Computer Science Journal of Moldova, vol.**26**, no.3(78), 2018

A Short Introduction to Program Algebra with Instructions for Boolean Registers

Jan A. Bergstra, Cornelis A. Middelburg

Abstract

A parameterized algebraic theory of instruction sequences, objects that represent the behaviours produced by instruction sequences under execution, and objects that represent the behaviours exhibited by the components of the execution environment of instruction sequences is the basis of a line of research in which issues relating to a wide variety of subjects from computer science have been rigorously investigated thinking in terms of instruction sequences. In various papers that belong to this line of research, use is made of an instantiation of this theory in which the basic instructions are instructions to read out and alter the content of Boolean registers and the components of the execution environment are Boolean registers. In this paper, we give a simplified presentation of the most general such instantiated theory.

Keywords: program algebra, thread algebra, thread-service interaction, Boolean register.

MSC 2010: 68Q05, 68Q55, 03B70.

1 Introduction

We are carrying out a line of research in which issues relating to a wide variety of subjects from computer science are rigorously investigated thinking in terms of instruction sequences (see e.g. [1]). The groundwork for this line of research is the combination of an algebraic theory of single-pass instruction sequences, called program algebra, and an algebraic theory of mathematical objects that represent the behaviours

produced by instruction sequences under execution, called basic thread algebra, extended to deal with the interaction between instruction sequences under execution and components of their execution environment (see e.g. [2]). This groundwork is parameterized by a set of basic instructions and a set of objects that represent the behaviours exhibited by the components of the execution environment.

In various papers that have resulted from this line of research, use is made of an instantiation of this theory in which certain instructions to read out and alter the content of Boolean registers are taken as basic instructions and Boolean registers are taken as the components of the execution environment (see [3]–[8]). In the current paper, we give a simplified presentation of the instantiation in which all possible instructions to read out and alter the content of Boolean registers are taken as basic instructions.

In the papers referred to above, the rationale for taking certain instructions to read out and alter the content of Boolean registers as basic instructions is that the instructions concerned are sufficient to compute each function on bit strings of any fixed length by a finite instruction sequence. However, shorter instruction sequences may be possible if certain additional instructions to read out and alter the content of Boolean registers are taken as basic instructions (see [9]). That is why we opted for the most general instantiation.

Both program algebra and basic thread algebra were first presented in [10].[1] An extension of basic thread algebra to deal with the interaction between instruction sequences under execution and components of their execution environment, called services, was presented for the first time in [11]. A substantial re-design of this extension was first presented in [12]. The presentation of both extensions is rather involved because they are parameterized and owing to this cover a generic set of basic instructions and a generic set of services. In the current paper, a much less involved presentation is obtained by covering only the case where the basic instructions are instructions to read out and alter the content of Boolean registers and the services are Boolean registers.

[1] In that paper and the first subsequent papers, basic thread algebra was introduced under the name basic polarized process algebra.

This paper is organized as follows. First, we introduce program algebra (Section 2) and basic thread algebra (Section 3) and extend their combination to make precise which behaviours are produced by instruction sequences under execution (Section 4). Next, we present the instantiation of the resulting theory in which all possible instructions to read out and alter Boolean registers are taken as basic instructions (Section 5), introduce an algebraic theory of Boolean register families (Section 6), and extend the combination of the theories presented in the two preceding sections to deal with the interaction between instruction sequences under execution and Boolean registers (Section 7). Then, we formalize in the setting of the resulting theory what it means that a given instruction sequence computes a given partial function from $\mathbb{B}^n$ to $\mathbb{B}^m$ ($n, m \in \mathbb{N}$) (Section 8) and give a survey of uses for the resulting theory (Section 9). Finally, we make some concluding remarks (Section 10).

The following should be mentioned in advance. The set $\mathbb{B}$ is a set with two elements whose intended interpretations are the truth values *false* and *true*. As is common practice, we represent the elements of $\mathbb{B}$ by the bits 0 and 1. In line with generally accepted conventions, we use terminology based on identification of the elements of $\mathbb{B}$ with their representation where appropriate. For example, the elements of $\mathbb{B}^n$ are loosely called bit strings of length n.

In this paper, some familiarity with algebraic specification is assumed. The relevant notions are explained in handbook chapters and books on algebraic specification, e.g. [13]–[16].

This paper is to a large extent a compilation of material from several earlier publications. Various examples, various explanatory remarks, and the axioms from Section 7 do not occur in earlier publications.

2 Program Algebra

In this section, we present PGA (ProGram Algebra). The starting-point of PGA is the perception of a program as a single-pass instruction sequence, i.e. a possibly infinite sequence of instructions of which each instruction is executed at most once and can be dropped after it has

been executed or jumped over. The concepts underlying the primitives of program algebra are common in programming, but the particular form of the primitives is not common. The predominant concern in the design of PGA has been to achieve simple syntax and semantics, while maintaining the expressive power of arbitrary finite control.

It is assumed that a fixed but arbitrary set $\mathcal{A}$ of *basic instructions* has been given. $\mathcal{A}$ is the basis for the set of instructions that may occur in the instruction sequences considered in PGA. The intuition is that the execution of a basic instruction may modify a state and must produce the Boolean value 0 or 1 as reply at its completion. The actual reply may be state-dependent.

In applications of PGA, the instructions taken as basic instructions vary, in effect, from instructions relating to unbounded counters, unbounded stacks or Turing tapes through instructions relating to Boolean registers or natural number registers to machine language instructions of actual computers.

The set of instructions of which the instruction sequences considered in PGA are composed is the set that consists of the following elements:

- for each $a \in \mathcal{A}$, a *plain basic instruction* a;
- for each $a \in \mathcal{A}$, a *positive test instruction* $+a$;
- for each $a \in \mathcal{A}$, a *negative test instruction* $-a$;
- for each $l \in \mathbb{N}$, a *forward jump instruction* $\#l$;
- a *termination instruction* $!$.

We write $\mathcal{I}$ for this set. The elements from this set are called *primitive instructions*.

Primitive instructions are the elements of the instruction sequences considered in PGA. On execution of such an instruction sequence, these primitive instructions have the following effects:

- the effect of a positive test instruction $+a$ is that basic instruction a is executed and execution proceeds with the next primitive

instruction if 1 is produced and otherwise the next primitive instruction is skipped and execution proceeds with the primitive instruction following the skipped one — if there is no primitive instruction to proceed with, inaction occurs;

- the effect of a negative test instruction $-a$ is the same as the effect of $+a$, but with the role of the value produced reversed;
- the effect of a plain basic instruction a is the same as the effect of $+a$, but execution always proceeds as if 1 is produced;
- the effect of a forward jump instruction $\#l$ is that execution proceeds with the lth next primitive instruction — if l equals 0 or there is no primitive instruction to proceed with, inaction occurs;
- the effect of the termination instruction ! is that execution terminates.

Inaction occurs if no more basic instructions are executed, but execution does not terminate.

A plain basic instruction a is generally used in the case of a basic instruction a that modifies a state and a positive test instruction $+a$ or a negative test instruction $-a$ is generally used in the case of a basic instruction a that does not modify a state. However, there are no rules prescribing such use.

PGA has one sort: the sort **IS** of *instruction sequences*. We make this sort explicit to anticipate the need for many-sortedness later on. To build terms of sort **IS**, PGA has the following constants and operators:

- for each $u \in \mathcal{I}$, the *instruction* constant $u :\to \mathbf{IS}$;
- the binary *concatenation* operator $\_ ; \_ : \mathbf{IS} \times \mathbf{IS} \to \mathbf{IS}$;
- the unary *repetition* operator $\_^\omega : \mathbf{IS} \to \mathbf{IS}$.

Terms of sort **IS** are built as usual in the one-sorted case. We assume that there are infinitely many variables of sort **IS**, including X, Y, Z. We use infix notation for concatenation and postfix notation for repetition. Taking these notational conventions into account, the syntax of

closed PGA terms (of sort **IS**) can be defined in Backus-Naur style as follows:

$$CT_{\mathbf{IS}} ::= a \mid +a \mid -a \mid \#l \mid ! \mid (CT_{\mathbf{IS}} \,;\, CT_{\mathbf{IS}}) \mid ({CT_{\mathbf{IS}}}^{\omega}) \;,$$

where $a \in \mathcal{A}$ and $l \in \mathbb{N}$.[2]

Throughout the paper, we generally omit grouping parentheses if they can be unambiguously added or they are unnecessary because it is axiomatized that the operator concerned stands for an associative operation.

A PGA term in which the repetition operator does not occur is called a *repetition-free* PGA term. A PGA term that is not repetition-free is said to be a PGA term that *has a repeating part*.

One way of thinking about closed PGA terms is that they represent non-empty, possibly infinite sequences of primitive instructions with finitely many distinct suffixes. The instruction sequence represented by a closed term of the form $t;t'$ is the instruction sequence represented by t concatenated with the instruction sequence represented by t'.[3] The instruction sequence represented by a closed term of the form t^{ω} is the instruction sequence represented by t concatenated infinitely many times with itself. A closed PGA term represents a finite instruction sequence if and only if it is a closed repetition-free PGA term.

A simple example of a closed PGA term is

$$(-a \,;\, (\#3 \,;\, (b \,;\, !)))^{\omega} \;.$$

On execution of the infinite instruction sequence denoted by this term, first the basic instruction a is executed repeatedly until its execution produces the reply 1, next the basic instruction b is executed, and after that execution terminates. Because $(X \,;Y) \,;Z = X \,;(Y \,;Z)$ is an axiom of PGA (see below), we could have written $(-a \,;\#3 \,;b \,;!)^{\omega}$ instead of $(-a \,;(\#3 \,;(b \,;!)))^{\omega}$ above.

The axioms of PGA are given in Table 1. In this table, u, $u_1, \ldots, u_k$

[2] We use $CT_{\mathbf{S}}$, where **S** is a sort, as nonterminal standing for closed terms of sort **S**.

[3] The concatenation of an infinite sequence with a finite or infinite sequence yields the former sequence.

Table 1. Axioms of PGA

$(X\,;Y)\,;Z = X\,;(Y\,;Z)$	PGA1
$(X^n)^\omega = X^\omega$	PGA2
$X^\omega\,;Y = X^\omega$	PGA3
$(X\,;Y)^\omega = X\,;(Y\,;X)^\omega$	PGA4
$\#k{+}1\,;u_1\,;\ldots\,;u_k\,;\#0 = \#0\,;u_1\,;\ldots\,;u_k\,;\#0$	PGA5
$\#k{+}1\,;u_1\,;\ldots\,;u_k\,;\#l = \#l{+}k{+}1\,;u_1\,;\ldots\,;u_k\,;\#l$	PGA6
$(\#l{+}k{+}1\,;u_1\,;\ldots\,;u_k)^\omega = (\#l\,;u_1\,;\ldots\,;u_k)^\omega$	PGA7
$\#l{+}k{+}k'{+}2\,;u_1\,;\ldots\,;u_k\,;(v_1\,;\ldots\,;v_{k'+1})^\omega =$ $\#l{+}k{+}1\,;u_1\,;\ldots\,;u_k\,;(v_1\,;\ldots\,;v_{k'+1})^\omega$	PGA8

and $v_1,\ldots,v_{k'+1}$ stand for arbitrary primitive instructions from $\mathcal{I}$, k, k', and l stand for arbitrary natural numbers from $\mathbb{N}$, and n stands for an arbitrary natural number from $\mathbb{N}_1$.[4] For each $n \in \mathbb{N}_1$, the term t^n, where t is a PGA term, is defined by induction on n as follows: $t^1 = t$, and $t^{n+1} = t\,;t^n$.

Some simple examples of equations derivable from the axioms of PGA are

$$(a\,;b)^\omega\,;c = a\,;(b\,;a)^\omega\ ,$$
$$+a\,;(b\,;(-c\,;\#2\,;!)^\omega)^\omega = +a\,;b\,;(-c\,;\#2\,;!)^\omega\ .$$

Closed PGA terms t and t' represent the same instruction sequence iff $t = t'$ is derivable from PGA1–PGA4. In this case, we say that the represented instruction sequences are *instruction sequence congruent*. We write $\mathrm{PGA}^{\mathrm{isc}}$ for the algebraic theory whose sorts, constants and operators are those of PGA, but whose axioms are PGA1–PGA4.

The informal explanation of closed PGA terms as sequences of primitive instructions given above can be looked upon as a sketch of the in-

[4] We write $\mathbb{N}_1$ for the set $\{n \in \mathbb{N} \mid n \geq 1\}$ of positive natural numbers.

tended model of the axioms of $\mathrm{PGA}^{\mathrm{isc}}$. This model, which is described in detail in, for example, [2], is an initial model of the axioms of $\mathrm{PGA}^{\mathrm{isc}}$.

The *unfolding equation* $X^\omega = X \,;\, X^\omega$ is derivable from the axioms of $\mathrm{PGA}^{\mathrm{isc}}$ by first taking the instance of PGA2 in which $n = 2$, then applying PGA4, and finally applying the instance of PGA2 in which $n = 2$ again.

A closed PGA term is in *first canonical form* if it is of the form t or $t \,;\, t'^\omega$, where t and t' are closed repetition-free PGA terms. The following proposition, proved in [2], relates $\mathrm{PGA}^{\mathrm{isc}}$ and first canonical forms.

Proposition 1. *For all closed* PGA *terms* t*, there exists a closed* PGA *term* t' *that is in first canonical form such that* $t = t'$ *is derivable from the axioms of* $\mathrm{PGA}^{\mathrm{isc}}$*.*

The examples given above of equations derivable from the axioms of PGA are derivable from the axioms of $\mathrm{PGA}^{\mathrm{isc}}$ only. Their left-hand sides are not in first canonical form and their right-hand sides are in first canonical form. Simple examples of equations derivable from the axioms of PGA and not derivable from the axioms of $\mathrm{PGA}^{\mathrm{isc}}$ are

$$-a \,;\, \#2 \,;\, (+b \,;\, \#2)^\omega = -a \,;\, \#0 \,;\, (+b \,;\, \#0)^\omega \;,$$
$$+a \,;\, \#6 \,;\, b \,;\, (-c \,;\, \#9)^\omega = +a \,;\, \#2 \,;\, b \,;\, (-c \,;\, \#1)^\omega \;.$$

Closed PGA terms t and t' represent the same instruction sequence after changing all chained jumps into single jumps and making all jumps as short as possible iff $t = t'$ is derivable from PGA1–PGA8. In this case, we say that the represented instruction sequences are *structurally congruent*.

A closed PGA term t *has chained jumps* if there exists a closed PGA term t' such that $t = t'$ is derivable from the axioms of $\mathrm{PGA}^{\mathrm{isc}}$ and t' contains a subterm of the form $\#n{+}1 \,;\, u_1 \,;\, \ldots \,;\, u_n \,;\, \#l$. A closed PGA term t of the form $u_1 \,;\, \ldots \,;\, u_m \,;\, (v_1 \,;\, \ldots \,;\, v_k)^\omega$ *has shortest possible jumps* if: (i) for each $i \in [1, m]$ for which u_i is of the form $\#l$, $l \leq k + m - i$; (ii) for each $j \in [1, k]$ for which v_j is of the form $\#l$, $l \leq k - 1$. A closed PGA term is in *second canonical form* if it is in first canonical form,

does not have chained jumps, and has shortest possible jumps if it has a repeating part. The following proposition, proved in [2], relates PGA and second canonical forms.

Proposition 2. *For all closed* PGA *terms* t*, there exists a closed* PGA *term* t' *that is in second canonical form such that* $t = t'$ *is derivable from the axioms of* PGA*.*

The examples given above of equations derivable from the axioms of PGA and not derivable from the axioms of $\mathrm{PGA}^{\mathrm{isc}}$ have left-hand sides that are not in second canonical form and right-hand sides that are in second canonical form.

Henceforth, the instruction sequences of the kind considered in PGA are called PGA instruction sequences.

In Section 7, we will use the notation $;_{i=1}^{n} t_i$. For each $i \in \mathbb{N}_1$, let t_i be PGA terms. Then, for each $n \in \mathbb{N}_1$, the term $;_{i=1}^{n} t_i$ is defined by induction on n as follows: $;_{i=1}^{1} t_i = t_1$ and $;_{i=1}^{n+1} t_i = ;_{i=1}^{n} t_i \,;\, t_{n+1}$.

3 Basic Thread Algebra for Finite and Infinite Threads

In this section, we present BTA (Basic Thread Algebra) and an extension of BTA that reflects the idea that infinite threads are identical if their approximations up to any finite depth are identical.

BTA is concerned with mathematical objects that model in a direct way the behaviours produced by PGA instruction sequences under execution. The objects in question are called threads. A thread models a behaviour that consists of performing basic actions in a sequential fashion. Upon performing a basic action, a reply from an execution environment determines how the behaviour proceeds subsequently. The possible replies are the Boolean values 0 and 1.

The basic instructions from $\mathcal{A}$ are taken as basic actions. Besides, tau is taken as a special basic action. It is assumed that $\mathsf{tau} \notin \mathcal{A}$. We write $\mathcal{A}_{\mathsf{tau}}$ for $\mathcal{A} \cup \{\mathsf{tau}\}$.

BTA has one sort: the sort $\mathbf{T}$ of *threads*. We make this sort explicit to anticipate the need for many-sortedness later on. To build terms of sort $\mathbf{T}$, BTA has the following constants and operators:

- the *inaction* constant $\mathsf{D} :\to \mathbf{T}$;
- the *termination* constant $\mathsf{S} :\to \mathbf{T}$;
- for each $\alpha \in \mathcal{A}_{\mathsf{tau}}$, the binary *postconditional composition* operator $\_ \trianglelefteq \alpha \trianglerighteq \_ : \mathbf{T} \times \mathbf{T} \to \mathbf{T}$.

Terms of sort $\mathbf{T}$ are built as usual in the one-sorted case. We assume that there are infinitely many variables of sort $\mathbf{T}$, including x, y, z. We use infix notation for postconditional composition. Taking this notational convention into account, the syntax of closed BTA terms (of sort $\mathbf{T}$) can be defined in Backus-Naur style as follows:

$$CT_{\mathbf{T}} ::= \mathsf{D} \mid \mathsf{S} \mid (CT_{\mathbf{T}} \trianglelefteq \alpha \trianglerighteq CT_{\mathbf{T}}) \,,$$

where $\alpha \in \mathcal{A}_{\mathsf{tau}}$. We introduce *basic action prefixing* as an abbreviation: $\alpha \circ t$, where $\alpha \in \mathcal{A}_{\mathsf{tau}}$ and t is a BTA term, abbreviates $t \trianglelefteq \alpha \trianglerighteq t$. We treat an expression of the form $\alpha \circ t$ and the BTA term that it abbreviates as syntactically the same.

Closed BTA terms are considered to represent threads. The thread represented by a closed term of the form $t \trianglelefteq \alpha \trianglerighteq t'$ models the behaviour that first performs α, and then proceeds as the behaviour modeled by the thread represented by t if the reply from the execution environment is 1 and proceeds as the behaviour modeled by the thread represented by t' if the reply from the execution environment is 0. Performing tau, which is considered performing an internal action, always leads to the reply 1. The thread represented by S models the behaviour that does nothing else but terminate and the thread represented by D models the behaviour that is inactive, i.e. it performs no more basic actions and it does not terminate.

