.10.2021. https://doi.org/10.18653/v1/2020.nlpcovid19-2.5

4. Enabling low-resource transfer learning across Covid-19 corpora by combining event-extraction and co-training / A. Spangher [et al.] // Proc. of the 1st Workshop on NLP for COVID-19 at ACL 2020. – July 2020. – Mode of access: https://aclanthology.org/2020.nlpcovid19-acl.4/. – Date of access: 12.10.2021.

5. Attention is all you need / A. Vaswani [et al.] // Proc. of the 31st Intern. Conf. on Neural Information Processing Systems, Long Beach, California, USA, 4–9 Dec. 2017. – Long Beach, 2017. – P. 6000–6010.

6. Качков, Д. И. Моделирование языка и двунаправленные представления кодировщиков: обзор ключевых технологий / Д. И. Качков // Информатика. – 2020. – Т. 17, № 4. – С. 61–72. https://doi.org/10.37661/1816-0301-2020-17-4-61-72

7. Cloze-driven pretraining of self-attention networks / A. Baevski [et al.] // Proc. of the 2019 Conf. on Empirical Methods in Natural Language Processing and the 9th Intern. Joint Conf. on Natural Language Processing (EMNLP-IJCNLP), Hong Kong, China, 3–7 Nov. 2019. – Hong Kong, 2019. – P. 5360–5369. https://doi.org/10.18653/v1/D19-1539

8. RoBERTa: A Robustly Optimized BERT Pretraining Approach [Electronic resource] / Y. Liu [et al.]. – 2019. – Mode of access: https://arxiv.org/abs/1907.11692. – Date of access: 12.10.2021.

9. Замятин, К. Как и зачем сохранять языки народов России / К. Замятин, А. Пасанен, Я. Саарикиви. – Хельсинки, 2012. – 181 с.

10. Meisel, J. M. First and Second Language Acquisition (Cambridge Textbooks in Linguistics) / J. M. Meisel. – Cambridge University Press, 2011. – 318 p.

11. Clark, E. V. First Language Acquisition / E. V. Clark. – Cambridge University Press, 2009. – 2nd ed. – 490 p.

12. Лурия, А. Р. Язык и сознание / А. Р. Лурия ; под ред. Е. Д. Хомской. – М. : Изд-во Моск. ун-та, 1979. – 320 с.

13. Бурлак, С. А. Происхождение языка. Факты, исследования, гипотезы / С. А. Бурлак. – М. : Альпина Диджитал, 2019. – 609 с.

14. Немов, Р. С. Общая психология в 3 т. Том II в 4 кн. Книга 4. Речь. Психические состояния : учебник и практикум для академического бакалавриата / Р. С. Немов. – 6-е изд., перераб. и доп. – М. : Юрайт, 2017. – 243 с.

15. Evans, V. The Language Myth Why Language Is Not an Instinct / V. Evans. – Cambridge University Press, 2014. – 314 p.

16. Пирс, Ч. С. Принципы философии : в 2 т. / Ч. С. Пирс ; пер. с англ. В. В. Кирющенко, М. В. Колопотина. – СПб. : Санкт-Петербургское философское общество, 2001. – Т. 2. – 313 с.

17. Виноград, Т. Программа, понимающая естественный язык / Т. Виноград. – М. : Мир, 1976. – 296 с.

18. VQA: visual question answering / S. Antol [et al.] // IEEE Intern. Conf. on Computer Vision (ICCV). – Santiago, Chile, 2015. – P. 2425–2433. https://doi.org/10.1109/ICCV.2015.279

19. Embodied question answering / A. Das [et al.] // Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, UT, USA, 18–23 June 2018. – Salt Lake City, 2018. – P. 1–10.