A simple example of a closed BTA term is

$$(b \circ \mathsf{S}) \trianglelefteq a \trianglerighteq \mathsf{D} \,.$$

Table 2. Axioms of BTA

$x \trianglelefteq \mathsf{tau} \trianglerighteq y = x \trianglelefteq \mathsf{tau} \trianglerighteq x$	T1

This term denotes the thread that first performs basic action a, if the reply from the execution environment on performing a is 1, it next performs the basic action b and then terminates, and if the reply from the execution environment on performing a is 0, it next becomes inactive.

BTA has only one axiom. This axiom is given in Table 2. Using the abbreviation introduced above, it can also be written as follows: $x \trianglelefteq \mathsf{tau} \trianglerighteq y = \mathsf{tau} \circ x$.

Each closed BTA term represents a finite thread, i.e. a thread with a finite upper bound to the number of basic actions that it can perform. Infinite threads, i.e. threads without a finite upper bound to the number of basic actions that it can perform, can be defined by means of a set of recursion equations (see e.g. [12]).

A simple example of a set of recursion equations that consists of a single equation is

$$x = (b \circ \mathsf{S}) \trianglelefteq a \trianglerighteq x \ .$$

Its solution is the thread that first repeatedly performs basic action a until the reply from the execution environment on performing a is 1, next performs the basic action b and then terminates.

A regular thread is a finite or infinite thread that can be defined by means of a finite set of recursion equations. The behaviours produced by PGA instruction sequences under execution are exactly the behaviours modeled by regular threads.

Two infinite threads are considered identical if their approximations up to any finite depth are identical. The approximation up to depth n of a thread models the behaviour that differs from the behaviour modeled by the thread in that it will become inactive after it has performed n actions unless it would terminate at this point. AIP (Approximation Induction Principle) is a conditional equation that formalizes the above-mentioned view on infinite threads. In AIP, the approximation

Table 3. Axioms for the projection operators and AIP

$\pi_0(x) = \mathsf{D}$	PR1
$\pi_{n+1}(\mathsf{D}) = \mathsf{D}$	PR2
$\pi_{n+1}(\mathsf{S}) = \mathsf{S}$	PR3
$\pi_{n+1}(x \trianglelefteq \alpha \trianglerighteq y) = \pi_n(x) \trianglelefteq \alpha \trianglerighteq \pi_n(y)$	PR4
$\bigwedge_{n \geq 0} \pi_n(x) = \pi_n(y) \Rightarrow x = y$	AIP

up to depth n is phrased in terms of the unary *projection* operator $\pi_n : \mathbf{T} \to \mathbf{T}$.

The axioms for the projection operators and AIP are given in Table 3. In this table, α stands for an arbitrary basic action from $\mathcal{A}_{\mathsf{tau}}$ and n stands for an arbitrary natural number from $\mathbb{N}$. We write BTA$^\infty$ for BTA extended with the projection operators, the axioms for the projection operators, and AIP.

By AIP, we have to deal in BTA$^\infty$ with conditional equational formulas with a countably infinite number of premises. Therefore, infinitary conditional equational logic is used in deriving equations from the axioms of BTA$^\infty$. A complete inference system for infinitary conditional equational logic can be found in, for example, [17].

For a simple example of the use of the axioms for the projection operators and AIP, we consider the (recursion) equations $x = a \circ x$ and $y = a \circ a \circ y$. With these equations as hypotheses, the following equations are derivable from the axioms for the projection operators:

$$\begin{aligned}
&\pi_0(x) = \mathsf{D}\,, && \pi_0(y) = \mathsf{D}\,,\\
&\pi_1(x) = a \circ \mathsf{D}\,, && \pi_1(y) = a \circ \mathsf{D}\,,\\
&\pi_2(x) = a \circ a \circ \mathsf{D}\,, && \pi_2(y) = a \circ a \circ \mathsf{D}\,,\\
&\pi_3(x) = a \circ a \circ a \circ \mathsf{D}\,, && \pi_3(y) = a \circ a \circ a \circ \mathsf{D}\,,\\
&\qquad\vdots
\end{aligned}$$

Hence, the conditional equation $x = a \circ x \wedge y = a \circ a \circ y \Rightarrow x = y$ is

derivable from the axioms for the projection operators and AIP. This conditional equation tells us that the recursion equations $x = a \circ x$ and $y = a \circ a \circ y$ have the same solution.

4 Thread Extraction and Behavioural Congruence

In this section, we make precise in the setting of BTA^∞ which behaviours are produced by PGA instruction sequences under execution and introduce the notion of behavioural congruence on PGA instruction sequences.

To make precise which behaviours are produced by PGA instruction sequences under execution, we introduce an operator $|\_|$ meant for extracting from each PGA instruction sequence the thread that models the behaviour produced by it under execution. For each closed PGA term t, $|t|$ represents the thread that models the behaviour produced by the instruction sequence represented by t under execution.

Formally, we combine PGA with BTA^∞ and extend the combination with the *thread extraction* operator $|\_| : \mathbf{IS} \to \mathbf{T}$ and the axioms given in Table 4. In this table, a stands for an arbitrary basic instruction from $\mathcal{A}$, u stands for an arbitrary primitive instruction from $\mathcal{I}$, and l stands for an arbitrary natural number from $\mathbb{N}$. We write

Table 4. Axioms for the thread extraction operator

$\lvert a\rvert = a \circ \mathsf{D}$	TE1	$\lvert \#l\rvert = \mathsf{D}$	TE7
$\lvert a\,;X\rvert = a \circ \lvert X\rvert$	TE2	$\lvert \#0\,;X\rvert = \mathsf{D}$	TE8
$\lvert +a\rvert = a \circ \mathsf{D}$	TE3	$\lvert \#1\,;X\rvert = \lvert X\rvert$	TE9
$\lvert +a\,;X\rvert = \lvert X\rvert \trianglelefteq a \trianglerighteq \lvert \#2\,;X\rvert$	TE4	$\lvert \#l+2\,;u\rvert = \mathsf{D}$	TE10
$\lvert -a\rvert = a \circ \mathsf{D}$	TE5	$\lvert \#l+2\,;u\,;X\rvert = \lvert \#l+1\,;X\rvert$	TE11
$\lvert -a\,;X\rvert = \lvert \#2\,;X\rvert \trianglelefteq a \trianglerighteq \lvert X\rvert$	TE6	$\lvert !\rvert = \mathsf{S}$	TE12
		$\lvert !\,;X\rvert = \mathsf{S}$	TE13

PGA/BTA$^\infty$ for the combination of PGA and BTA$^\infty$ extended with the thread extraction operator and the axioms for the thread extraction operator. The syntax of closed PGA/BTA$^\infty$ terms of sort $\mathbf{T}$ can be defined in Backus-Naur style as follows:

$$CT'_{\mathbf{T}} ::= \mathsf{D} \mid \mathsf{S} \mid (CT'_{\mathbf{T}} \trianglelefteq \alpha \trianglerighteq CT'_{\mathbf{T}}) \mid |CT_{\mathbf{IS}}| \;,$$

where $\alpha \in \mathcal{A}_{\mathsf{tau}}$. $CT_{\mathbf{IS}}$ is defined in Section 2.

A simple example of thread extraction is

$$|{+}a \,;\, \#2 \,;\, \#3 \,;\, b \,;\, !| = (b \circ \mathsf{S}) \trianglelefteq a \trianglerighteq \mathsf{D} \;.$$

In the case of infinite instruction sequences, thread extraction yields threads definable by means of a set of recursion equations. For example,

$$|({+}a \,;\, \#2 \,;\, \#3 \,;\, b \,;\, !)^\omega|$$

is the solution of the set of recursion equations that consists of the single equation

$$x = (b \circ \mathsf{S}) \trianglelefteq a \trianglerighteq x \;.$$

If a closed PGA term t represents an instruction sequence that starts with an infinite chain of forward jumps, then TE9 and TE11 can be applied to $|t|$ infinitely often without ever showing that a basic action is performed. In this case, we have to do with inaction and, being consistent with that, $|t| = \mathsf{D}$ is derivable from the axioms of PGA and TE1–TE13. By contrast, $|t| = \mathsf{D}$ is not derivable from the axioms of PGA$^{\mathrm{isc}}$ and TE1–TE13. However, if closed PGA terms t and t' represent instruction sequences in which no infinite chains of forward jumps occur, then $t = t'$ is derivable from the axioms of PGA only if $|t| = |t'|$ is derivable from the axioms of PGA$^{\mathrm{isc}}$ and TE1–TE13.

If a closed PGA term t represents an infinite instruction sequence, then we can extract the approximations of the thread modeling the behaviour produced by that instruction sequence under execution up to every finite depth: for each $n \in \mathbb{N}$, there exists a closed BTA term t'' such that $\pi_n(|t|) = t''$ is derivable from the axioms of PGA, TE1–TE13, the axioms of BTA, and PR1–PR4. If closed PGA terms t

and t' represent infinite instruction sequences that produce the same behaviour under execution, then this can be proved using the following instance of AIP: $\bigwedge_{n\geq 0} \pi_n(|t|) = \pi_n(|t'|) \Rightarrow |t| = |t'|$.

PGA instruction sequences are behaviourally equivalent if they produce the same behaviour under execution. Behavioural equivalence is not a congruence. Instruction sequences are behaviourally congruent if they produce the same behaviour irrespective of the way they are entered and the way they are left.

Let t and t' be closed PGA terms. Then:

- t and t' are *behaviourally equivalent*, written $t \equiv_{\mathsf{be}} t'$, if $|t| = |t'|$ is derivable from the axioms of PGA/BTA$^\infty$.
- t and t' are *behaviourally congruent*, written $t \cong_{\mathsf{bc}} t'$, if, for each $l, n \in \mathbb{N}$, $\#l \,;\, t \,;\, !^n \equiv_{\mathsf{be}} \#l \,;\, t' \,;\, !^n$.[5]

Some simple examples of behavioural equivalence are

$$a \,;\, \#2 \,;\, +b \,;\, ! \equiv_{\mathsf{be}} a \,;\, \#2 \,;\, +c \,;\, ! \;,$$
$$(+a \,;\, \#2 \,;\, \#3 \,;\, b \,;\, !)^\omega \equiv_{\mathsf{be}} (-a \,;\, \#3 \,;\, b \,;\, !)^\omega \;.$$

We cannot lift these examples to behavioural congruence, i.e.

$$a \,;\, \#2 \,;\, +b \,;\, ! \not\cong_{\mathsf{bc}} a \,;\, \#2 \,;\, +c \,;\, ! \;,$$
$$(+a \,;\, \#2 \,;\, \#3 \,;\, b \,;\, !)^\omega \not\cong_{\mathsf{bc}} (-a \,;\, \#3 \,;\, b \,;\, !)^\omega \;.$$

A simple example of behavioural congruence is

$$(+a \,;\, \#3 \,;\, \#2 \,;\, b)^\omega \cong_{\mathsf{bc}} (-a \,;\, \#3 \,;\, \#2 \,;\, b)^\omega \;.$$

It is proved in [2] that each closed PGA term is behaviourally equivalent to a term of the form t^ω, where t is a closed repetition-free PGA term.

Proposition 3. *For all closed* PGA *terms t, there exists a closed repetition-free* PGA *term t' such that $t \equiv_{\mathsf{be}} t'^\omega$.*

[5] We use the convention that $t \,;\, t'^0$ stands for t.

Behavioural congruence is the largest congruence contained in behavioural equivalence. Moreover, structural congruence implies behavioural congruence.

Proposition 4. *For all closed* PGA *terms* t *and* t', $t = t'$ *is derivable from the axioms of* PGA *only if* $t \mathrel{\mathsf{bc}} t'$.

Proof. The proof is basically the proof of Proposition 2.2 from [2]. In that proof use is made of the uniqueness of solutions of sets of recursion equations where each right-hand side is a BTA term of the form D, S or $s \trianglelefteq \alpha \trianglerighteq s'$ with BTA terms s and s' that contain only variables occurring as one of the right-hand sides. This uniqueness follows from AIP (see also Corollary 2.1 from [2]). □

Conversely, behavioural congruence does not imply structural congruence. For example, $+a\,;!\,;!\mathrel{\mathsf{bc}} -a\,;!\,;!$, but $+a\,;!\,;! = -a\,;!\,;!$ is not derivable from the axioms of PGA.

In [18], we present an equational axiom system for behavioural congruence that is sound for closed PGA terms and complete for closed repetition-free PGA terms.

The following proposition, proved in [2], puts the expressiveness of PGA in terms of producible behaviours.

Proposition 5. *Let* $\mathcal{M}$ *be a model of* PGA/BTA$^\infty$. *Then, for each element* p *from the domain associated with the sort* $\mathbf{T}$ *in* $\mathcal{M}$, *there exists a closed* PGA *term* t *such that* p *is the interpretation of* $|t|$ *in* $\mathcal{M}$ *iff* p *is a component of the solution of a finite set of recursion equations* $\{V = t_V \mid V \in \mathcal{V}\}$, *where* $\mathcal{V}$ *is a set of variables of sort* $\mathbf{T}$ *and each* t_V *is a* BTA *term that is not a variable and contains only variables from* $\mathcal{V}$.

More results on the expressiveness of PGA can be found in [2].

5 The Case of Instructions for Boolean Registers

In this section, we present the instantiation of PGA in which all possible instructions to read out and alter Boolean registers are taken as basic

instructions.

In this instantiation, it is assumed that a fixed but arbitrary set $\mathcal{F}$ of *foci* has been given. Foci serve as names of Boolean registers.

The set of basic instructions used in this instantiation consists of the following:

- for each $f \in \mathcal{F}$ and $p, q : \mathbb{B} \to \mathbb{B}$, a *basic Boolean register instruction* $f.p/q$.

We write $\mathcal{A}_{\mathrm{br}}$ for this set.

Each basic Boolean register instruction consists of two parts separated by a dot. The part on the left-hand side of the dot plays the role of the name of a Boolean register and the part on the right-hand side of the dot plays the role of an operation to be carried out on the named Boolean register when the instruction is executed. The intuition is basically that carrying out the operation concerned modifies the content of the named Boolean register and produces as a reply a Boolean value that depends on the content of the named Boolean register. More precisely, the execution of a basic Boolean register instruction $f.p/q$ has the following effects:

- if the content of the Boolean register named f is b when the execution of $f.p/q$ starts, then its content is $q(b)$ when the execution of $f.p/q$ terminates;
- if the content of the Boolean register named f is b when the execution of $f.p/q$ starts, then the reply produced on termination of the execution of $f.p/q$ is $p(b)$.

The execution of $f.p/q$ has no effect on the content of Boolean registers other than the one named f.

$\mathbb{B} \to \mathbb{B}$, the set of all unary Boolean functions, consists of the following four functions:

- the function 0, satisfying $0(0) = 0$ and $0(1) = 0$;
- the function 1, satisfying $1(0) = 1$ and $1(1) = 1$;

- the function i, satisfying $\mathrm{i}(0) = 0$ and $\mathrm{i}(1) = 1$;
- the function c, satisfying $\mathrm{c}(0) = 1$ and $\mathrm{c}(1) = 0$.

In [3]–[5],[7],[8], we actually used the operations $0/0$, $1/1$, and i/i, but denoted them by $\mathsf{set}{:}0$, $\mathsf{set}{:}1$ and get, respectively. In [6], we actually used, in addition to these operations, the operation c/c, but denoted it by com. Two examples of peculiar operations are $0/\mathrm{i}$ and $1/\mathrm{i}$. Carrying out one of these operations on a Boolean register does not modify the content of the Boolean register and produces as a reply, irrespective of the content of the Boolean register, always the same Boolean value.

We write $[\mathrm{PGA/BTA}^\infty](\mathcal{A}_{\mathrm{br}})$ for $\mathrm{PGA/BTA}^\infty$ with $\mathcal{A}$ instantiated by $\mathcal{A}_{\mathrm{br}}$. Notice that $[\mathrm{PGA/BTA}^\infty](\mathcal{A}_{\mathrm{br}})$ is itself parameterized by a set of foci.

In the papers just mentioned, $\mathcal{F}$ is instantiated by

$$\{\mathsf{in}{:}i \mid i \in \mathbb{N}_1\} \cup \{\mathsf{out}{:}i \mid i \in \mathbb{N}_1\} \cup \{\mathsf{aux}{:}i \mid i \in \mathbb{N}_1\}$$

if the computation of functions from $\mathbb{B}^n$ to $\mathbb{B}^m$ with $m > 1$ is in order and

$$\{\mathsf{in}{:}i \mid i \in \mathbb{N}_1\} \cup \{\mathsf{out}\} \cup \{\mathsf{aux}{:}i \mid i \in \mathbb{N}_1\}$$

if only the computation of functions from $\mathbb{B}^n$ to $\mathbb{B}$ is in order. These foci are employed as follows:

- the foci of the form $\mathsf{in}{:}i$ serve as names of Boolean registers that are used as input registers in instruction sequences;
- the foci of the form $\mathsf{out}{:}i$ and out serve as names of Boolean registers that are used as output registers in instruction sequences;
- the foci of the form $\mathsf{aux}{:}i$ serve as names of Boolean registers that are used as auxiliary registers in instruction sequences.

The above sets of foci are just examples of sets by which $\mathcal{F}$ may be instantiated. In the algebraic theories presented in Sections 6 and 7, $\mathcal{F}$ is not instantiated.

6 Boolean Register Families

PGA instruction sequences under execution may interact with the named Boolean registers from a family of Boolean registers provided by their execution environment. In this section, we introduce an algebraic theory of Boolean register families called BRFA (Boolean Register Family Algebra). Boolean register families are reminiscent of the Boolean register files found in the central processing unit of a computer (see e.g. [19]).

In BRFA, as in [PGA/BTA$^\infty$]($\mathcal{A}_{\mathrm{br}}$), it is assumed that a fixed but arbitrary set $\mathcal{F}$ of foci has been given.

BRFA has one sort: the sort **BRF** of *Boolean register families*. To build terms of sort **BRF**, BRFA has the following constants and operators:

- the *empty Boolean register family* constant $\emptyset :\to \mathbf{BRF}$;
- for each $f \in \mathcal{F}$ and $b \in \mathbb{B} \cup \{*\}$, the *singleton Boolean register family* constant $f.\mathsf{br}(b) :\to \mathbf{BRF}$;
- the binary *Boolean register family composition* operator $\_ \oplus \_ : \mathbf{BRF} \times \mathbf{BRF} \to \mathbf{BRF}$;
- for each $F \subseteq \mathcal{F}$, the unary *encapsulation* operator $\partial_F : \mathbf{BRF} \to \mathbf{BRF}$.