20. A survey of reinforcement learning informed by natural language / J. Luketina [et al.] // Proc. of the Twenty-Eighth Intern. Joint Conf. on Artificial Intelligence, Macao, China, 10–16 Aug. 2019. – Macao, 2019. – P. 6309–6317. https://doi.org/10.24963/ ijcai.2019/880

21. Janner, M. Representation learning for grounded spatial reasoning / M. Janner, K. Narasimhan, R. Barzilay // Transactions of the Association for Computational Linguistics. – 2018. – Vol. 6. – P. 49–61. https://doi.org/10.1162/tacl_a_00004

22. Côté, M.-A. TextWorld: A learning environment for text-based games / M.-A. Côté ; T. Cazenave, A. Saffidine, N. Sturtevant (eds.) // Computer Games. CGW 2018. Communications in Computer and Information Science. – Cham : Springer, 2018. – Vol. 1017. – P. 41–75. https://doi.org/10.1007/978-3-030-24337-1_3

23. Arora, S. A survey of inverse reinforcement learning: Challenges, methods and progress [Electronic resource] / S. Arora, P. Doshi // Artificial Intelligence. – 2021. – Vol. 297. – Mode of access: https://arxiv.org/abs/1806.06877. – Date of access: 12.10.2021. https://doi.org/10.1016/j.artint.2021.103500

24. A general reinforcement learning algorithm that masters chess, shogi, and go through self-play / D. Silver [et al.] // Science. – 2018. – Vol. 362, no. 6419. – P. 1140–1144. https://dx.doi.org/10.1126%2Fscience.aar6404

25. Freudenthal, D. Computational models of language development / D. Freudenthal, A. Alishahi ; P. J. Brooks, V. Kempe (eds.) // Encyclopedia of Language Development. – 1st ed. – SAGE Publications Inc., 2014. – P. 92–96.

26. Fazly, A. A probabilistic computational model of cross-situational word learning / A. Fazly, A. Alishahi, S. Stevenson // Cognitive Science. – 2010. – Vol. 34, iss. 6. – P. 1017–1063. https://doi.org/10.1111/j.1551-6709.2010.01104.x

27. Christiansen, M. H. Connectionist natural language processing: the state of the art / M. H. Christiansen, N. Chater // Cognitive Science. – 1999. – Vol. 23, iss. 4. – P. 417–437. https://doi.org/10.1207/s15516709cog2304_2

28. Buttery, P. J. Computational models for first language acquisition / P. J. Buttery // Technical Report UCAM-CL-TR-675. – University of Cambridge, 2006. – Mode of access: https://www.cl. cam.ac.uk/techreports/UCAM-CL-TR-675.pdf. – Date of access: 21.03.2021.

29. MacWhinney, B. The CHILDES Project: Tools for Analyzing Talk: Transcription Format and Programs (3rd ed.) / B. MacWhinney. – Lawrence Erlbaum Associates Publishers, 2000.

30. Jones, G. A process model of children's early verb use / G. Jones, F. Gobet, J. M. Pine // Proc. of the 22th Annual Conf. of the Cognitive Science Society, Philadelphia, PA, 13–15 Aug. 2000. – Philadelphia, 2000. – P. 723–728.

31. Alishahi, A. Computational Modeling of Human Language Acquisition / A. Alishahi. – Morgan & Claypool, 2010. – 107 p.

32. Andersen, E. S. The impact of input: language acquisition in the visually impaired / E. S. Andersen, A. Dunlea, L. Kekelis // First Language. – 1993. – Vol. 13, no. 37. – P. 23–49. https://doi.org/10.1177/014272379301303703

33. Vlasov, V. Dialogue Transformers [Electronic resource] / V. Vlasov, J. E. M. Mosig, A. Nicho. – 2019. – Mode of access: https://arxiv.org/abs/1910.00486. – Date of access: 12.10.2021.

34. Андреев, А. В. Введение в формальную семантику : учеб. пособие / А. В. Андреев, О. А. Митрофанова, К. В. Соколов. – СПб. : СПбГУ, 2014. – 88 с.