We assume that there are infinitely many variables of sort **BRF**, including u, v, w. We use infix notation for the Boolean register family composition operator. Taking this notational convention into account, the syntax of closed BRFA terms (of sort **BRF**) can be defined in Backus-Naur style as follows:

$$CT_{\mathbf{BRF}} ::= \emptyset \mid f.\mathsf{br}(b) \mid (CT_{\mathbf{BRF}} \oplus CT_{\mathbf{BRF}}) \mid \partial_F(CT_{\mathbf{BRF}}) \,,$$

where $f \in \mathcal{F}$, $b \in \mathbb{B} \cup \{*\}$, and $F \subseteq \mathcal{F}$.

The Boolean register family denoted by $\emptyset$ is the empty Boolean register family. The Boolean register family denoted by a closed term

of the form $f.\mathsf{br}(b)$, where $b \in \mathbb{B}$, consists of one named Boolean register only, the Boolean register concerned is an operative Boolean register named f whose content is b. The Boolean register family denoted by a closed term of the form $f.\mathsf{br}(*)$ consists of one named Boolean register only, the Boolean register concerned is an inoperative Boolean register named f. The Boolean register family denoted by a closed term of the form $t \oplus t'$ consists of all named Boolean registers that belong to either the Boolean register family denoted by t or the Boolean register family denoted by t'. In the case where a named Boolean register from the Boolean register family denoted by t and a named Boolean register from the Boolean register family denoted by t' have the same name, they collapse to an inoperative Boolean register with the name concerned. The Boolean register family denoted by a closed term of the form $\partial_F(t)$ consists of all named Boolean registers with a name not in F that belong to the Boolean register family denoted by t.

A simple example of a Boolean register family is

$$\begin{aligned}&\mathsf{aux}{:}8.\mathsf{br}(1) \oplus \mathsf{aux}{:}7.\mathsf{br}(1) \oplus \mathsf{aux}{:}6.\mathsf{br}(0) \oplus \mathsf{aux}{:}5.\mathsf{br}(0)\\ &\quad \oplus \mathsf{aux}{:}4.\mathsf{br}(1) \oplus \mathsf{aux}{:}3.\mathsf{br}(1) \oplus \mathsf{aux}{:}2.\mathsf{br}(1) \oplus \mathsf{aux}{:}1.\mathsf{br}(0)\ .\end{aligned}$$

This Boolean register family can be seen as a storage cell whose content is the bit string 01110011. Taking the content of such storage cells for binary representations of natural numbers, the functions on bit strings of length 8 that model addition, subtraction, and multiplication modulo 2^8 of natural numbers less than 2^8 can be computed using the instructions for Boolean registers introduced in Section 5.

An inoperative Boolean register can be viewed as a Boolean register whose content is unavailable. Carrying out an operation on an inoperative Boolean register is impossible.

The axioms of BRFA are given in Table 5. In this table, f stands for an arbitrary focus from $\mathcal{F}$, F stands for an arbitrary subset of $\mathcal{F}$, and b and b' stand for arbitrary values from $\mathbb{B} \cup \{*\}$. These axioms simply formalize the informal explanation given above.

The following two propositions, proved in [2], concern an elimination result and a representation result for closed BRFA terms.

Table 5. Axioms of BRFA

$u \oplus \emptyset = u$	BRFC1	$\partial_F(\emptyset) = \emptyset$		BRFE1
$u \oplus v = v \oplus u$	BRFC2	$\partial_F(f.\mathsf{br}(b)) = \emptyset$	if $f \in F$	BRFE2
$(u \oplus v) \oplus w = u \oplus (v \oplus w)$	BRFC3	$\partial_F(f.\mathsf{br}(b)) = f.\mathsf{br}(b)$	if $f \notin F$	BRFE3
$f.\mathsf{br}(b) \oplus f.\mathsf{br}(b') = f.\mathsf{br}(*)$	BRFC4	$\partial_F(u \oplus v) = \partial_F(u) \oplus \partial_F(v)$		BRFE4

Proposition 6. *For all closed* BRFA *terms t, there exists a closed* BRFA *term t' in which encapsulation operators do not occur such that $t = t'$ is derivable from the axioms of* BRFA.

Proposition 7. *For all closed* BRFA *terms t, for all $f \in \mathcal{F}$, either $t = \partial_{\{f\}}(t)$ is derivable from the axioms of* BRFA *or there exists a $b \in \mathbb{B} \cup \{*\}$ such that $t = f.\mathsf{br}(b) \oplus \partial_{\{f\}}(t)$ is derivable from the axioms of* BRFA.

In Section 8, we will use the notation $\bigoplus_{i=1}^{n} t_i$. For each $i \in \mathbb{N}_1$, let t_i be a term of sort **BRF**. Then, for each $n \in \mathbb{N}_1$, the term $\bigoplus_{i=1}^{n} t_i$ is defined by induction on n as follows: $\bigoplus_{i=1}^{1} t_i = t_1$ and $\bigoplus_{i=1}^{n+1} t_i = \bigoplus_{i=1}^{n} t_i \oplus t_{n+1}$.

7 Interaction of Threads with Boolean Registers

A PGA instruction sequence under execution may interact with the named Boolean registers from the family of Boolean registers provided by its execution environment. In line with this kind of interaction, a thread may perform a basic action basically for the purpose of modifying the content of a named Boolean register or receiving a reply value that depends on the content of a named Boolean register. In this section, we introduce related operators.

We combine PGA/BTA$^{\infty}(\mathcal{A}_{\mathrm{br}})$ with BRFA and extend the combination with the following operators for interaction of threads with Boolean registers:

- the binary *use* operator $\_ \,/\, \_ : \mathbf{T} \times \mathbf{BRF} \to \mathbf{T}$;
- the binary *apply* operator $\_ \bullet \_ : \mathbf{T} \times \mathbf{BRF} \to \mathbf{BRF}$;
- the unary *abstraction* operator $\tau_{\mathsf{tau}} : \mathbf{T} \to \mathbf{T}$;

and the axioms given in Tables 6.[6] In these tables, f stands for an arbitrary focus from $\mathcal{F}$, p and q stand for arbitrary Boolean functions from $\mathbb{B} \to \mathbb{B}$, b stands for an arbitrary Boolean value from $\mathbb{B}$, n stands for an arbitrary natural number from $\mathbb{N}$, and t and s stand for arbitrary terms of sort **BRF**. We use infix notation for the use and apply operators. We write [PGA/BTA$^\infty$]($\mathcal{A}_{\mathrm{br}}$)/BRI for the combination of [PGA/BTA$^\infty$]($\mathcal{A}_{\mathrm{br}}$) and BRFA extended with the use operator, the apply operator, the abstraction operator, and the axioms for these operators. The syntax of closed [PGA/BTA$^\infty$]($\mathcal{A}_{\mathrm{br}}$)/BRI terms of sort **T** and **BRF** can be defined in Backus-Naur style as follows:

$$\begin{aligned} CT''_{\mathbf{T}} ::= {} & \mathsf{D} \mid \mathsf{S} \mid (CT''_{\mathbf{T}} \trianglelefteq \alpha \trianglerighteq CT''_{\mathbf{T}}) \mid |CT_{\mathbf{IS}}| \\ & \mid (CT''_{\mathbf{T}} \,/\, CT'_{\mathbf{BRF}}) \mid \tau_{\mathsf{tau}}(CT''_{\mathbf{T}}) \,, \end{aligned}$$

$$\begin{aligned} CT'_{\mathbf{BRF}} ::= {} & \emptyset \mid f.\mathsf{br}(b) \mid (CT'_{\mathbf{BRF}} \oplus CT'_{\mathbf{BRF}}) \mid \partial_F(CT'_{\mathbf{BRF}}) \\ & \mid (CT''_{\mathbf{T}} \bullet CT'_{\mathbf{BRF}}) \,, \end{aligned}$$

where $\alpha \in \mathcal{A}_{\mathrm{br}} \cup \{\mathsf{tau}\}$, $f \in \mathcal{F}$, $b \in \mathbb{B} \cup \{*\}$, $F \subseteq \mathcal{F}$. $CT_{\mathbf{IS}}$ is defined in Section 2.

Axioms U1–U7 and A1–A7 formalize the informal explanation of the use operator and the apply operator given below and in addition stipulate what is the result of apply if an unavailable focus is involved (A4) and what is the result of use and apply if an inoperative Boolean register is involved (U7 and A7). Axioms U8 and A8 allow of reasoning about infinite threads, and therefore about the behaviour produced by infinite instruction sequences under execution, in the context of use and apply, respectively.

On interaction between a thread and a Boolean register, the thread affects the Boolean register and the Boolean register affects the thread.

[6] We write $t[t'/x]$ for the result of substituting term t' for variable x in term t.

Table 6. Axioms for the use, apply and abstraction operator

$\mathsf{S} \mathbin{/} u = \mathsf{S}$	U1
$\mathsf{D} \mathbin{/} u = \mathsf{D}$	U2
$(\mathsf{tau} \circ x) \mathbin{/} u = \mathsf{tau} \circ (x \mathbin{/} u)$	U3
$(x \trianglelefteq f.p/q \trianglerighteq y) \mathbin{/} \partial_{\{f\}}(u) = (x \mathbin{/} \partial_{\{f\}}(u)) \trianglelefteq f.p/q \trianglerighteq (y \mathbin{/} \partial_{\{f\}}(u))$	U4
$(x \trianglelefteq f.p/q \trianglerighteq y) \mathbin{/} (f.\mathsf{br}(b) \oplus \partial_{\{f\}}(u)) = \mathsf{tau} \circ (x \mathbin{/} (f.\mathsf{br}(q(b)) \oplus \partial_{\{f\}}(u)))$ if $p(b) = 1$	U5
$(x \trianglelefteq f.p/q \trianglerighteq y) \mathbin{/} (f.\mathsf{br}(b) \oplus \partial_{\{f\}}(u)) = \mathsf{tau} \circ (y \mathbin{/} (f.\mathsf{br}(q(b)) \oplus \partial_{\{f\}}(u)))$ if $p(b) = 0$	U6
$(x \trianglelefteq f.p/q \trianglerighteq y) \mathbin{/} (f.\mathsf{br}(*) \oplus \partial_{\{f\}}(u)) = \mathsf{D}$	U7
$\pi_n(x \mathbin{/} u) = \pi_n(x) \mathbin{/} u$	U8
$\mathsf{S} \bullet u = u$	A1
$\mathsf{D} \bullet u = \emptyset$	A2
$(\mathsf{tau} \circ x) \bullet u = \mathsf{tau} \circ (x \bullet u)$	A3
$(x \trianglelefteq f.p/q \trianglerighteq y) \bullet \partial_{\{f\}}(u) = \emptyset$	A4
$(x \trianglelefteq f.p/q \trianglerighteq y) \bullet (f.\mathsf{br}(b) \oplus \partial_{\{f\}}(u)) = x \bullet (f.\mathsf{br}(q(b)) \oplus \partial_{\{f\}}(u))$ if $p(b) = 1$	A5
$(x \trianglelefteq f.p/q \trianglerighteq y) \bullet (f.\mathsf{br}(b) \oplus \partial_{\{f\}}(u)) = y \bullet (f.\mathsf{br}(q(b)) \oplus \partial_{\{f\}}(u))$ if $p(b) = 0$	A6
$(x \trianglelefteq f.p/q \trianglerighteq y) \bullet (f.\mathsf{br}(*) \oplus \partial_{\{f\}}(u)) = \emptyset$	A7
$\bigwedge_{k \geq n} t[\pi_k(x)/z] = s[\pi_k(y)/z] \Rightarrow t[x/z] = s[y/z]$	A8
$\tau_{\mathsf{tau}}(\mathsf{S}) = \mathsf{S}$	C1
$\tau_{\mathsf{tau}}(\mathsf{D}) = \mathsf{D}$	C2
$\tau_{\mathsf{tau}}(\mathsf{tau} \circ x) = \tau_{\mathsf{tau}}(x)$	C3
$\tau_{\mathsf{tau}}(x \trianglelefteq f.p/q \trianglerighteq y) = \tau_{\mathsf{tau}}(x) \trianglelefteq f.p/q \trianglerighteq \tau_{\mathsf{tau}}(y)$	C4
$\bigwedge_{n \geq 0} \tau_{\mathsf{tau}}(\pi_n(x)) = \tau_{\mathsf{tau}}(\pi_n(y)) \Rightarrow \tau_{\mathsf{tau}}(x) = \tau_{\mathsf{tau}}(y)$	C5

The use operator concerns the effects of Boolean registers on threads and the apply operator concerns the effects of threads on Boolean registers. The thread denoted by a closed term of the form $t \mathbin{/} t'$ and the Boolean register family denoted by a closed term of the form $t \bullet t'$ are the thread and Boolean register family, respectively, that result from carrying out the operation that is part of each basic action performed by the thread denoted by t on the Boolean register in the Boolean register family denoted by t' with the focus that is part of the basic action as its name. When the operation that is part of a basic action performed by a thread is carried out on a Boolean register, the content of the Boolean register is modified according to the operation concerned and the thread is affected as follows: the basic action turns into the internal action tau and the two ways to proceed reduce to one on the basis of the reply value produced according to the operation concerned.

With the use operator the internal action tau is left as a trace of each basic action that has led to carrying out an operation on a Boolean register. The abstraction operator serves to abstract fully from such internal activity by concealing tau. Axioms C1–C4 formalize the concealment of tau. Axiom C5 allows of reasoning about infinite threads in the context of abstraction.

A simple example of use and apply is

$$\begin{aligned}
&|\,;_{i=1}^{4}(-\mathsf{aux}{:}i.\mathrm{i}/\mathrm{i}\,;\#3\,;\mathsf{aux}{:}i.0/0\,;!\,;\mathsf{aux}{:}i.1/1)| \\
&\quad /\ \mathsf{aux}{:}4.\mathsf{br}(1) \oplus \mathsf{aux}{:}3.\mathsf{br}(1) \oplus \mathsf{aux}{:}2.\mathsf{br}(1) \oplus \mathsf{aux}{:}1.\mathsf{br}(0) \\
&= \mathsf{tau} \circ \mathsf{tau} \circ \mathsf{tau} \circ \mathsf{tau} \circ \mathsf{S}\ ,
\end{aligned}$$

$$\begin{aligned}
&|\,;_{i=1}^{4}(-\mathsf{aux}{:}i.\mathrm{i}/\mathrm{i}\,;\#3\,;\mathsf{aux}{:}i.0/0\,;!\,;\mathsf{aux}{:}i.1/1)| \\
&\quad \bullet\ \mathsf{aux}{:}4.\mathsf{br}(1) \oplus \mathsf{aux}{:}3.\mathsf{br}(1) \oplus \mathsf{aux}{:}2.\mathsf{br}(1) \oplus \mathsf{aux}{:}1.\mathsf{br}(0) \\
&= \mathsf{aux}{:}4.\mathsf{br}(1) \oplus \mathsf{aux}{:}3.\mathsf{br}(1) \oplus \mathsf{aux}{:}2.\mathsf{br}(0) \oplus \mathsf{aux}{:}1.\mathsf{br}(1)\ .
\end{aligned}$$

In this example, the behaviour of the instructions sequence under execution affects the Boolean registers from the Boolean register family such that it corresponds to decrement by one on the natural number represented by the combined content of the Boolean registers. The

equations show that, if the combined content of the Boolean registers represents 14, (a) the Boolean registers reduce the behaviour of the instruction sequence under execution to termination after four internal actions and (b) the behaviour of the instruction sequence under execution modifies the combined content of the Boolean registers to the binary representation of 13.

The following two propositions are about elimination results for closed $[\mathrm{PGA}/\mathrm{BTA}^{\infty}](\mathcal{A}_{\mathrm{br}})/\mathrm{BRI}$ terms.

Proposition 8. *For all closed* $[\mathrm{PGA}/\mathrm{BTA}^{\infty}](\mathcal{A}_{\mathrm{br}})/\mathrm{BRI}$ *terms* t *of sort* **T** *in which all subterms of sort* **IS** *are repetition-free, there exists a closed* $[\mathrm{PGA}/\mathrm{BTA}^{\infty}](\mathcal{A}_{\mathrm{br}})$ *term* t' *of sort* **T** *such that* $t = t'$ *is derivable from the axioms of* $[\mathrm{PGA}/\mathrm{BTA}^{\infty}](\mathcal{A}_{\mathrm{br}})/\mathrm{BRI}$.

Proof. It is easy to prove by structural induction that, for all closed repetition-free $[\mathrm{PGA}/\mathrm{BTA}^{\infty}](\mathcal{A}_{\mathrm{br}})$ terms s of sort **IS**, there exists a closed $[\mathrm{PGA}/\mathrm{BTA}^{\infty}](\mathcal{A}_{\mathrm{br}})$ term s' of sort **T** such that $|s| = s'$ is derivable from the axioms of $[\mathrm{PGA}/\mathrm{BTA}^{\infty}](\mathcal{A}_{\mathrm{br}})$. Therefore, it is sufficient to prove the proposition for all closed $[\mathrm{PGA}/\mathrm{BTA}^{\infty}](\mathcal{A}_{\mathrm{br}})/\mathrm{BRI}$ terms t of sort **T** in which no subterms of sort **IS** occur. This is proved similarly to part (1) of Theorem 3.1 from [2]. □

Proposition 9. *For all closed* $[\mathrm{PGA}/\mathrm{BTA}^{\infty}](\mathcal{A}_{\mathrm{br}})/\mathrm{BRI}$ *terms* t *of sort* **BRF** *in which all subterms of sort* **IS** *are repetition-free, there exists a closed* $[\mathrm{PGA}/\mathrm{BTA}^{\infty}](\mathcal{A}_{\mathrm{br}})$ *term* t' *of sort* **BRF** *such that* $t = t'$ *is derivable from the axioms of* $[\mathrm{PGA}/\mathrm{BTA}^{\infty}](\mathcal{A}_{\mathrm{br}})/\mathrm{BRI}$.

Proof. As in the proof of Proposition 8, it is sufficient to prove the proposition for all closed $[\mathrm{PGA}/\mathrm{BTA}^{\infty}](\mathcal{A}_{\mathrm{br}})/\mathrm{BRI}$ terms t of sort **BRF** in which no subterms of sort **IS** occur. This is proved similarly to part (2) of Theorem 3.1 from [2]. □

8 Computing Partial Functions from $\mathbb{B}^n$ to $\mathbb{B}^m$

In this section, we make precise in the setting of the algebraic theory $[\mathrm{PGA}/\mathrm{BTA}^{\infty}](\mathcal{A}_{\mathrm{br}})/\mathrm{BRI}$ what it means that a given instruction sequence computes a given partial function from $\mathbb{B}^n$ to $\mathbb{B}^m$ $(n, m \in \mathbb{N})$.

For each $n, m \in \mathbb{N}$, we define the following set:

$$\mathcal{F}_{\mathsf{br}}^{n,m} = \{\mathsf{in}{:}i \mid 1 \leq i \leq n\} \cup \{\mathsf{aux}{:}i \mid i \geq 1\} \cup \{\mathsf{out}{:}i \mid 1 \leq i \leq m\} \,.$$

We use the instantiation of $[\mathrm{PGA/BTA}^\infty](\mathcal{A}_{\mathrm{br}})/\mathrm{BRI}$ in which the set of foci is $\bigcup_{n,m\in\mathbb{N}} \mathcal{F}_{\mathsf{br}}^{n,m}$. We write $\mathcal{F}_{\mathsf{br}}$ for this set and we write $[[\mathrm{PGA/BTA}^\infty](\mathcal{A}_{\mathrm{br}})/\mathrm{BRI}](\mathcal{F}_{\mathsf{br}})$ for $[\mathrm{PGA/BTA}^\infty](\mathcal{A}_{\mathrm{br}})/\mathrm{BRI}$ with $\mathcal{F}$ instantiated by $\mathcal{F}_{\mathsf{br}}$.

Let $n, m \in \mathbb{N}$, let $F : \mathbb{B}^n \nrightarrow \mathbb{B}^m$,[7] and let t be a closed repetition-free $[[\mathrm{PGA/BTA}^\infty](\mathcal{A}_{\mathrm{br}})/\mathrm{BRI}](\mathcal{F}_{\mathsf{br}})$ term of sort **IS** in which only foci from $\mathcal{F}_{\mathsf{br}}^{n,m}$ occur. Then t *computes* F if there exists a $k \in \mathbb{N}$ such that:

- for all $b_1, \ldots, b_n, b'_1, \ldots, b'_m \in \mathbb{B}$ with $F(b_1, \ldots, b_n) = b'_1, \ldots, b'_m$:

$$(|t| \,/\, ((\textstyle\bigoplus_{i=1}^{n} \mathsf{in}{:}i.\mathsf{br}(b_i)) \oplus (\bigoplus_{i=1}^{k} \mathsf{aux}{:}i.\mathsf{br}(0)))) \bullet (\textstyle\bigoplus_{i=1}^{m} \mathsf{out}{:}i.\mathsf{br}(0)) = \textstyle\bigoplus_{i=1}^{m} \mathsf{out}{:}i.\mathsf{br}(b'_i) \;;$$

- for all $b_1, \ldots, b_n \in \mathbb{B}$ with $f(b_1, \ldots, b_n)$ undefined:

$$(|t| \,/\, ((\textstyle\bigoplus_{i=1}^{n} \mathsf{in}{:}i.\mathsf{br}(b_i)) \oplus (\bigoplus_{i=1}^{k} \mathsf{aux}{:}i.\mathsf{br}(0)))) \bullet (\textstyle\bigoplus_{i=1}^{m} \mathsf{out}{:}i.\mathsf{br}(0)) = \emptyset \,.$$

With this definition, we can establish whether an instruction sequence of the kind considered in $[[\mathrm{PGA/BTA}^\infty](\mathcal{A}_{\mathrm{br}})/\mathrm{BRI}](\mathcal{F}_{\mathsf{br}})$ computes a given partial function from $\mathbb{B}^n$ to $\mathbb{B}^m$ ($n, m \in \mathbb{N}$) by equational reasoning using the axioms of $[[\mathrm{PGA/BTA}^\infty](\mathcal{A}_{\mathrm{br}})/\mathrm{BRI}](\mathcal{F}_{\mathsf{br}})$.

The following proposition tells us that, for each partial function from $\mathbb{B}^n$ to $\mathbb{B}^m$, there exists an instruction sequence of the kind considered here that computes it.

Proposition 10. *For all $n, m \in \mathbb{N}$, for all $F : \mathbb{B}^n \nrightarrow \mathbb{B}^m$, there exists a closed repetition-free $[[\mathrm{PGA/BTA}^\infty](\mathcal{A}_{\mathrm{br}})/\mathrm{BRI}](\mathcal{F}_{\mathsf{br}})$ term t in which only basic instructions of the forms $f.0/0$, $f.1/1$, and $f.\mathrm{i}/\mathrm{i}$ with $f \in \mathcal{F}_{\mathsf{br}}^{n,m}$ occur such that t computes F.*

[7] We write $f : \mathbb{B}^n \nrightarrow \mathbb{B}^m$ to indicate that f is a partial function from $\mathbb{B}^n$ to $\mathbb{B}^m$.

Proof. As an immediate corollary of the proof of Theorem 5.6 in [2] we have the following: for all $n, m \in \mathbb{N}$, for all $F : \mathbb{B}^n \to \mathbb{B}^m$, there exists a closed repetition-free $[[\mathrm{PGA}/\mathrm{BTA}^\infty](\mathcal{A}_{\mathrm{br}})/\mathrm{BRI}](\mathcal{F}_{\mathsf{br}})$ term t in which only basic instructions of the forms $f.0/0$, $f.1/1$, and $f.\mathrm{i}/\mathrm{i}$ with $f \in \mathcal{F}_{\mathsf{br}}^{n,m}$ occur such that t computes F. It is easy to see from the same proof that this corollary generalizes from total functions to partial functions. □

The following proposition tells us that an instruction sequence in which not only basic instructions of the forms $f.0/0$, $f.1/1$, and $f.\mathrm{i}/\mathrm{i}$ occur can be transformed primitive instruction by primitive instruction to an at most linearly longer instruction sequence computing the same function in which only basic instructions of the forms $f.0/0$, $f.1/1$, and $f.\mathrm{i}/\mathrm{i}$ occur.

The *functional equivalence* relation $\sim_{\mathsf{f}}$ on the set of all closed repetition-free $[[\mathrm{PGA}/\mathrm{BTA}^\infty](\mathcal{A}_{\mathrm{br}})/\mathrm{BRI}](\mathcal{F}_{\mathsf{br}})$ terms of sort **IS** is defined by $t \sim_{\mathsf{f}} t'$ iff there exist $n, m \in \mathbb{N}$ such that:

- t and t' are terms in which only foci from $\mathcal{F}_{\mathsf{br}}^{n,m}$ occur;
- there exists a $F : \mathbb{B}^n \nrightarrow \mathbb{B}^m$ such that t computes F and t' computes F.

Proposition 11. *There exists a unary function ϕ on the set of all closed repetition-free $[[\mathrm{PGA}/\mathrm{BTA}^\infty](\mathcal{A}_{\mathrm{br}})/\mathrm{BRI}](\mathcal{F}_{\mathsf{br}})$ terms of sort* **IS** *such that:*

- *ϕ is the homomorphic extension of a function ϕ' from the set of all $[[\mathrm{PGA}/\mathrm{BTA}^\infty](\mathcal{A}_{\mathrm{br}})/\mathrm{BRI}](\mathcal{F}_{\mathsf{br}})$ constants of sort* **IS** *to the set of all closed repetition-free $[[\mathrm{PGA}/\mathrm{BTA}^\infty](\mathcal{A}_{\mathrm{br}})/\mathrm{BRI}](\mathcal{F}_{\mathsf{br}})$ terms of sort* **IS***;*
- *for all closed repetition-free $[[\mathrm{PGA}/\mathrm{BTA}^\infty](\mathcal{A}_{\mathrm{br}})/\mathrm{BRI}](\mathcal{F}_{\mathsf{br}})$ terms t of sort* **IS***:*
  - *$t \sim_{\mathsf{f}} \phi(t)$;*
  - *$\phi(t)$ is a term in which only basic instructions of the forms $f.0/0$, $f.1/1$, and $f.\mathrm{i}/\mathrm{i}$ occur;*

– *$\phi(t)$ is at most $3 \cdot p$ primitive instructions longer than t, where p is the number of occurrences of basic instructions in t that are not of the form $f.0/0$, $f.1/1$ or $f.\mathrm{i}/\mathrm{i}$.*

Proof. It follows immediately from part (3) of Proposition 3.1 in [2] that the definition of $\sim_{\mathsf{f}}$ given above is a reformulation of the instance of the definition of $\sim_{\mathsf{f}}$ given in [9], where the set $\mathcal{F}$ of foci is instantiated by $\mathcal{F}_{\mathsf{br}}$. This makes the current proposition a corollary of Proposition 2 and Theorem 3 in [9]. □

The view put forward in this section on what it means in the setting of $[\mathrm{PGA}/\mathrm{BTA}^{\infty}](\mathcal{A}_{\mathrm{br}})/\mathrm{BRI}$ that a given instruction sequence computes a given partial function from $\mathbb{B}^n$ to $\mathbb{B}^m$ $(n, m \in \mathbb{N})$ is the view taken in the work on complexity of computational problems, efficiency of algorithms, and algorithmic equivalence of programs presented in [3]–[8]. We remark that Boolean registers cannot only be used to compute partial functions from $\mathbb{B}^n$ to $\mathbb{B}^m$. For example, it is shown in [20] that jump instructions are not necessary if use can be made of Boolean registers.

9 Uses for the Theory

In this section, we give a survey of uses for $[\mathrm{PGA}/\mathrm{BTA}^{\infty}](\mathcal{A}_{\mathrm{br}})/\mathrm{BRI}$.

It is often said that a program is an instruction sequence and, if this characterization has any value, it must be the case that it is somehow easier to understand the concept of an instruction sequence than to understand the concept of a program. The first objective of the work on instruction sequences that started with [10], and of which an enumeration is available at [1], is to understand the concept of a program. The basis of all this work is the parameterized algebraic theory $\mathrm{PGA}/\mathrm{BTA}^{\infty}$ extended to deal with the interaction between instruction sequences under execution and components of their execution environment. The body of theory developed through this work is such that its use as a conceptual preparation for programming is practically feasible.

The notion of an instruction sequence appears in the work in question as a mathematical abstraction for which the rationale is based on

the objective mentioned above. In this capacity, instruction sequences constitute a primary field of investigation in programming comparable to propositions in logic and rational numbers in arithmetic. The structure of the mathematical abstraction at issue has been determined in advance with the hope of applying it in diverse circumstances where in each case the fit may be less than perfect.

Until now, the work in question has, among other things, yielded an approach to computational complexity, where program size is used as complexity measure, a contribution to the conceptual analysis of the notion of an algorithm, and new insights into such diverse issues as the halting problem, program parallelization for the purpose of explicit multi-threading and virus detection.

The work done in the setting of $[\mathrm{PGA}/\mathrm{BTA}^{\infty}](\mathcal{A}_{\mathrm{br}})/\mathrm{BRI}$, which is just an instantiation of the above-mentioned basis, includes:

- Work yielding an approach to computational complexity in which algorithmic problems are viewed as families of functions that consist of a function from $\mathbb{B}^n$ to $\mathbb{B}$ for each natural number n and the complexity of such problems is assessed in terms of the length of instruction sequences that compute the members of these families. Several kinds of non-uniform complexity classes have been introduced. One kind includes a counterpart of the well-known complexity class P/poly and another kind includes a counterpart of the well-known complexity class NP/poly (see [4]).

- Work contributing to the conceptual analysis of the notion of an algorithm. Two equivalence relations on instruction sequences have been defined, an algorithmic equivalence relation and a computational equivalence relation. The algorithmic equivalence relation captures to a reasonable degree the intuitive notion that two instruction sequences express the same algorithm. Any equivalence relation that captures the notion that two instruction sequences express the same algorithm to a higher degree must be finer than the computational equivalence relation (see [5]).

- Work showing that, in the case of computing the parity function on bit strings of length n, for each natural number n, shorter

instruction sequences are possible with the use of an auxiliary Boolean register than without the use of auxiliary Boolean registers. This result supports, in a setting where programs are instruction sequences acting on Boolean registers, a basic intuition behind the storage of auxiliary data, namely the intuition that this makes possible a reduction of the size of a program (see [6]).

- Work providing mathematically precise alternatives to the natural language and pseudo code descriptions of the long multiplication algorithm and the Karatsuba multiplication algorithm. One established result is that the instruction sequence expressing the latter algorithm is shorter than the instruction sequence expressing the former algorithm only if the length of the bit strings involved is greater than 256. Another result is that in a setting with backward jump instructions the long multiplication algorithm can be expressed by an instruction sequence that is shorter than both these instruction sequences if the length of the bit strings involved is greater than 2 (see [7]).

- Work showing that the problem of deciding whether an instruction sequence computes the function modeling the non-zeroness test on natural numbers less than 2^n with respect to their binary representation by bit strings of length n, for natural number n, can only be efficiently solved under the restriction that the length of the instruction sequence is close to the length of the shortest possible instruction sequences that compute this function (see [8]).

10 Concluding Remarks

We have presented the theory underlying a considerable part of the work done so far in a line of research in which issues relating to a wide variety of subjects from computer science are rigorously investigated thinking in terms of instruction sequences. The distinguishing feature of this presentation is that it is less involved than previous presenta-

tions. Sections 2, 3, and 4 concern the part of the presented theory that is relevant to all the work done so far in the line of research referred to.

The restriction to instructions that operate on Boolean registers is a classical restrictions in computer science. Other such classical restrictions are the restriction to instructions that operate on natural number registers in register machines and the restriction to instructions that operate on Turing tapes in Turing machines (see e.g. [21]). Adaptation of Sections 5, 6, and 7 to these restrictions is rather straightforward (cf. [12]).

Notice that we have fixed in Section 8, for each use of a Boolean register that must be distinguished to make precise what it means that a given instruction sequence computes a given partial function from $\mathbb{B}^n$ to $\mathbb{B}^m$ ($n, m \in \mathbb{N}$), the focus by which the Boolean register for that use is named. Because of this and the required generality, the possibility that the same Boolean register is used as both input register and output register is excluded. Exclusion of possibilities like this can be circumvented by abandoning the fixed assignment of foci to register uses and defining "t computes f" relative to an assignment of foci to register uses. This approach complicates matters, but seems indispensable to find conclusive answers to open questions like "what are the shortest instruction sequences that compute the function on bit strings of length n that models addition modulo 2^n on natural numbers less than 2^n, for $n \in \mathbb{N}_1$?".

The instruction sequences with instructions for Boolean registers considered in this paper constitute essentially a programming language in which all variables are Boolean variables. Such programming languages are actually used in toolkits for software model checking that make use of the abstract interpretation technique known as predicate abstraction (see e.g. [22], [23]).

Acknowledgements

We thank an anonymous referee for carefully reading a preliminary version of this paper, for pointing out some slips made in it, and for posing questions that have led to improvements of the presentation.

References

[1] C. A. Middelburg, "Instruction sequences as a theme in computer science," http://instructionsequence.wordpress.com/, 2015.

[2] J. A. Bergstra and C. A. Middelburg, *Instruction Sequences for Computer Science*, ser. Atlantis Studies in Computing. Amsterdam: Atlantis Press, 2012, vol. 2.

[3] J. A. Bergstra and C. A. Middelburg, "Instruction sequence expressions for the secure hash algorithm SHA-256," arXiv:1308.0219v7 [cs.PL], August 2013.

[4] J. A. Bergstra and C. A. Middelburg, "Instruction sequence based non-uniform complexity classes," *Scientific Annals of Computer Science*, vol. 24, no. 1, pp. 47–89, 2014.

[5] J. A. Bergstra and C. A. Middelburg, "On algorithmic equivalence of instruction sequences for computing bit string functions," *Fundamenta Informaticae*, vol. 138, no. 4, pp. 411–434, 2015.

[6] J. A. Bergstra and C. A. Middelburg, "Instruction sequence size complexity of parity," *Fundamenta Informaticae*, vol. 149, no. 3, pp. 297–309, 2016.

[7] J. A. Bergstra and C. A. Middelburg, "Instruction sequences expressing multiplication algorithms," To appear in Scientific Annals of Computer Science. Preprint: arXiv:1312.1529v4 [cs.PL], 2018.

[8] J. A. Bergstra and C. A. Middelburg, "On the complexity of the correctness problem for non-zeroness test instruction sequences," arXiv:1805.05845v1 [cs.LO], May 2018.

[9] J. A. Bergstra and C. A. Middelburg, "On instruction sets for Boolean registers in program algebra," *Scientific Annals of Computer Science*, vol. 26, no. 1, pp. 1–26, 2016.

[10] J. A. Bergstra and M. E. Loots, "Program algebra for sequential code," *Journal of Logic and Algebraic Programming*, vol. 51, no. 2, pp. 125–156, 2002.

[11] J. A. Bergstra and A. Ponse, "Combining programs and state machines," *Journal of Logic and Algebraic Programming*, vol. 51, no. 2, pp. 175–192, 2002.

[12] J. A. Bergstra and C. A. Middelburg, "Instruction sequence processing operators," *Acta Informatica*, vol. 49, no. 3, pp. 139–172, 2012.

[13] H. Ehrig and B. Mahr, *Fundamentals of Algebraic Specification I: Equations and Initial Semantics*, ser. EATCS Monographs. Berlin: Springer-Verlag, 1985, vol. 6.

[14] D. Sannella and A. Tarlecki, "Algebraic preliminaries," in *Algebraic Foundations of Systems Specification*, E. Astesiano, H.-J. Kreowski, and B. Krieg-Brückner, Eds. Berlin: Springer-Verlag, 1999, pp. 13–30.

[15] D. Sannella and A. Tarlecki, *Foundations of Algebraic Specification and Formal Software Development*, ser. Monographs in Theoretical Computer Science, An EATCS Series. Berlin: Springer-Verlag, 2012.

[16] M. Wirsing, "Algebraic specification," in *Handbook of Theoretical Computer Science*, J. van Leeuwen, Ed. Amsterdam: Elsevier, 1990, vol. B, pp. 675–788.

[17] R. J. van Glabbeek and F. W. Vaandrager, "Modular specification of process algebras," *Theoretical Computer Science*, vol. 113, no. 2, pp. 293–348, 1993.

[18] J. A. Bergstra and C. A. Middelburg, "Axioms for behavioural congruence of single-pass instruction sequences," *Scientific Annals of Computer Science*, vol. 27, no. 2, pp. 111–135, 2017.

[19] G. B. Steven, S. M. Gray, and R. G. Adams, "HARP: A parallel pipelined RISC processor," *Microprocessors and Microsystems*, vol. 13, no. 9, pp. 579–587, 1989.

[20] J. A. Bergstra and C. A. Middelburg, "On the expressiveness of single-pass instruction sequences," *Theory of Computing Systems*, vol. 50, no. 2, pp. 313–328, 2012.

[21] J. E. Hopcroft, R. Motwani, and J. D. Ullman, *Introduction to Automata Theory, Languages and Computation*, 3rd ed. Reading, MA: Addison-Wesley, 2007.

[22] T. Ball and S. K. Rajamani, "Bebop: A symbolic model checker for Boolean programs," in *SPIN 2000*, ser. Lecture Notes in Computer Science, K. Havelund, J. Penix, and W. Visser, Eds., vol. 1885. Springer-Verlag, 2000, pp. 113–130.

[23] T. Ball and S. K. Rajamani, "The SLAM toolkit," in *CAV 2001*, ser. Lecture Notes in Computer Science, G. Berry, H. Comon, and A. Finkel, Eds., vol. 2102. Springer-Verlag, 2001, pp. 260–264.