35. Goddard, C. The search for the shared semantic core of all languages / C. Goddard ; C. Goddard, A. Wierzbicka (eds.) // Meaning and Universal Grammar – Theory and Empirical Findings. – Amsterdam : John Benjamins, 2002. – Vol. I. – P. 5–40.

36. Barnes, J. Evidentials in the Tuyuca Verb / J. Barnes // Intern. J. of American Linguistics. – 1984. – Vol. 50, no. 3. – P. 255–271.

References

1. Hedderich M. A., Lange L., Adel H., Strötgen J., Klakow D. *A Survey on Recent Approaches for Natural Language Processing in Low-Resource Scenarios*, 2020. Available at: https://arxiv.org/abs/2010.12309 (accessed 12.10.2021).

2. Dai A. M., Le Q. V. Semi-supervised sequence learning. *Proceedings of the 28th International Conference on Neural Information Processing Systems*, 2015, vol. 2, pp. 3079–3087. https://doi.org/10.18653/v1/P17-1161

3. Anastasopoulos A., Cattelan A., Dou Z.-Y., Federico M., Federman C., …, Tur S. TICO-19: the translation initiative for Covid-19. *Proceedings of the 1st Workshop on NLP for COVID-19 (Part 2) at EMNLP 2020*. December 2020. Available at: https://aclanthology.org/2020.nlpcovid19-2.5/ (accessed 12.10.2021). https://doi.org/10.18653/v1/2020.nlpcovid19-2.5

4. Spangher A., Peng N., May J., Ferrara E. Enabling low-resource transfer learning across Covid-19 corpora by combining event-extraction and co-training. *Proceedings of the 1st Workshop on NLP for COVID-19 at ACL 2020*. July 2020. Available at: https://aclanthology.org/2020.nlpcovid19-acl.4/ (accessed 12.10.2021).

5. Vaswani A., Shazeer N., Parmar N., Uszkoreit J., Jones L., …, Polosukhin I. Attention is all you need. *Proceedings of the 31st International Conference on Neural Information Processing Systems, Long Beach, California, USA, 4–9 December 2017*. Long Beach, 2017, pp. 6000–6010.

6. Kachkou D. I. Language modeling and bidirectional coders representations: an overview of key technologies. Informatika [*Informatics*], 2020, vol. 17, no. 4, pp. 61−72 (In Russ.). https://doi.org/10.37661/1816-0301-2020-17-4-61-72

7. Baevski A., Edunov S., Liu Y., Zettlemoyer L., Auli M. Cloze-driven pretraining of self-attention networks. *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), Hong Kong, China, 3–7 November 2019.* Hong Kong, 2019, pp. 5360–5369. https://doi.org/10.18653/v1/D19-1539

8. Liu Y., Ott M., Goyal N., Du J., Joshi M., …, Stoyanov V. *RoBERTa: A Robustly Optimized BERT Pretraining Approach*, 2019. Available at: https://arxiv.org/abs/1907.11692 (accessed 12.10.2021).

9. Zamyatin K., Pasanen A., Saarikivi Ya. Kak i zachem sohranyat' yazyki narodov Rossii. *How and Why to Save Languages of Ethnic Groups in Russia*. Helsinkim, 2012, 181 p. (In Russ.).

10. Meisel J. M. *First and Second Language Acquisition (Cambridge Textbooks in Linguistics)*. Cambridge University Press, 2011, 318 p.

11. Clark E. V. *First Language Acquisition*. Cambridge University Press, 2nd ed., 2009, 490 p.

12. Luriya A. R. Yazyk i soznanie. *Language and Conscience*. In Homskaya E. D. (ed.). Moscow, Izdatel'stvo Moskovskogo universtiteta, 1979, 320 p. (In Russ.).

13. Burlak S. A. Proishozhdenie yazyka. Fakty, issledovaniya, gipotezy. *Origin of the language. Facts, Researches, Hypothesis*. Moscow, Alpina digital, 2019, 609 p. (In Russ.).