Jan A. Bergstra, Cornelis A. Middelburg, Received August 16, 2018

Jan A. Bergstra
Informatics Institute, Faculty of Science, University of Amsterdam
Science Park 904, 1098 XH Amsterdam, the Netherlands
E–mail: J.A.Bergstra@uva.nl

Cornelis A. Middelburg
Informatics Institute, Faculty of Science, University of Amsterdam
Science Park 904, 1098 XH Amsterdam, the Netherlands
E–mail: C.A.Middelburg@uva.nl

Computer Science Journal of Moldova, vol.**26**, no.3(78), 2018

Imbrication algebras – algebraic structures of nesting order

Ioachim M. Drugus, Volodymyr G. Skobelev

Abstract

This paper is about "imbrication algebras", universal algebras with one binary operator in their signature, the operator for formation of ordered pairs, called here "pairing operator", and with the "characteristic property of ordered pairs" as their sole axiom. These algebras have been earlier introduced by the first author as reducts of "aggregate algebras", universal algebras proposed as models for a set theory convenient for formalization of data structures. The term "aggregate" is used to generalize three fundamental notions of set theory: set, atom and ordered pair. Thus, this paper initiates the research of aggregate algebras by narrowing the focus to one type of their main reducts – the reduct which deals with ordered pairs.

Keywords: cancellative magma, Catalan number, Merkle tree, ordered pair, quasi-variety.

MSC 2010: 08A70, 68P05, 05C05.

1 Introduction

The full name of the algebras, which are the subject matter of current paper, is intended to be "*imbrication order algebras*". In [1], (p. 87), where this kind of algebraic structures was introduced, these algebras were referenced shorter, as "order algebras". In this paper, we will prefer for them another short term – "imbrication algebras". The main reason why different short forms of the same term are preferred in different situations is that the term denotes a very general notion, which manifests as different phenomena in more concrete settings.

The term "imbrication" is often used in linguistics and computer science interchangeably with the term "nesting". However, this term is also extensively used with a meaning different from "nesting" – a meaning conveyed by the words: "overlapping", "interlacing", "interweaving". The second meaning is more relevant to the topic of the paper [1], and the first meaning of "imbrication", that of "nesting", is perfectly relevant to the topic of current paper.

Whereas nesting is a phenomenon which might have a large number of manifestations, we will focus on one kind of such manifestations – the nesting of ordered pairs, and we will put this in precise terms in the next section.

2 The imbrication algebras

It is a wide practice to refer to a symbol of an operation as "operator", and this practice is convenient for universal algebra, since the signature consists of operation symbols, i.e. "operators". There is also a practice to refer as "operator" to a mapping from a space to another space (e.g. "linear operator"), and this practice is convenient because it uses one term – "operator", rather than two terms – "operation" and "operator". In this second case, the expression "symbol of the operator" stands for what is called "operator" in the first case.

An ordered pair $(a,\ b)$ can be treated as the result of application of a binary operation, which we will prefer to call "operator", both because it can be treated as an action upon symbols, a "syntactic operator" (see subsection 2.2), and because the ordered pairs make up a "space", to be more precise – a plane. We will call this operator "pairing operator", because a function which encodes an ordered pair of natural numbers into one natural number is usually called "pairing function" (Cantor defined one of the first pairing functions – the "Cantor's pairing function"). We will treat $(a,\ b)$ as the result of application of the pairing operator to the objects a and b, and since in the notation "$(a,\ b)$" a symbol of operation or operator (like here "$f(a,\ b)$") is missing, we will consider the "empty symbol" as the symbol of the pairing operator.

Definition 1. *An imbrication algebra is a universal algebra with a sole operator's symbol in its signature, the empty symbol, and with the universal closure of the formula below as its single axiom:*

$$(x,\ y) = (x',\ y') \to x = x' \ \&\ y = y'.$$

The axiom of the imbrication algebras is the property owned by ordered pairs defined in one set theory or another. No matter how these are defined, this property is called "characteristic property of the ordered pairs". Whereas in set theory this is treated as a property of ordered pairs, in algebra this should be treated as a property of the pairing operator. Going forward, the universal closure of the formula above will be referenced as "pairing axiom".

Even though in ZF set theory, the ordered pair is a notion defined through the notion of set, there are also set theories, like Bourbaki's set theory, where the pairing operator is in the signature and it has the pairing axiom.

The first example of an imbrication algebra given in this paper is the universe (of discourse) of ZF set theory (usually denoted as V), which needs to be equipped with a pairing operator (like the one defined by Kuratowski, see next paragraph), to form a universal algebra. This imbrication algebra has a proper class as its support, and thus, this is a "large algebra". There are many large imbrication algebras – such are the universes (of discourse) of various set theories equipped with a pairing operator, which, by definition of the notion of ordered pair, must satisfy the pairing axiom.

There are many pairing operators in set theory, among which the best known is the Kuratowski's paring operator, used to define an ordered pair (x, y) as $\{\{x\}, \{\{x, y\}\}\}$. It is easy to check that any two pairing operators which equip V define the same imbrication algebra.

2.1 The class of imbrication algebras

According to one of several equivalent definitions (see e.g. [4], p. 219), a class of universal algebras of same signature ("similar algebras") is a *quasivariety*, if it contains a one-element algebra and is closed under

isomorphisms, subalgebras and reduced products. A quasivariety is also closed under products, subdirect products, and ultrafilter products.

The following property of quasivarieties is important for any constructive approach used in algebra: if a class of universal algebras is a quasivariety, then proceeding from a subset of this class, regarded as a "generating basis", one can construct other algebras by applying the operations over algebras mentioned in previous sentence.

The next proposition could be named "theorem" because of its high importance to this domain of research.

Proposition. *The class of imbrication algebras is a quasi-variety.*

Proof. The pairing axiom is equivalent to the conjunction of the universal closures of the following two formulas:

$$(x, y) = (x', y) \to x = x',$$
$$(x, y) = (x, y') \to y = y'.$$

Unlike the pairing axiom, these two formulas are quasi-identities, i.e. each of them has the form of an implication, the antecedent of which is a conjunction of equations of two terms (here, there is only one equation in conjunction), and the consequent of which is one such equation. According to the Theorem 2.25 of [4], since any imbrication algebra satisfies a set of quasi-identities, this class is a quasi-variety.

Q.E.D.

2.2 Application domains of imbrication algebras

We have mentioned above important examples of implication algebras – the universes of set theories equipped with a pairing operator. In this section, we will discuss about two domains, which can be considered as "native land" of the imbrication algebras – domains, where imbrication algebras can be used as an apparatus. One of these domains is the syntax of languages, natural or artificial, and the other is "brain informatics" – a discipline preoccupied by modeling mental phenomena, in particular, mental structures. One can say that the mental structures are constructed also according to a certain kind of "syntax". Thus, it

sounds appropriate to say that imbrication algebras relate to syntax, and pairing operator is a "syntactic operator".

Imbrication algebras reflect a special kind of order – the order, which appears as a result of using *balanced* distribution of brackets (we prefer round brackets, parentheses) for the *complete* disambiguation of an expression. Notice, that in previous sentence, two words are emphasized – "balanced" and "complete". The word "balanced" is emphasized because the nesting order appears for balanced, and only for balanced, brackets – i.e. the balance is essential for nesting. The word "complete" is emphasized because a distribution of balanced brackets enclosing *two*, and only *two*, subterms is essential for the nesting order called "imbrication".

To *completely* disambiguate an expression of form "$a_1 * ... * a_n$", where "*" is an operator, one needs to apply, in steps, a process of enclosing a *pair* of adjacent terms – subexpressions processed in this manner at a previous step, a process which can be called "pairing". The imbrication algebras can be described as algebras reflecting *complete* disambiguation by using pairing, and not partial disambiguation done by the use of an arbitrary distribution of balanced parentheses.

Whereas an application domain of imbrication algebras is the syntax of languages (natural or artificial), these algebras are most useful for the practice of grouping the subexpressions of an expression. The process of grouping (in particular, of grouping done by pairing) is sometimes referenced as "association" like in the term "left (right) association rule". For generality sake, we will refer to a balanced distribution of brackets as "association pattern". Thus, *left association* or *right association* are *association patterns*.

In [2], an approach to data called "Atomification-Aggregation-Association approach" (see also [3]) denoted as "A3" was introduced to serve as an alternative to currently widely used "Entity-Relationship" approach denoted as "ER". The A3 approach was proposed as a data model for "mental content" – a concept which makes sense in brain informatics. The A3 approach presupposes that all data structures are built by iterative application of three mental operations, one of which associates one entity *to* another entity (the order is important) and,

in this manner, builds an "association pair" (an *association pair* is an *association pattern* – the simplest association pattern).

The imbrication algebras explicate the algebraic aspect of the data structures built by iterative application of the "association operation" of A3 approach. We will refer to such structures simply, as "associations". The fact that an entity is associated with another entity can be imaged graphically by representing the entities as small cirles connected by arrows. In such a representation, when an entity is associated with itself, the arrow has a source coinciding with the target, and the direction of the arrow does not matter, so that one can drop the arrowhead and just use a non-oriented loop.

However, associations differ from those data structures which can be represented by directed graphs, since an association can, in turn, be associated with another association, like in the diagram in Figure 1, where the loop in b is associated with the node c and this association pair is, in turn, the target of another association. Since the kind of graphs which allow such "imbrication" differs from "directed graphs", they require a name and we will refer to them as "association graphs".

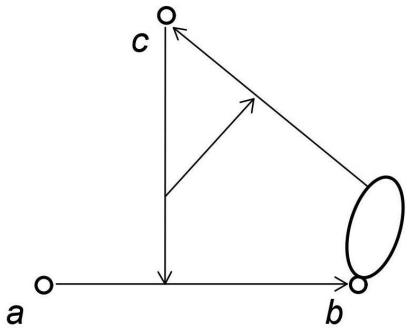


Figure 1. The diagram of the association $((c,(a,b)),((b,b),c))$

3 Imbrication algebras and the order theory

It is common to treat the preorder (also called quasi-order), as a binary relation, and this treatment is considered as the widest explication of the conception of order. However the notion of "cyclic order", a kind of order known to humanity since the expression "clockwise" and "counter-clockwise" were coined, cannot be explicated in terms of preorder. Therefore, the algebraists use a ternary relation to explicate the conception of cyclic order. Since there might exist also other kinds of order which cannot be expressed like preorder, the term "order theory" was coined to cover all phenomena called "order". However, there does not seem to exist a discipline "under this name" with a unified approach to the conception of order.

As earlier mentioned, the imbrication algebras were introduced in [1], where these were referenced as "order algebras". The reason for using the word "order" within this compound term was that in [1] these algebras were treated as reducts of "aggregate algebras" – a kind of algebraic structures intended for algebraization of a special set theory. The term "order algebra" was used in that paper for two reasons: (a) the term refers to an algebra with one operator – that of formation of the *ordered* pair and (b) the notion of ordered pair is used in all definitions of order within the aggregate algebras, where you can also use the concept of set.

In the "aggregate theory" of [1], alongside the paring operator, there are also the operator of formation of a singleton $\{x\}$, and the operation of union of two sets. The aggregate theory can be extended by introducing the operation of union for a family of sets, so that the theory treats arbitrary (i.e. also infinite) aggregates. Thus, the notion of preoder can be easily expressed in terms of aggregate theory, and thus all the theories about the "classical" explication of order in terms of a binary relation can have the aggregate theory as a foundation.

The pairing operator adds a new type of order to aggregate theory – the imbrication order. This is the order of nesting in expressions and the order of the ordered binary trees. There can be defined also other orders – orders, which "mix up" the classic order and the imbrication

order. This raises hopes that the aggregate theory can serve as an appropriate foundation for the discipline called "order theory".

As an exercise, one can try to explicate in terms of imbrication algebra, without involving set theoretic operations, the conception of "clockwise order". An explication of this is given by the association ((...((1,2),3),...), 12), but one should expect that this kind of order can also be defined by any other of 11 associations obtained from this one by a circular substitution. There can be used also other association patterns for explication of clockwise order. The choice of one explication or another is necessarily *a matter of convention*, similar to how the Kuratowski's definition of ordered pair is a matter of convention.

4 Imbrication algebras and Jónsson-Tarski algebras

For an ordered pair $a = (u, v)$, denote u as a^+ and v as $a^\times$ (these are the original notations used in [5]). One can temporarily call "projections" the two maps, $x \mapsto x^+$ and $x \mapsto x^\times$, even though the expressions like "left (right) projection" or "first (second) projection" make little sense. The "characteristic property of ordered pairs" guarantees that the projections are univocal maps, where they are defined, but nothing in the definition of imbrication algebra guarantees that the projections are defined for all elements of the algebra.

Definition 2. *An imbrication algebra with both projections defined for all its elements is called Jónsson-Tarski algebra.*

The algebras introduced in [5] and called here "Jónsson-Tarski algebras" are examples of a proper subclass of the class of imbrication algebras which is known to mathematical community.

5 Algebraic closure of pairing operator

Let A be any fixed non-empty set. We consider the binary operator, the value of which for any elements $x, y \in A$ (taken in this order) is

the object denoted (x, y) (it is supposed that the symbols "(", "," and ")" are not elements of the set A). It is evident that the object (x, y) $(x, y \in A)$ can be treated as an ordered pair, if we accept the following axiom postulating the properties of such objects:

$$(\forall x_1, x_2, y_1, y_2 \in A)((x_1, y_1) = (x_2, y_2) \Leftrightarrow x_1 = x_2 \& y_1 = y_2). \quad (1)$$

We define

$$A^{(0)} = A, \quad (2)$$

$$A^{(n)} = \bigcup_{i=0}^{n-1} \{(x, y) | x \in A^{(i)}, y \in A^{(n-1-i)}\} \ (n = 1, 2, \dots). \quad (3)$$

For each non-empty set A formulae (2) and (3) define inductively the sequence

$$A^{(0)}, A^{(1)}, \dots, A^{(n)}, \dots \quad (4)$$

of non-empty sets. Due to this, the axiom (2) can be extended from the set $A = A^{(0)}$ onto the set

$$\mathcal{A} = \bigcup_{n=0}^{\infty} A^{(n)}, \quad (5)$$

i.e. the following axiom can be accepted:

$$(\forall x_1, x_2, y_1, y_2 \in \mathcal{A})((x_1, y_1) = (x_2, y_2) \Leftrightarrow x_1 = x_2 \& y_1 = y_2). \quad (6)$$

Due to formulae (2)-(6) the following three propositions are true.

Proposition 1. *For each non-empty set A, if $(x_1, y_1) = (x_2, y_2)$ $(x_1, x_2, y_1, y_2 \in \mathcal{A})$, then there exist the single non-negative integers i and j, such that $x_1, x_2 \in A^{(i)}$ and $y_1, y_2 \in A^{(j)}$.*

Proposition 2. *For each non-empty set A the sequence (4) consists of non-empty pair-wise non-intersecting sets.*

Proposition 3. *For each non-empty set A the set $\mathcal{A}$ is an infinite set.*

Proceeding from formulae (2)-(6), we can define for each non-empty set A the A-associated magma

$$\mathcal{M}_A = (\mathcal{A}, \circ), \quad (7)$$

such that

$$x \circ y = (x, y) \tag{8}$$

for all $x, y \in \mathcal{A}$.

Now we establish the basic characteristics of the A-associated magma $\mathcal{M}_A = (\mathcal{A}, \circ)$, i.e. those ones, that are true for each non-empty set A.

Theorem 1. *For each non-empty set A the binary operation in the A-associated magma $\mathcal{M}_A = (\mathcal{A}, \circ)$ is a surjection $\circ : \mathcal{A} \times \mathcal{A} \to \bigcup\limits_{n=1}^{\infty} A^{(n)}$.*

Proof. Let A be any non-empty set.

Due to formula (5), for any elements $x, y \in \mathcal{A}$ there exist the single non-negative integers i and j, such that $x \in A^{(i)}$ and $y \in A^{(j)}$.

Due to formulae (3) and (8), we get that $x \circ y = (x, y) \in A^{(i+j+1)}$.

Since i and j are non-negative integers, then $i + j + 1$ is a positive integer. Thus, $x \circ y \in \bigcup\limits_{n=1}^{\infty} A^{(n)}$ for all $x, y \in \mathcal{A}$, i.e. the inclusion $\mathrm{Val}\circ \subseteq \bigcup\limits_{n=1}^{\infty} A^{(n)}$ holds.

Let z be any element of the set $\bigcup\limits_{n=1}^{\infty} A^{(n)}$. Then (see Proposition 2) there exists the single positive integer n, such that $z \in A^{(n)}$.

Due to Proposition 2, and formulae (3) and (8), there exists the single non-negative integer $i \leq n - 1$, such that $z = (x, y) = x \circ y$, where $x \in A^{(i)}$ and $y \in A^{(n-1-i)}$. Thus, $z \in \mathrm{Val}\circ$ for any $z \in \bigcup\limits_{n=1}^{\infty} A^{(n)}$, i.e. the inclusion $\bigcup\limits_{n=1}^{\infty} A^{(n)} \subseteq \mathrm{Val}\circ$ holds.

Inclusions $\mathrm{Val}\circ \subseteq \bigcup\limits_{n=1}^{\infty} A^{(n)}$ and $\bigcup\limits_{n=1}^{\infty} A^{(n)} \subseteq \mathrm{Val}\circ$ imply that the identity $\mathrm{Val}\circ = \bigcup\limits_{n=1}^{\infty} A^{(n)}$ holds, i.e. the mapping $\circ : \mathcal{A} \times \mathcal{A} \to \bigcup\limits_{n=1}^{\infty} A^{(n)}$ is some surjection.

Q.E.D.

Theorem 2. *For each non-empty set A the A-associated magma $\mathcal{M}_A = (\mathcal{A}, \circ)$ is a cancellative magma.*

Proof. Let A be any non-empty set.

Formulae (6) and (8) imply that for any $x, y, z \in \mathcal{A}$

$$x \circ y = x \circ z \Leftrightarrow (x, y) = (x, z) \Leftrightarrow x = x \& y = z \Leftrightarrow y = z,$$

i.e. $\mathcal{M}_A = (\mathcal{A}, \circ)$ is a left-cancellative magma.

Similarly, formulae (6) and (8) imply that for any $x, y, z \in \mathcal{A}$

$$x \circ y = z \circ y \Leftrightarrow (x, y) = (z, y) \Leftrightarrow x = z \& y = y \Leftrightarrow x = z,$$

i.e. $\mathcal{M}_A = (\mathcal{A}, \circ)$ is a right-cancellative magma.

Thus, the A-associated magma $\mathcal{M}_A = (\mathcal{A}, \circ)$ is a left-cancellative and a right-cancellative, both. Due to this factor, the A-associated magma $\mathcal{M}_A = (\mathcal{A}, \circ)$ is a cancellative magma.

Q.E.D.

Theorems 1 and 2 imply that the following proposition is true.

Proposition 4. *For each non-empty set A the A-associated magma $\mathcal{M}_A = (\mathcal{A}, \circ)$ is not a quasigroup.*

Remark 1. In proof of Theorem 1 it has been pointed that for any $i, j = 0, 1, \ldots$, if $x \in A^{(i)}$ and $y \in A^{(j)}$, then $x \circ y = (x, y) \in A^{(i+j+1)}$. This factor implies that for any fixed non-negative integers i and n, such that $n \leq i$, if $a \in A^{(i)}$ and $b \in A^{(n)}$, then each of the equations $a \circ x = b$ and $y \circ a = b$ has no solutions in the A-associated magma $\mathcal{M}_A = (\mathcal{A}, \circ)$.