14. Nemov R. S. Obshchaya psihologiya. *General Psychology.* Vol. 2, kniga 4. Rech'. Psihicheskie sostoyaniya: uchebnik i praktikum dlya akademicheskogo bakalavriata. *Speech. Psychological States: Textbook and Workshop for Bachelors*. 6th ed., Moscow, Yurait, 2017, 243 p. (In Russ.).

15. Evans V. *The Language Myth Why Language Is Not an Instinct*. Cambridge University Press, 2014, 314 p.

16. Peirce C. S. *Collected Papers of Charles Sanders Peirce, Volumes I and II: Principles of Philosophy and Elements of Logic*. Belknap Press, 1932, vol. II, 535 p.

17. Winograd T. Programma, ponimaushchaya estestvennyj yazyk. *Understanding Natural Language*. Moscow, Mir, 1976, 296 p. (In Russ.).

18. Antol S., Agrawal A., Lu J., Mitchell M., Batra D., …, Parikh D. VQA: visual question answering. *IEEE International Conference on Computer Vision (ICCV)*. Santiago, Chile, 2015, pp. 2425–2433. https://doi.org/10.1109/ICCV.2015.279

19. Das A., Datta S., Gkioxari G., Lee S., Parikh D., Batra D. Embodied question answering. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, UT, USA, 18–23 June 2018.* Salt Lake City, 2018, pp. 1–10.

20. Luketina J., Nardelli N., Farquhar G., Foerster J., Andreas J., …, Rocktäschel T. A survey of reinforcement learning informed by natural language. *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, Macao, China, 10–16 August 2019*. Macao, 2019, pp. 6309–6317. https://doi.org/10.24963/ijcai.2019/880

21. Janner M., Narasimhan K., Barzilay R. Representation learning for grounded spatial reasoning. *Transactions of the Association for Computational Linguistics*, 2018, vol. 6, pp. 49–61. https://doi.org/10.1162/tacl_a_00004

22. Côté M.-A., Kádár Á., Yuan X., Kybartas B., Barnes T., …, Trischler A. TextWorld: A learning environment for text-based games. *Computer Games. CGW 2018. Communications in Computer and Information Science*, Cham, Springer, 2018, vol. 1017, pp. 41–75. https://doi.org/10.1007/978-3-030-24337-1_3

23. Arora S., Doshi P. A survey of inverse reinforcement learning: Challenges, methods and progress. *Artificial Intelligence*, 2021, vol. 297. Available at: https://arxiv.org/abs/1806.06877 (accessed 12.10.2021). https://doi.org/10.1016/j.artint.2021.103500

24. Silver D., Hubert T., Schrittwieser J., Antonoglou I., Lai M., …, Hassabis D. A general reinforcement learning algorithm that masters chess, shogi, and go through self-play. *Science*, 2018, vol. 362, no. 6419, pp. 1140–1144. https://dx.doi.org/10.1126%2Fscience.aar6404

25. Freudenthal D., Alishahi A. Computational models of language development. *Encyclopedia of Language Development*. In Brooks P. J., Kempe V. (eds.). 1st ed., SAGE Publications Inc., 2014, pp. 92–96.

26. Fazly A., Alishahi A., Stevenson S. A probabilistic computational model of cross-situational word learning. *Cognitive Science*, 2010, vol. 34, iss. 6, pp. 1017–1063. https://doi.org/10.1111/j.1551-6709.2010.01104.x

27. Christiansen M. H., Chater N. Connectionist natural language processing: the state of the art. *Cognitive Science*, 1999, vol. 23, iss. 4, pp. 417–437. https://doi.org/10.1207/s15516709cog2304_2

28. Buttery P. J. Computational models for first language acquisition. *Technical Report UCAM-CL-TR-675*, University of Cambridge, 2006. Available at: https://www.cl.cam.ac.uk/techreports/ UCAM-CL-TR-675.pdf (accessed 21.03.2021).