6 The interrelation between elements of the set $\mathcal{A}$ and finite binary trees

Let A be any fixed non-empty set.

Each element $w \in \mathcal{A}$ can be uniquely presented by the rooted labeled finite binary tree $\mathcal{D}_w$, designed by the following procedure (V is

the set of vertices, $v^{(R)}$ is the root, E is the set of arcs, $f : V \to \mathcal{A}$ is the labeling mapping):

L-TREE
Input: a string $w \in \mathcal{A}$.
Output: the tree $\mathcal{D}_w$.

begin;
$V := \{v^{(R)}\}$;
$f(v^{(R)}) := w$;
$E := \emptyset$;
M_1: **if** $V = \emptyset$
then HALT,
else go to M_2;
M_2: **select** $v \in V$;
$V := V \setminus \{v\}$;
if $f(v) \in A$
then go to M_1,
else (in this case $f(v) = (u_L, u_R)$, where $u_L, u_R \in \mathcal{A}$)
$V := V \cup \{v_L, v_R\}$;
$E := E \cup \{(v, v_L), (v, v_R)\}$;
$f(v_L) := u_L$;
$f(v_R) := u_R$;
go to M_1;
end;

It is evident that the procedure L-TREE is some variant of a top-down parser. Due to formulae (2)-(5), it terminates for any element $w \in \mathcal{A}$, and its output is the tree $\mathcal{D}_w$.

Remark 2. In any tree $\mathcal{D}_w$ ($w \in \mathcal{A}$) each vertex either has two sons (if it is an internal vertex), or has no sons (if it is a leaf).

We set

$$\mathfrak{D}_{\mathcal{A}} = \{\mathcal{D}_w | w \in \mathcal{A}\}. \tag{9}$$

Due to formulae (2)-(6), the following two propositions are true.

Proposition 5. *For each non-empty set A an element $\mathcal{D}_w \in \mathfrak{D}_{\mathcal{A}}$ is a rooted labeled binary tree with n ($n = 2, 3, \dots$) leafs if and only if there are n appearances of elements of the set A in the string w.*

Proposition 6. *For each non-empty set A the following formula holds*

$$(\forall w_1, w_2 \in \mathcal{A})(w_1 \neq w_2 \Leftrightarrow \mathcal{D}_{w_1} \neq \mathcal{D}_{w_2}). \tag{10}$$

Remark 3. Formulae (9) and (10) imply that for each non-empty set A, the sets $\mathcal{A}$ and $\mathfrak{D}_{\mathcal{A}}$ have the same cardinality.

The more subtle characteristics of the structure of the set $\mathcal{A}$ can be established as follows.

Let $\tilde{\mathcal{D}}_w$ ($w \in \mathcal{A}$) be the rooted unlabeled binary tree that can be obtained by erasing all labels of vertices in the tree $\mathcal{D}_w$. It is evident that for each non-empty set A, the unlabeled rooted binary tree $\tilde{\mathcal{D}}_w$ ($w \in \mathcal{A}$) can be treated as the structure of the string $w \in \mathcal{A}$.

We set

$$\tilde{\mathfrak{D}}_{\mathcal{A}} = \{\tilde{\mathcal{D}}_w | w \in \mathcal{A}\}. \tag{11}$$

and define the mapping $\psi_{\mathcal{A}} : \mathfrak{D}_{\mathcal{A}} \to \tilde{\mathfrak{D}}_{\mathcal{A}}$ by the identity:

$$\psi_{\mathcal{A}}(\mathcal{D}_w) = \tilde{\mathcal{D}}_w \ \ (w \in \mathcal{A}). \tag{12}$$

Formulae (11), (12), and Proposition 6 imply that for each non-empty set A, the elements of the factor-set $\mathfrak{D}_{\mathcal{A}}/\ker\psi_{\mathcal{A}}$ define the sets of all strings $w \in \mathcal{A}$ with the same structure $\tilde{\mathcal{D}}_w$.

Remark 4. For each non-empty set A, the set $\tilde{\mathfrak{D}}_{\mathcal{A}}$ is a countable set, since it is the set of all non-empty finite rooted binary trees, such that each vertex either has two sons, or has no sons. Due to this factor, in what follows, we will omit the subscript $\mathcal{A}$, i.e. we will write $\tilde{\mathfrak{D}}$, since for any non-empty sets A_1 and A_2 the identity $\tilde{\mathfrak{D}}_{\mathcal{A}_1} = \tilde{\mathfrak{D}}_{\mathcal{A}_2}$ holds.

Due to Proposition 5, we can define on the set $\tilde{\mathfrak{D}}$ the partition $\pi = \{B_n | n = 2, 3, \dots\}$ as follows: B_n ($n = 2, 3, \dots$) consists of all elements of the set $\tilde{\mathfrak{D}}$ that are unlabeled rooted binary trees with n leafs. It is well known that the following proposition is true.

Proposition 7. *For any* $n = 2, 3, \dots$

$$|B_n| = C_{n-1},$$

where C_{n-1} *is the Catalan number.*

Remark 5. The Catalan numbers can be computed by formula:

$$C_n = \frac{1}{n+1}\binom{2n}{n} \quad (n = 1, 2, \dots).$$

The structure of the strings $w \in \mathcal{A}$ has been identified above with the finite rooted unlabeled binary trees $\tilde{\mathcal{D}}_w \in \tilde{\mathfrak{D}}$. Another (equivalent) model for the structure of the strings $w \in \mathcal{A}$ can be designed as follows.

Let $B = A \cup \{\ ,\}$, and $del_B(w)$ $(w \in \mathcal{A})$ be the string obtained from the string w by deleting all letters $b \in B$. We set

$$del_B(\mathcal{A}) = \{del_B(w) | w \in \mathcal{A}\}.$$

The language $del_B(\mathcal{A})$ can be characterized as follows.

Proposition 8. *For any non-empty set* A *the language* $del_B(\mathcal{A})$ *is some proper non-empty sub-language of the Dyck language* $L_{D(2)}$ *over the* 2*-letters alphabet.*

Proof. The Dyck language $L_{D(2)}$ over the 2-letters alphabet $\{\alpha, \beta\}$ is the context-free language that can be generated, for example, by the following two production rules (S is the single non-terminal symbol, and λ is the empty symbol): $S \to \lambda$ and $S \to \alpha S \beta S$.

Identifying the symbol α with the opening parentheses (, and the symbol β with the closing parentheses), we get that $del_B(\mathcal{A}) \subseteq L_{D(2)}$.

Since $() \in del_B(\mathcal{A})$, we get that $del_B(\mathcal{A}) \neq \emptyset$.

Since $()() \in L_{D(2)}$ and $()() \notin del_B(\mathcal{A})$, we get that $del_B(\mathcal{A}) \subset L_{D(2)}$.

Q.E.D.

7 Applications of the A-associated magma

Let A be any non-empty set.

We illustrate briefly, how the A-associated magma $\mathcal{M}_A = (\mathcal{A}, \circ)$ can be used as some conceptual model in mathematics and its applications.

Example 1. The operation $\circ$ can be naturally extended on the power set $\mathcal{B}(\mathcal{A})$ as follows:

$$X \circ Y = \{(x, y) | x \in X \& y \in Y\} \tag{13}$$

for any subsets $X, Y \in \mathcal{B}(\mathcal{A})$.

It is evident that formula (13) defines the operation of the Cartesian product on the set $\mathcal{A}$.

Thus, the operation of the Cartesian product can be treated as some binary operation over some magma that satisfies Theorems 1 and 2, and Proposition 4.

Example 2. Let $\diamond$ be any binary operation defined on the set A. i.e. $\mathcal{M}_\diamond = (A, \diamond)$ is any magma with the carrier A. The A-associated magma $\mathcal{M}_A = (\mathcal{A}, \circ)$ can be treated as the formal presentation for the magma $\mathcal{M}_\diamond = (A, \diamond)$ of all possible results for the operation $\diamond$ over the strings of elements of the set A as follows.

Let $a_1, \ldots, a_n \in A$ $(n \geq 2)$. We can select any finite rooted unlabeled binary tree $\tilde{\mathcal{D}} \in \tilde{\mathfrak{D}}$ with n leafs, and label the leafs, from left to right, by the elements $a_1, \ldots, a_n$. Thus, we have defined the order for the execution of operation $\diamond$ over the string $a_1 \diamond \cdots \diamond a_n$.

Using down-top parsing, we can label all internal vertices of the selected tree due to the following rule: if for the internal vertex v its left son v_L is labeled by the element $b_1 \in A$ and its right son v_R is labeled by the element $b_2 \in A$, then the vertex v is labeled by the element $b_1 \diamond b_2 \in A$.

It is evident that the label of the root of the selected tree is the result of the operation $\diamond$ over the string $a_1 \diamond \cdots \diamond a_n$, when the order for execution of this operation is defined by the selected finite rooted unlabeled binary tree $\tilde{\mathcal{D}} \in \tilde{\mathfrak{D}}$ with n leafs, when the leafs are labeled, from left to right, by the elements $a_1, \ldots, a_n$.

Example 3. Any Merkle tree [7] is a rooted complete finite labeled binary tree, such that each vertex either has two sons, or has no sons.

The leafs of a Merkle tree are labeled by some data. Any internal vertex of a Merkle tree is labeled due to the following rule: if for the internal vertex v its left son v_L is labeled by the element b_1 and its right son v_R is labeled by the element b_2, then the vertex v is labeled by the element $HASH(b_1, b_2)$. Thus, the root of a Merkle tree is labeled by the hash of all the data in the tree.

The Merkle trees are applied for resolving various Problems in Cryptography, such as consistency and audit proofs, data synchronization, etc. Currently an incremental construction of the Merkle trees, i.e. uncomplete and growing Merkle trees, are applied for time-stamps based on a public-key cryptosystems [8].

Let A be the set of all strings that are either data, or hash values for some fixed hash-function $HASH$, and the binary operation $\diamond$ on the set A is computing the hash-value for concatenation of two strings. Then we get the situation, that has been considered in Example 2.

8 Conclusions

Same as above, in this final section, "A" is the denotation of an arbitrary "fixed" set, and "$\mathcal{A}$" denotes both a superset of "A" and the magma with this superset as its support – a magma, which is referenced as "A-associated magma". These denotations introduced in section 5 will be used with the same meaning here.

This paper presents the results of a first attempt to characterize the imbrication algebras in terms of universal algebra. Among these results, the most noteworthy is elucidation of a number of deep internal links between any imbrication algebra and the infinite magma A.

The importance of the A-associated magma is that it can serve as a theoretical basis for the research of various algebraic systems with a single binary operation – algebraic systems, which are also referenced as "(algebraic) binary structures". Also, taking into account the established here interrelationship between the elements of the superset $\mathcal{A}$ of the set A and the finite binary trees, one can expect that imbrication algebras can serve as a theoretical foundation for the implementation of provers and solvers, intended to deal with various binary structures.

The development of the structure of provers and solvers can be envisioned as one of the main directions of future research. In this regard, many problems arise naturally – problems, which have an importance on their own. Some of such problems are listed immediately below.

Firstly, this is the problem of software implementation of tokenization and parsing. The set A, which is "fixed" in this paper, can be either finite or countable, and it is natural to assume that the elements of the set A are always presented as consecutive positive binary integers. Therefore, the problem of tokenization can be formulated as follows: *Is it true that, for any finite or countable set A, and a given string w, the string w consists only of symbols "0", "1", "(", "," and ")"*? In case of an affirmative answer, the problem of parsing can be formulated like this: *for any finite or countable set A, and any given string w, is it true that $w \in \mathcal{A}$*? Possibly, the most effective is such an interaction of the two modules, when the parsing module calls, when necessary, the tokenization module.

Secondly, this is the problem of software implementation of checking of the parsing equivalence for the strings $w_1, w_2 \in \mathcal{A}$. This problem can be formulated like this: *given an arbitrary finite or countable set A, and two strings $w_1, w_2 \in \mathcal{A}$, is it true that $\tilde{\mathcal{D}}_{w_1} = \tilde{\mathcal{D}}_{w_1}$*?

Thirdly, this is the problem of software implementation for generation of all solutions of an equation in a given magma.

Fourthly, this is the problem of software implementation for generating all generalized Merkle trees (for the fixed hash function) of the given height, with the same label of the root.

References

[1] I. Drugus, "Towards an Algebraic Set Theory for the Formalization of Data Structures," in *Proc. Conf. on Mathematical Foundations of Informatics MFOI2018, July 2-6, 2018*, (Chisinau, Moldova), 2018, pp. 73–89.

[2] I. Drugus, "A Wholebrain approach to the Web," in *Proc. of Web Intelligence – Intelligent Agent Technology Conference*, (Silicon Valley), 2007, pp. 68–71.

[3] I. Drugus, "Universics: a Common Formalization Framework for Brain Informatics and Semantic Web," in *Web Intelligence and Intelligent Agents*, Vucovar: InTech Publishers, 2010, pp. 55–78.

[4] S. Burris and H. P. Sankappanavar, *A course in universal algebra* (Graduate Texts in Mathematics, vol. 78), 1st ed., New-York, Heidelberg, Berlin: Springer-Verlag, 1981, 276 p. ISBN: 0-387-90578-2.

[5] B. Jónsson and A. Tarski, "On two properties of free algebras," *Mathematica Scandinavica*, vol. 9, 1961, pp. 95–101. https://doi.org/10.7146/math.scand.a-10627.

[6] V.A. Shcherbacov, *Elements of quasigroup theory and applications*, Boca Raton, Florida, USA: CRC Press, 2017, 576 p.

[7] A. Narayanan, J. Bonneau, E. Felten, A. Miller, and S. Goldfeder, *Bitcoin and Cryptocurrency Technologies: A Comprehensive Introduction*, Princeton, NJ, USA: Princeton University Press, 2016, 336 p.

[8] M. Ogava, E. Horita, and S. Ono, "Proving properties of incremental Merkel trees," *LNAI*, vol. 3632, 2005, pp. 424–440.

Ioachim M. Drugus, Volodymyr G. Skobelev Received September 30, 2018

Ioachim M. Drugus
Institute of Mathematics and Computer Science
5 Academiei str., MD-2028, Chisinau, Republic of Moldova
Phone: +373 699 79 938
E–mail: ioachim.drugus@math.md

Volodymyr G. Skobelev
V.M. Glushkov Institute of Cybernetics of NAS of Ukraine
40 Glushkova ave., Kyiv, Ukraine, 03187
Phone: +38 063 431 86 05
E–mail: skobelevvg@gmail.com

Computer Science Journal of Moldova, vol.**26**, no.3(78), 2018

Closeness centrality in some splitting networks

Vecdi Aytaç Tufan Turacı

Abstract

A central issue in the analysis of complex networks is the assessment of their robustness and vulnerability. A variety of measures have been proposed in the literature to quantify the robustness of networks, and a number of graph-theoretic parameters have been used to derive formulas for calculating network reliability. *Centrality* parameters play an important role in the field of network analysis. Numerous studies have proposed and analyzed several *centrality* measures. We consider *closeness centrality* which is defined as the total graph-theoretic distance to all other vertices in the graph. In this paper, closeness centrality of some splitting graphs is calculated, and exact values are obtained.

Keywords: Graph theory, network design and communication, complex networks, closeness centrality, splitting graphs.

1 Introduction

Most of the communication systems of the real world can be represented as complex networks, in which the nodes are the elementary components of the system and the links connect a pair of nodes that mutually interact exchanging information. A central issue in the analysis of complex networks is the assessment of their robustness and vulnerability. One of the major concerns of network analysis is the definition of the concept of *centrality*. This concept measures the importance of a node's position in a network. In social, biological, communication, and transportation networks, among others, it is important to know the relative structural prominence of nodes to identify the key elements in the network.

There are several *centrality* measures like *degree centrality, closeness centrality, residual closeness, vertex-betweenness centrality, edge-betweenness centrality*, etc. [1]–[9]. *Centrality* is a complex notion that requires a clear definition. For example, a node has high centrality if it can communicate directly with many other nodes, or if it serves as an intermediary point in communication among other nodes. *Degree centrality* is defined as the number of links incident upon a node. It is a straightforward and efficient metric; however, it is less relevant since a node having a few high influential neighbors may have much higher influence than a node having a larger number of less influential neighbors. A node with larger degree is likely to have higher influence than a node with smaller degree. However, in some cases, this method fails to identify influential nodes since it considers only very limited information. For example, as it is shown in Figure 1, although node 1 has the largest degree among all 15 nodes, the information, if it origins at node 1, may not spread the fastest or the most broadly since all neighbors of node 1 have a very low degree. In contrast, node 15 may be of higher influence although it has lower degree comparing with node 1.

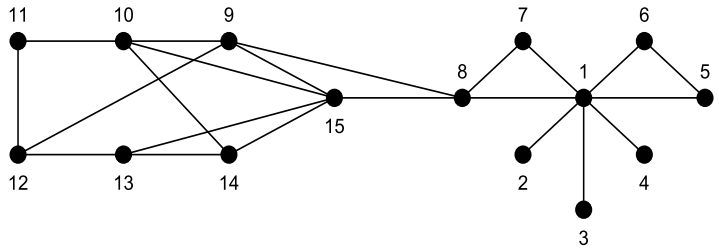


Figure 1. A graph with 15-vertices and 21-edges

Although some well-known global metrics such as *betweenness* and *closeness* centralities can give better results, due to the very high computational complexity, they are not easy to manage very large-scale online social webs [2], [5], [8]. For a network $G = (V, E)$ with $n = |V|$ nodes and $m = |E|$ edges, time complexities of betweenness and close-

ness centralities are $O(nm)$ and $O(n^3)$, respectively [10]. Closeness can be considered as a measure of how long it will spread information from a given node to other reachable nodes in the network. Calculation of closeness for simple graph types is important because if one can break a more complex network into smaller networks, then under some conditions the solutions for the optimization problem on the smaller networks can be combined to a solution for the optimization problem on the larger network and by calculating the closeness centrality for some real networks very good practical results can be achieved. Thus, we want to find exact values, upper bounds or lower bounds for metrics which are difficult to compute.

Graph theory has become one of the most powerful mathematical tools in the analysis and study of the architecture of a network. As usual, a network is described by an undirected simple graph. For example, if we want to occur a computer network with graph topologies the following correspondences are used: vertices represent computers and edges represent connections between computers.