29. MacWhinney, B. *The CHILDES Project: Tools for Analyzing Talk: Transcription Format and Programs*. 3rd ed., Lawrence Erlbaum Associates Publishers, 2000.

30. Jones G., Gobet F., Pine J. M. A process model of children's early verb use. *Proceedings of the 22th Annual Conference of the Cognitive Science Society, Philadelphia, PA, 13–15 August 2000*. Philadelphia, 2000, pp. 723–728.

31. Alishahi A. *Computational Modeling of Human Language Acquisition*. Morgan & Claypool, 2010, 107 p.

32. Andersen E. S, Dunlea A., Kekelis L. The impact of input: language acquisition in the visually impaired. *First Language*, 1993, vol. 13, no. 37, pp. 23–49. https://doi.org/10.1177/014272379301303703

33. Vlasov V., Mosig J. E. M., Nicho A. *Dialogue Transformers*, 2019. Available at: https://arxiv.org/abs/1910.00486 (accessed 12.10.2021).

34. Andreev A. V., Mitrofanova O. A., Sokolov K. V. Vvedenie v formal'nuyu semantiky: uchebnoe posobie. *Introduction Into Formal Semantics: Handbook*. Saint-Petersburg, Saint-Petersburg State University, 2014, 88 p. (In Russ.).

35. Goddard C. The search for the shared semantic core of all languages. In Goddard C., Wierzbicka A. (eds.). *Meaning and Universal Grammar – Theory and Empirical Findings*. Amsterdam, John Benjamins, 2002, vol. I, pp. 5–40.

36. Barnes J. Evidentials in the tuyuca verb. *International Journal of American Linguistics*, 1984, vol. 50, no. 3, pp. 255–271.

Информация об авторе

Качков Дмитрий Ильич, аспирант кафедры многопроцессорных систем и сетей факультета прикладной математики и информатики, Белорусский государственный университет.
E-mail: dmitriydikanskiy@gmail.com

Information about the author

Dzmitry I. Kachkou, Postgraduate Student of Department of Multiprocessor Systems and Networks of the Faculty of Applied Mathematics and Informatics, Belarusian State University.
E-mail: dmitriydikanskiy@gmail.com

Правила для авторов

Редакция журнала «Информатика» просит авторов руководствоваться приведенными ниже правилами.

I. Статьи принимаются в редакцию через электронную систему подачи по адресу http://inf.grid.by в формате файлов текстовых редакторов Microsoft Word. Объем оригинальной статьи – от 8 до 16 стр., включая рисунки, таблицы и достаточное количество наиболее актуальных ссылок; объем обзорной статьи – от 16 до 32 стр., включая все основные ссылки. Текст набирается с переносами, шрифт Times New Roman 11 пт, интервал между строками одинарный, абзацный отступ 0,5 см, поля по 2,5 см со всех сторон.

Материал статьи должен быть четко структурированным: Введение; основные разделы, в которых изложены цели и задачи, методы, результаты; Заключение (выводы).

II. Статьи о результатах работ, проведенных в научных учреждениях, должны иметь разрешение на публикацию (сопроводительное письмо за подписью руководителя или выписку из заседания ученого совета, отдела или кафедры, акт экспертизы).

III. Статьи в обязательном порядке должны включать аннотацию, ключевые слова, список литературы, информацию об авторах на русском и английском языках.

На титульной странице располагаются следующие метаданные:

1. Индекс по универсальной десятичной классификации (УДК); *на русском и английском языках* тип статьи (оригинальная или обзорная), название статьи, инициалы и фамилии всех авторов, полное наименование учреждений, где работают авторы, с указанием почтового адреса, при наличии указывается ученая степень и ORCID, e-mail ответственного лица.

2. Аннотация (Abstract) объемом 150–250 слов в оригинальной статье должна быть структурирована отдельными подразделами: Цели, Методы, Результаты, Заключение, а также максимально характеризовать содержательную часть рукописи. Сюда не следует включать впервые введенные термины, аббревиатуры (за исключением общеизвестных), ссылки на литературу.

3. Ключевые слова (Keywords) – наиболее значимые слова или словосочетания по теме работы, отражающие специфику темы, объекты и результаты исследования; перечень ключевых слов должен содержать 5–10 слов.

4. В разделе Благодарности (Acknowledgements) указываются все источники финансирования исследования, а также благодарности людям, которые участвовали в работе над статьей.

5. Автор обязан уведомить редакцию о реальном или потенциальном конфликте интересов, включив информацию в раздел Конфликт интересов (Conflict of interest).

6. Формулы, рисунки, таблицы в статье нумеруются в соответствии с порядком их упоминания в тексте. Ссылки на рисунки и таблицы в тексте обязательны. Рисунки должны быть выполнены с хорошим разрешением в масштабе, позволяющем четко различать надписи и обозначения. Цветные иллюстрации печатаются только в том случае, когда это необходимо для понимания излагаемого материала. Подрисуночные подписи с расшифровкой всех позиций, представленных на рисунке, и названия таблиц набираются шрифтом гарнитуры основного текста размером 9 пт. Перевод подрисуночной подписи и пояснений к рисунку, а также перевод названия таблицы, заголовки строк или столбцов располагаются курсивом после русскоязычной версии.

7. Набор формул выполняется в формульном редакторе Microsoft Equation или Math Type. Прямым шрифтом набираются: греческие и русские буквы; математические символы (sin, lg, ∞); символы химических элементов (C, Cl, $CHCl_3$); цифры (римские и арабские); индексы (верхние и нижние), являющиеся сокращениями слов. Курсивом набираются латинские буквы, символы физических величин (в том числе и в индексе).

8. Список использованной литературы оформляется в соответствии с требованиями Высшей аттестационной комиссии Республики Беларусь (ГОСТ 7.5–2008). Номер литературной ссылки в тексте дается порядковым номером в квадратных скобках. Ссылаться на неопубликованные работы не допускается.

10. Отдельно оформляется References со следующей структурой: авторы (транслитерация), транслитерированное название монографии, *Перевод названия монографии на английский язык*. Выходные данные с обозначениями на английском языке. От транслитераций названий статей можно отказаться.

Ссылки на учебно-методическую литературу, ГОСТы, авторефераты, статистические отчеты в список не включаются, а оформляются в виде сносок (с подробными рекомендациями можно ознакомиться на сайте журнала в разделе Правила для авторов).

11. В разделе Информация об авторах (Information about the authors) приводятся ФИО авторов полностью, ученая степень, звание, должность, название организации, ORCID (при наличии).

IV. Все поступающие в редакцию рукописи проходят предварительную проверку на соответствие Правилам для авторов. Статья может быть возвращена автору на доработку с просьбой устранить недостатки или дополнить информацию. После проверки на соответствие правилам статья направляется рецензенту с указанием сроков рецензирования.

V. При наличии замечаний рецензента автору предоставляется определенное время на доработку рукописи. Статьи, направляемые на доработку, должны быть возвращены в исправленном виде с ответами на все замечания. Окончательное решение о публикации или отклонении рукописи принимается редколлегией журнала. При положительном заключении рецензента статья передается редактору для подготовки к печати. Редакция оставляет за собой право на редакционные изменения, не искажающие основное содержание статьи.

VI. Редакция журнала предоставляет возможность первоочередного опубликования статей, представленных лицами, которые осуществляют послевузовское обучение (аспирантура, докторантура, соискательство) в год завершения обучения.

VII. Авторы несут ответственность за направление в редакцию статей, уже опубликованных ранее или принятых к публикации другими изданиями.

ИНДЕКСЫ

00827
для индивидуальных
подписчиков

008272
для предприятий
и организаций

ИНФОРМАТИКА ▪ INFORMATICS ТОМ ▪ VOL. 19 1 | 2022