We consider simple finite undirected graphs without loops and multiple edges in this paper. Let $G = (V; E)$ be a graph with vertex set V and edge set E. We also specify the vertex set of G by $V(G)$ and the edge set by $E(G)$ instead of V and E, respectively. For vertices u and x of a graph G, the *open neighborhood* of u is $N(u) = \{v \in V(G) \,|uv \in E(G)\}$ and $N_x(u) = \{v \in V(G - \mathrm{x}) \,|uv \in E(G - \mathrm{x})\}$, the $G - x$ graph corresponds to the graph from which the x vertex is removed from G. The *closed neighborhood* of u is $N[u] = N(u) \cup \{u\}$. For a set $S \subseteq V(G)$, its open neighborhood is the set $N(S) = \bigcup_{v \in S} N(v)$, and its closed neighborhood is the set $N[S] = N(S) \cup S$. A set $S \subseteq V(G)$ is a dominating set of G, if $N[S] = V(G)$, that is a set $S \subseteq V(G)$ is a *dominating set* if every vertex in $V(G) - S$ is adjacent to at least one vertex in S. The *domination number* $\gamma(G)$ is defined as the minimum cardinality of a dominating set of G. The *diameter* of G, denoted by $diam(G)$ is the largest distance between two vertices in $V(G)$. The *degree* $\deg_G(v)$ of a vertex $v \in V(G)$ is the number of edges incident to v. The *maximum degree* of G is $\Delta(G) = \max\{\deg_G(v) \,|v \in V(G)\}$ and the *minimum*

degree of G is $\delta(G) = \min\{\deg_G(v) \,|v \in V(G)\}$. The *distance* $d(u, v)$ between two vertices u and v in G is the length of a shortest path between them. If u and v are not connected, then $d(u, v) = \infty$, and for $u = u$, $d(u, u) = 0$ [11].

Our aim in this paper is to consider the computing of the closeness centrality of some splitting graphs. In Section 2, some definitions and results for closeness centrality are given. In Section 3, closeness centralities of some splitting graphs are determined.

2 Closeness centrality

In a network, even nodes with the same available resources, vary in their importance due to their different locations. It is reasonable to choose, firstly, the substrate nodes with the same available resources in a more important location. Furthermore, the importance of a node is more complex when the network is dynamically changing. The current states of all the elements in the global network determine the importance of a node. *Centrality* analysis provides effective methods for measuring the importance of nodes in a complex network and it has been widely used in complex network analysis [5].

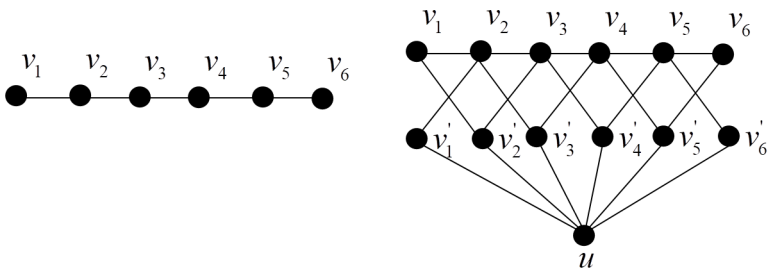
Closeness centrality can be considered as a measure of how long it will spread information from a given node to other reachable nodes in the network. *Closeness centrality* of node v is defined as the reciprocal of the sum of geodesic distances to all other nodes of $V(G)$. This definition was modified by Dangalchev in [3], [4]. The *closeness centrality* of a graph is defined as

$$C(G) = \sum_{v \in V(G)} C(v),$$

where $C(v)$ is the *closeness centrality* of a vertex v, and $C(v) = \sum_{t \in (V-v)} \frac{1}{2^{d(v,\,t)}}$ [4], [11]. Next, we give some results for closeness centrality.

Theorem 1. **[1], [3], [4], [9]** *The closeness centrality of*

a) the complete graph K_n with n vertices is $C(K_n) = (n(n-1))/2$;



(a) (b)

Figure 2. a) The graph P_6 (b) The graph $\mu(P_6)$

b) the star graph S_n with n vertices is $C(S_n) = (n-1)(n+2)/4$;

c) the path P_n with n vertices is $C(P_n) = 2n - 4 + 1/2^{n-2}$;

d) the cycle C_n with n vertices

$$C(C_n) = \begin{cases} n\left(2 - 3/2^{n/2}\right) & \text{, if } n \text{ is even;} \\ 2n\left(1 - 1/2^{(n-1)/2}\right) & \text{, if } n \text{ is odd;} \end{cases}$$

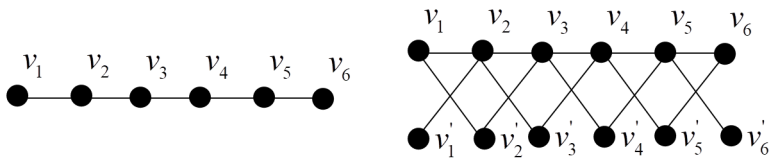
e) the wheel W_n with $n+1$ vertices is $C(W_n) = n(n+5)/4$;

f) the gear graph G_n with $2n+1$ vertices is $C(G_n) = n(9n+49)/16$;

g) the friendship graph f_n with $2n+1$ vertices is $C(f_n) = n(n+2)$;

h) the fan F_n with n vertices is $C(F_n) = \left(n^2 + 3n - 6\right)/4$.

Definition 1. **[12]** *Let G be a graph with vertex set $V(G) = \{v_1, v_2, ..., v_n\}$. The Mycielski graph of G, denoted $\mu(G)$, has for its vertex set $V(\mu(G)) = \{v_1, v_2, ..., v_n, v'_1, v'_2, ..., v'_n, u\}$. As for adjacency, v_i is adjacent with v_j in $\mu(G)$ if and only if v_i is adjacent with v_j in G, v_i is adjacent with v'_j in $\mu(G)$ if and only if v_i is adjacent with v_j in G, and v'_i is adjacent with u in $\mu(G)$ for all $i = \overline{1,n}$. We display the Mycielski graph $\mu(P_6)$ in Figure 2.*



(a) (b)

Figure 3. (a) The graph P_6 (b) The graph $S'(P_6)$

Theorem 2. **[6]** *The closeness centrality of Mycielski graphs of*

a) the cycle C_n with n vertices for $n \geq 8$ is $C(\mu(C_n)) = (9n^2 + 77n)/16$;

b) the wheel K_n with n vertices for $n \geq 3$ is $C(\mu(K_n)) = (7n^2+n)/4$;

c) the star graph S_n with n vertices for 4 is $C(\mu(S_n)) = (2n^2+5n-3)/2$;

Definition 2. **[13]** *For a graph G on vertices $V(G) = \{v_1, v_2, ..., v_n\}$ and edges $E(G)$, let splitting graph $S'(G)$ be the graph on vertices and edges $V(G) \cup V'(G) = \{v_1, v_2, ..., v_n, v'_1, v'_2, ..., v'_n\}$ and $E(G) \cup \{v_i v'_j | \, v_i v_j \in E(G)\}$, respectively. We display the Splitting graph $S'(P_6)$ in Figure 3.*

In this paper, the following notations will be used throughout the article to make the proof of the given theorems understandable. Let the vertex-set of graph $S'(G)$ be $V(S'(G)) = V_1 \cup V_2$, where:

V_1: The set contains the vertices of the graph G, that is, $V_1 = \{v_i \in V(G),\ 1 \leq i \leq n\}$.

V_2: The set contains the new vertices which are obtained by definition of splitting graph, that is, $V_2 = \{v'_i \in V(G'),\ 1 \leq i \leq n\}$

Lemma 1. **[11]** *Let G be any connected graph of order n and size m. Then,*

$$\sum_{i=1}^{n} \deg_G(v_i) = 2m.$$

Theorem 3. *Let G be any connected graph of order n and size m. If $\gamma(G) = 1$ and $\delta(G) \geq 2$, then $C(S'(G)) = (2n^2 + 3m - n)/2$.*

Proof. Since the structure of the splitting graph $S'(G)$ and definition of the domination number, the distance between two vertices is at most 2. Let $v_i \in V(G)$. Clearly, the distance from the vertex v_i to $2\deg_G(v_i)$-vertices is 1, similarly the distance from the vertex v_i to $((2n-1) - 2\deg_G(v_i))$-vertices is 2. Thus we get

$$C(v_i) = 2\deg_G(v_i)\, 2^{-1} + (2n - 1 - 2\ \deg_G(v_i))\ 2^{-2}$$

for every vertices of $v_i \in V_1$, where $i = \overline{1, n}$.

Similarly, we get

$$C(v'_i) = \deg_G(v'_i)\, 2^{-1} + \left(2n - 1 - \ \deg_G(v'_i)\right)\ 2^{-2}$$

for every vertices of $v'_i \in V_2$, where $i = \overline{1, n}$.
Thus,

$$\begin{aligned} C(S'(G)) &= \sum_{i=1}^{n} C(v_i) + \sum_{i=1}^{n} C(v'_i) \\ &= \sum_{i=1}^{n} \deg_G(v_i) + \frac{1}{4}\sum_{i=1}^{n}(2n - 1 - 2\deg_G(v_i)) + \frac{1}{2}\sum_{i=1}^{n} \deg_G(v'_i) \\ &\quad + \frac{1}{4}\sum_{i=1}^{n}(2n - 1 - \deg_G(v'_i)). \end{aligned}$$

By Lemma 1, we have

$$C(S'(G)) = 2m + \frac{2n^2 - n}{2} - \frac{2m}{4} = \frac{2n^2 + 3m - n}{2}.$$

□

Corollary 1. *Let G be any connected graph of order n and size m. If $\gamma(G) = 1$, then $C(S'(G)) \geq C(S'(S_n))$.*

Proof. Star graph S_n with n vertices provides the requirement of the theorem. $\gamma(S_n) = 1$ and star graph S_n has $(n-1)$-vertices with degree 1. Thus, we get lower bound of the $C(S'(G))$ of any graph G whose domination number is 1. As a result, $C(S'(G)) \geq C(S'(S_n))$ is obtained. □

Corollary 2. *Let G be any connected graph of order n. Then, $C(\mu(G) - \{u\}) = C(S'(G))$, where the vertex u is the vertex u of the definition of the Mycielski graph $\mu(G)$.*

Proof. It is clear. □

3 Calculation of closeness centrality of some splitting graphs

In this section, we consider the *closeness centrality* of the splitting graphs $S'(G)$ when G is a specified family of graphs.

Theorem 4. *The closeness centrality of $S'(C_n)$ is*

$$C(S'(C_n)) = \begin{cases} n\left(31/4 - 12/2^{n/2}\right) & \text{, if } n \text{ is even} \\ n\left(31/4 - 8/2^{(n-1)/2}\right) & \text{, if } n \text{ is odd.} \end{cases}$$

Proof. We have two cases depending on the vertices of $S'(C_n)$:

Case1. If n is even, then we have also two cases:

Case1.1. For any vertex of v_i $\left(i = \overline{1, n}\right)$ of V_1 in the graph $S'(C_n)$. Hence, the closeness centrality of subgraph $S'(C_n)\backslash V_2$ is

$$C(v_i) = \sum_{\substack{v_j \neq v_i \\ v_j \in V_1}} 2^{-d(v_i, v_j)} + \sum_{\substack{{v'}_j \neq v_i \\ {v'}_j \in V_2}} 2^{-d(v_i, {v'}_j)}, \tag{1}$$

where the first and second sums are denoted by C_1 and C_2, respectively, that is $C(v_i) = C_1 + C_2$. Furthermore, let S_1 be $nC(v_i)$. It is clear that C_1 is equal to the closeness centrality of the path C_n with n vertices for n even by Theorem 2(a).
To calculate C_2, the closeness centrality of each vertex v_i of V_1 is determined by the sum of the minimum distances from v_i to all the other vertices $v'_i \in V_2$ in the graph $S'(C_n)$. Hence, we obtain

$$\begin{aligned} C_2 &= 2/2^1 + 3/2^2 + 2/2^3 + ... + 1/2^{diam(C_n)} \\ &= 2/2^1 + 3/2^2 + 2/2^3 + ... + 2/2^{(n/2)-1} + 1/2^{n/2} \\ &= 2\sum_{j=1}^{\frac{n}{2}-1} 1/2^j + 1/2^2 + 1/2^{n/2}. \end{aligned} \tag{2}$$

To calculate the sum (1), we use the formula of the finite geometric series. As a result, we get, $C_2 = 9/4 - 3/2^{n/2}$.
Consequently, we get the following result:

$$\begin{aligned} S_1 &= n\left(2 - 3/2^{n/2}\right) + n\left(9/4 - 3/2^{n/2}\right) \\ &= 17n/4 - 6n/2^{n/2}. \end{aligned} \tag{3}$$

Case1.2. For any vertex of $v'_i\big(i = \overline{1,\ n}\big)$ of V_2 in the graph $S'(C_n)$. Hence, the closeness centrality of subgraph $S'(C_n)\backslash\ V_1$ is

$$C(v'_i) = \sum_{\substack{v_j \neq v'_i \\ v_j \in V_1}} 2^{-d(v'_i, v'_j)} + \sum_{\substack{v'_j \neq v'_i \\ v'_j \in V_2}} 2^{-d(v'_i, v'_j)}, \tag{4}$$

where the first and second sums are denoted by C'_1 and C'_2, respectively, that is $C(v'_i) = C'_1 + C'_2$. Let S_2 be $nC(v'_i)$. It is clear that C'_1 is $9/4 - 3/2^{n/2}$ by the Case 1.1. To calculate C'_2, we consider only the distance $d(v'_i, v'_j)$ between v'_i and v'_j in the graph $S'(C_n)$. Then we get C'_2

$$\begin{aligned} C'_2 &= 2/2^2 \ + 4/2^3 \ + 2/2^4 \ + 2/2^5 + \dots \\ &\quad + 2/2^{diam(C_n)-1} \ + 1/2^{diam(C_n)} \\ &= 2/2^2 \ \left(1 + 1/2^1 \ + 1/2^2 + \dots \right. \\ &\quad \left. + 1/2^{(n/2)-3} \ \right) + 2/2^3 \ + 1/2^{n/2}. \end{aligned} \tag{5}$$

To calculate the sum (5), we use the formula of the finite geometric series. So, we get $C'_2 = 5/4 \ - 3/2^{n/2}$. Consequently, we get the following result:

$$\begin{aligned} S_2 &= n\left(9/4 - 3/2^{n/2}\right) + n\left(5/4 - 3/2^{n/2}\right) \\ &= 7n/2 - 6n/2^{n/2}. \end{aligned} \tag{6}$$

Clearly, $C(S'(C_n)) = S_1 + S_2$. By summing (3) and (6), we obtain the closeness centrality of $S'(C_n)$ for n is even as follows:

$$\begin{aligned} C(S'(C_n)) &= 17n/4 - 6n/2^{n/2} + 7n/2 - 6n/2^{n/2} \ = \\ &= n\left(31/4 - 12/2^{n/2}\right). \end{aligned}$$

Case2. If n is odd, then the proof is similar to that of n is even. Hence, by using the Theorem 1.(d), the closeness centrality for each vertex v_i $\left(i = \overline{1,\ n}\right)$ of V_1 is $C(v_i) = 4\left(1 - 1/2^{(n-1)/2}\right) + 1/4$, and also the closeness centrality for each vertex $v'_i\left(i = \overline{1,\ n}\right)$ of V_2 is $C(v'_i) = 7/2 \ - 4/2^{(n-1)/2}$. Consequently, the closeness centrality of $S'(C_n)$ for n is odd,

$$C(S'(C_n)) = \ nC(v_i) + nC(v'_i) = \ n\left(31/4 - 8/2^{(n-1)/2}\right).$$

□

Theorem 5. *The closeness centrality of $S'(P_n)$ is*

$$C(S'(P_n)) = 4\left(2n - 4 + 1/2^{n-2}\right) - (n-3)/4.$$

Proof. The proof is similar to that of Theorem 4. Hence, similarly we have two cases depending on the vertices of $S'(P_n)$:

Case1. For any vertex of $v_i\big(i = \overline{1, n}\big)$ of V_1 in the graph $S'(P_n)$. Hence, the closeness centrality of subgraph $S'(P_n)\backslash V_2$ is as like the equation (1), where the first and second sums are denoted by C_1 and C_2, respectively, that is $C(v_i) = C_1 + C_2$. Furthermore, let S_1 be the sum of the all $C(v_i)$, where $i = \overline{1, n}$.
The value of C_2, due to the structure of the $S'(P_n)$ graph, is $C_2 = \sum\limits_{\substack{v_j \neq v_i \\ v_j \in V_1}} 2^{-d(v_i,v_j)} + n/2^2$. Thus, the closeness centrality of each vertex v_i in $S'(P_n)$ is

$$C(v_i) = 2 \sum_{\substack{v_j \neq v_i \\ v_j \in V_1}} 2^{-d(v_i,v_j)} + n/2^2. \tag{7}$$

It is clear that the equation (7) is equal to the closeness centrality of the path P_n with n vertices by Theorem 1.(c). Consequently, we get the following result:

$$S_1 = 2\left(2n - 4 + 1/2^{n-2}\right) + n/2^2. \tag{8}$$

Case2. For any vertex of $v'_i\big(i = \overline{1, n}\big)$ of V_2 in the graph $S'(P_n)$. Hence, the closeness centrality of subgraph $S'(P_n)\backslash V_1$ is as like the equation (4), where the first and second sums are denoted by C'_1 and C'_2, respectively, that is $C(v'_i) = C'_1 + C'_2$. Furthermore, let S_2 be the sum of the all $C(v'_i)$, where $i = \overline{1, n}$.
By the Case 1, we have

$$C'_1 = (9n - 16)/4 \ + 1/2^{n-2}. \tag{9}$$

To calculate C'_2, we get

$$C'_2 = 2(n-2)/2^2 + 2(n-3)/2^3 + ... + 2.2/2^{n-2} + 2.1/2^{n-1} + 2(n-2)/2^3 + 2/2^3.$$

By using Theorem 1.(c), we get

$$\begin{aligned} C'_2 &= \left(2n-4+1/2^{n-2}\right) - 2(n-1)/2^1 + (n-2)/2^2 + 1/2^2 \\ &= \left(2n-4+1/2^{n-2}\right) - 3(n-1)/4. \end{aligned} \tag{10}$$

Consequently, we get the following result:

$$\begin{aligned} S_2 &= (9n-16)/4 + 1/2^{n-2} + \left(2n-4+1/2^{n-2}\right) - 3(n-1)/4 \\ &= 2\left(2n-4+1/2^{n-2}\right) + (3-2n)/4. \end{aligned} \tag{11}$$

Clearly, $C(S'(P_n)) = S_1 + S_2$. By summing (8) and (11), we obtain the closeness centrality of $C(S'(P_n))$ as follows:

$$C(S'(P_n)) = 4\left(2n-4+1/2^{n-2}\right) + (n-3)/4.$$

□

Theorem 6. *The closeness centrality of $S'(W_n)$ is*

$$C(S'(W_n)) = \left(2n^2+9n+1\right)/2\ .$$

Proof. The proof is similar to that of Theorem 4. The wheel W_n with $n+1$ vertices contains an n-cycle and a central vertex c that is adjacent to all vertices of the cycle. Then, we have $\deg_G(v_c) = n$. Hence, we have two cases depending on the vertices of $S'(W_n)$:

Case1. For any vertex of $v_i\left(i = \overline{1,\ n+1}\right)$ of V_1 in the graph $S'(W_n)$. Clearly we have $|N(v_i)| = 6$, except the central vertex v_c of V_1. Thus, the closeness centrality of subgraph $S'(W_n)\backslash V_2$ is as like the equation (1), where the first and second sums are denoted by C_1 and C_2, respectively, that is $C(v_i) = C_1 + C_2$.
Furthermore, let S_1 be the sum of the all $C(v_i)$, where $i = \overline{1, n+1}$. It is clear that C_1 is equal to the closeness centrality of the path W_n by Theorem 2(b). To calculate C_2, we have two cases depending on the vertices of survival subgraph $S'(W_n)\backslash V_2$:

Case1.1. Let v_i be a vertex of an n-cycle. Since $N(v_i) = \{v_m, v_t, v_c, v'_m, v'_t, v'_c\}$, where v_m, v_t are the vertices of an n-cycle and v_c is the central vertex of $S'(W_n)\backslash V_2 = W_n$, also v_i is adjacent to v'_m, v'_t and v'_c in V_2. Each of these vertices, v'_m, v'_t and v'_c, is the vertex in the copy $S'(W_n)$. Since $d(v_i, v'_m) = d(v_i, v'_t) = d(v_i, v'_c) = 1$, v_i is at distance 2 to other $n-2$ vertices of V_2. Thus,

$$\begin{aligned} C(v_i) &= \sum_{\substack{v'_j \neq v_i \\ v'_j \in V_2}} 2^{-d(v_i, v'_j)} = 3/2^1 + (n-2)/2^2 \\ &= (n+4)/4. \end{aligned} \tag{12}$$

Case1.2. If v_c is the central vertex of $S'(W_n)\backslash V_2 = W_n$, then $|N(v_c)| = n$. $N_{v_1}(c) = \{v'_1, v'_2, ..., v'_n\}$, where $v'_i \in V_2$, $\left(i = \overline{1, n}\right)$. So, there remains only one vertex v'_c in G' is at distance 2 from v_c. Thus, we have

$$\begin{aligned} C(v_c) &= \sum_{\substack{v'_j \neq v_c \\ v'_j \in V_2}} 2^{-d(v_c, v'_j)} = n/2^1 + 1/2^2 = n/2 + 1/4 \\ &= (2n+1)/4. \end{aligned} \tag{13}$$

By Summing (12) and (13), we have the value of C_2, that is $C_2 = n((n+4)/4) + (2n+1)/4$. Consequently, we get the following result:

$$\begin{aligned} S_1 &= n(n+5) + n(n+4)/4 + (2n+1)/4 \\ &= (2n^2 + 11n + 1)/4. \end{aligned} \tag{14}$$

Case2. For any vertex of $v'_i \in V_2$ $\left(i = \overline{1, n+1}\right)$ in the graph $S'(W_n)$. Clearly the vertices of V_2 except v'_c, $|N(v'_i)| = 3$, $\left(i = \overline{1, n}\right)$

and $|N(v'_c)| = n$. Hence, the closeness centrality of subgraph $S'(W_n)\backslash V_1$ is as like the equation (4), where the first and second sums are denoted by C'_1 and C'_2, respectively, that is $C(v'_i) = C'_1 + C'_2$. Furthermore, let S_2 be the sum of the all $C(v'_i)$, where $i = \overline{1, n+1}$.

It is clear that the value of C'_1 is C_2 in Case1, that is $C'_1 = n((n+4)/4) + (2n+1)/4$. So, we must calculate C'_2. Since each of the vertices in V_2 is at distance 2 from v'_i, we get $n/2^2$. Consequently, we get the following result:

$$\begin{aligned} S_2 &= n(n+4)/4 \ + (2n+1)/4 \ + (n+1)(n/4\ ) \\ &= (n^2+4n+2n+1+n^2+n)/4 \\ &= (2n^2+7n+1)/4. \end{aligned} \tag{15}$$

Clearly, $C(S'(W_n)) = S_1 + S_2$. By summing (14) and (15), we obtain the closeness centrality of $C(S'(W_n))$ as follows:

$$\begin{aligned} C(S'(W_n)) &= (2n^2+11n+1)/4 + (2n^2+7n+1)/4 = \\ &= (2n^2+9n+1)/2. \end{aligned}$$

□

Theorem 7. *The closeness centrality of $S'(S_n)$ is $C(S'(S_n)) = \big(4n^2+11n+2\big)/4$.*

Proof. We have two cases depending on the vertices $S'(S_n)$:

Case1. For any vertex of $v_i \in V_1$ in the graph $S'(S_n)$. The vertices of V_1 are of two kinds: v_c and $v_i\big(i = \overline{1,\ n}\big)$. The vertex v_c will be referred to as central vertex and the vertex v_i – as minor vertex. For the central vertex v_c, clearly it is exactly adjacent to $2n$ vertices except v'_c of $S'(S_n)$. Thus $|N(v_c)| = 2n$. The vertex v_c is at distance 2 to other remaining one vertex v'_c of $S'(S_n)$. Then, the closeness centrality for the vertex v_c is

$$\begin{aligned} C(v_c) &= \sum_{v_i \neq v_c} 2^{-d(v_i,v_c)} = 2n/2^1 + 1/2^2 \\ &= (4n+1)/4. \end{aligned} \tag{16}$$

For minor vertex v_i of V_1 $i = \overline{1, n}$. v_i is exactly adjacent to 2 vertices of $S'(S_n)$, named v_c and ${v'}_c$. Thus $|N(v_i)| = 2$. The other remaining $n + (n-1)$ vertices are at distance 2 from v_i. Hence, we get:

$$C(v_i) = 2/2^1 + (2n-1)/2^2 = (2n+3)/4. \tag{17}$$

Let S_1 be the sum of the all $C(v_i)$, where $i = \overline{1, n+1}$. Then, we get the following:

$$\begin{aligned} S_1 &= (4n+1)/4 + n((2n+3)/4) \\ &= (2n^2 + 7n + 1)/4. \end{aligned} \tag{18}$$

Case2. For any vertex of ${v'}_i \in V_2$ in the graph $S'(S_n)$. For the central vertex ${v'}_c$ of V_2, ${v'}_c$ is adjacent to n minor vertices of V_1 in $S'(S_n)$. Thus, $|N({v'}_c)| = n$. The other remaining n minor vertices of V_2 and central vertex v_c of V_1 are at distance 3 and 2, respectively, from ${v'}_c$. Hence, we get:

$$C({v'}_c) = n/2^1 + n/2^3 + 1/2^2 = (5n+2)/8. \tag{19}$$

For minor vertex ${v'}_i$ of V_2 $i = \overline{1, n}$. ${v'}_i$ is adjacent to one central vertex v_c of V_1 in $S'(S_n)$.Thus $|N({v'}_i)| = 1$. The other remaining $2n-1$ vertices of $S'(S_n)$ and central vertex ${v'}_c$ of V_2 are at distance 2 and 3, respectively from ${v'}_i$. Hence, we get:

$$C({v'}_i) = 1/2^1 + (2n-1)/2^2 + 1/2^3 = (4n+3)/8. \tag{20}$$

Let S_2 be the sum of the all $C({v'}_i)$, where $i = \overline{1, n+1}$. Then, we get the following:

$$\begin{aligned} S_2 &= (5n+2)/8 + n((4n+3)/8) \\ &= (4n^2 + 8n + 2)/8. \end{aligned} \tag{21}$$

Clearly, $C(S'(S_n)) = S_1 + S_2$. By summing (18) and (21), we obtain the closeness centrality of $C(S'(S_n))$ as follows:

$$\begin{aligned} C(S'(S_n)) &= (2n^2 + 7n + 1)/4 + (4n^2 + 8n + 2)/8 \\ &= (4n^2 + 11n + 2)/4. \end{aligned}$$

□

Theorem 8. *The closeness centrality of* $S'(K_n)$ *is* $C(S'(K_n)) = (7n^2 - 5n)/4$.

Proof. We have two cases depending on the vertices $S'(K_n)$:

Case1. For any vertex of $v_i \in V_1$ in the graph $S'(K_n)$. Then $|N(v_i)| = 2(n-1)$. The vertex v'_i is a copy of the vertex v_i in $S'(K_n)$. Since the vertex v_i is adjacent to every vertex except v'_i in $S'(K_n)$, the vertex v'_i is at distance 2 from v_i. Thus, the closeness centrality for the vertex v_i is

$$C(v_i) = \sum_{v_j \neq v_i} 2^{-d(v_i,v_j)} = 2(n-1)/2^1 + 1/2^2 = (4n-3)/4.$$

Let S_1 be the sum of the all $C(v_i)$, where $i = \overline{1,n}$. Then, we get the following:

$$S_1 = n((4n-3)/4) = (4n^2 - 3n)/4. \tag{22}$$

Case2. For any vertex of $v'_i \in V_2$ in the graph $S'(K_n)$. The vertex v'_i is at distance 1 to the $n-1$ vertices in $V_1 \setminus \{v_i\}$. Therefore, we have $|N(v'_i)| = n-1$. The vertices of V_2 and the vertex v_i are at distance 2 from v'_i. Thus, the closeness centrality for the vertex v'_i is

$$C(v'_i) = (n-1)/2 + (n-1+1)/2^2 = (n-1)/2 + n/4.$$

Let S_2 be the sum of the all $C(v'_i)$, where $i = \overline{1,n}$. Then, we get the following:

$$S_2 = n((n-1)/2 + n/4) = (3n^2 - 2n)/4. \tag{23}$$

Clearly, $C(S'(K_n)) = S_1 + S_2$. By summing (22) and (23), we obtain the closeness centrality of $C(S'(K_n))$ as follows:

$$\begin{aligned} C(S'(K_n)) &= (4n^2 - 3n)/4 + (3n^2 - 2n)/4 \\ &= (7n^2 - 5n)/4. \end{aligned}$$

□

4 Conclusion

In this article, we have studied the closeness centrality of various networks as a centrality measure. Closeness centrality helps us to know how long it will spread information from a given node to other reachable nodes in the network. We present comparisons between popular interconnection networks and splitting graphs of these below. These networks are complete graph K_{20}, path graph P_{20}, cycle graph C_{20}, star graph S_{20} and wheel graph W_{20}. The splitting graphs of these networks were considered to measure how far information can spread throughout the graph. The closeness centrality values of the above graphs are shown in Table 1.

Table 1. The closeness centrality values of some graphs.

Graph G	$C(G)$	Splitting graph $S'(G)$	$C(S'(G))$
K_{20}	190	$S'(K_{20})$	675
P_{20}	36	$S'(P_{20})$	139,75
C_{20}	39,94	$S'(C_{20})$	155,77
S_{20}	104,5	$S'(S_{20})$	455,5
W_{20}	125	$S'(W_{20})$	490,5

By using Table 1, we say that the graphs $S'(G)$ are better than the graph G. Calculation of *closeness centrality* for graph types considered in the article is important because if one can break a more complex network into smaller networks, then under some conditions the solutions for the optimization problem on the smaller networks can be combined to a solution for the optimization problem on the larger network. Therefore, designers for choosing the appropriate networks can use these results.

5 Acknowledgment

The authors are grateful to the area editor and the anonymous referees for their constructive comments and valuable suggestions which have

helped very much to improve the paper.

References

[1] A. Aytaç and Z. N. Odabaş, "Residual closeness of wheels and related networks," *Internat. J. Found. Comput. Sci.*, vol. 22, no. 5, pp. 1229–1240, 2011.

[2] S. P. Borgatti, "Centrality and network flow," *Social Networks*, vol. 27, no. 1, pp. 55–71, Jan. 2005.

[3] C. Dangalchev, "Residual closeness in networks," *Physica A Statistical Mechanics and its Applications*, vol. 365, pp. 556–564, Jun. 2006.

[4] ——, "Residual closeness and generalized closeness," *Internat. J. Found. Comput. Sci.*, vol. 22, no. 8, pp. 1939–1948, 2011.

[5] L. C. Freeman, *The development of social network analysis: A study in the sociology of science*. Empirical Press, Vancouver, 2004.

[6] T. Turaci and M. Ökten, "Vulnerability of Mycielski graphs via residual closeness," *Ars Combin.*, vol. 118, pp. 419–427, 2015.

[7] T. Turaci and V. Aytaç, "Residual closeness of splitting networks," *Ars Combin.*, vol. 130, pp. 17–27, 2017.

[8] S. Wasserman and F. Katherine, *Social network analysis : methods and applications*. Cambridge: Cambridge University Press, 1994.

[9] Z. N. Odabaş and A. Aytaç, "Residual closeness in cycles and related networks," *Fund. Inform.*, vol. 124, no. 3, pp. 297–307, 2013.

[10] D. Chen, L. L, M.-S. Shang, Y.-C. Zhang, and T. Zhou, "Identifying influential nodes in complex networks," *Physica A: Statistical Mechanics and its Applications*, vol. 391, no. 4, pp. 1777 – 1787, 2012.

[11] J. A. Bondy and U. S. R. Murty, *Graph theory with applications.* American Elsevier Publishing Co., Inc., New York, 1976.

[12] J. Mycielski, "Sur le coloriage des graphs," *Colloq. Math.*, vol. 3, pp. 161–162, 1955.

[13] E. Sampathkumar and H. B. Walikar, "On splitting graph of a graph," *J. Karnatak University Sci.*, vol. 25-26, pp. 13–16, 1980-81.

Vecdi Aytaç, Tufan Turacı Received March 12, 2018

Vecdi Aytaç
Ege University, Engineering Faculty, Computer Engineering Dept.
Bornova-IZMIR-TURKEY
Phone:+90 232 311 53 24
E–mail: vecdi.aytac@ege.edu.tr

Tufan Turacı
Karabük University, Science Faculty, Mathematics Dept.
Karabük-TURKEY
Phone:+90 370 433 83 74 / 1229
E–mail: tufanturaci@karabuk.edu.tr

Computer Science Journal of Moldova, vol.**26**, no.3(78), 2018

New Bounds for the Harmonic Energy and Harmonic Estrada index of Graphs

Akbar Jahanbani

Abstract

Let G be a finite simple undirected graph with n vertices and m edges. The Harmonic energy of a graph G, denoted by $\mathcal{H}E(G)$, is defined as the sum of the absolute values of all Harmonic eigenvalues of G. The Harmonic Estrada index of a graph G, denoted by $\mathcal{H}EE(G)$, is defined as $\mathcal{H}EE = \mathcal{H}EE(G) = \sum_{i=1}^{n} e^{\gamma_i}$, where $\gamma_1 \geqslant \gamma_2 \geqslant \cdots \geqslant \gamma_n$ are the $\mathcal{H}$-*eigenvalues* of G. In this paper we present some new bounds for $\mathcal{H}E(G)$ and $\mathcal{H}EE(G)$ in terms of number of vertices, number of edges and the sum-connectivity index.

Keywords: Eigenvalue of graph, Energy, sum-connectivity index, Harmonic energy, Harmonic Estrada index.

1 Introduction

Let $G = (V, E)$ be a simple undirected graph with vertex set $V = V(G) = \{v_1, v_2, ..., v_n\}$ and edge set $E(G), |E(G)| = m$. The *order* and *size* of G are $n = |V|$ and $m = |E|$, respectively. For a vertex $v_i \in V$, the degree of v_i, denoted by $\deg(v_i)$ (or just d_i), is the number of edges incident to v. The independence number, denoted $\alpha(G)$, of graph G is defined as the size of the largest independent set in G. The chromatic number $\chi'(G)$ of G is the smallest number of colors needed to color all vertices of G in such a way that no pair of adjacent vertices get the same color. A graph G is *regular* if there exists a constant r such that each vertex of G has degree r, such graphs are also called *r-regular*. The *adjacency matrix* $A(G)$ of G is defined by its entries

as $a_{ij} = 1$ if $v_iv_j \in E(G)$ and 0 otherwise. Let $\lambda_1 \geqslant \lambda_2 \geqslant \cdots \geqslant \lambda_n$ denote the *eigenvalues* of $A(G)$. λ_1 is called the *spectral radius* of the graph G. The *energy* of the graph G is defined as:

$$\mathcal{E} = \mathcal{E}(G) = \sum_{i=1}^{n} \mid \lambda_i \mid, \tag{1}$$

where λ_i, $i = 1, 2, \ldots, n$, are the *eigenvalues* of graph G. This concept was introduced by I. *Gutman* and is intensively studied in *chemistry*, since it can be used to approximate the total π-*electron* energy of a *molecule* (see, e.g. [21], [23]). Since then, numerous other bounds for *energy* were found (see, e.g. [1], [2], [22], [24], [32], [33], [34]).

For a graph G, the $Harmonic$ index $\mathcal{H}(G)$ is defined in [19] as

$$\mathcal{H}(G) = \sum_{uv \in E(G)} \frac{2}{d(u) + d(v)},$$

where $d(u)$ denotes the degree of a vertex u in G. In 2012, *Zhong* reintroduced this index as $Harmonic$ index and found the minimum and maximum values of the $Harmonic$ index for simple connected graphs and trees [39]. To know more about this index, refer to [[3] – [5], [11] – [10], [28], [36], [39] – [41]]. In [19], *Favaron et al.* considered the relation between $Harmonic$ index and the eigenvalues of graphs. *Zhong* [39], found the minimum and maximum values of the $Harmonic$ index for connected graphs and trees, and characterized the corresponding extremal graphs. Recently, Wu et al. [38], give a best possible lower bound for the $Harmonic$ index of a graph (a triangle-free graph, respectively) with order n and minimum degree at least two and characterize the extremal graphs.

The sum-connectivity index $\chi(G)$ and the general sum-connectivity index $\chi_\beta(G)$ were recently proposed by Zhou and Trinajstić in ([42], [43]) and defined as

$$\chi(G) = \sum_{uv \in E(G)} (d(u) + d(v))^{\frac{-1}{2}}$$

and

$$\chi_\beta(G) = \sum_{uv \in E(G)} (d(u) + d(v))^\beta, \tag{2}$$

where β is a real number. Some mathematical properties of the (general) sum-connectivity index on trees, *molecular* trees, *unicyclic* graphs and *bicyclic* graphs were given in ([42], [43], [15]- [17]). The *Harmonic* matrix of a graph G is a square matrix $\mathcal{H}(G) = [h_{ij}]$ of order n, defined via [27]

$$h_{ij} = \begin{cases} 0 & \text{if the vertices } v_i \text{ and } v_j \text{ of } G \text{ are not adjacent} \\ \frac{2}{(d_i+d_j)} & \text{if the vertices } v_i \text{ and } v_j \text{ of } G \text{ are adjacent} \\ 0 & \text{if } i = j. \end{cases} \tag{3}$$

The eigenvalues of the Harmonic matrix $\mathcal{H}(G)$ are denoted by $\gamma_1, \gamma_2, \ldots, \gamma_n$ and are said to be the $\mathcal{H}$-eigenvalues of G and their collection is called *Harmonic* spectrum or $\mathcal{H}$-*spectrum* of G. We note that since the Harmonic matrix is symmetric, its eigenvalues are real and can be ordered as $\gamma_1 \geqslant \gamma_2 \geqslant \cdots \geqslant \gamma_n$.

This paper is organized as follows. In Section 2, we give a list of some previously known results. In Section 3, we obtain lower and upper bounds for the *Harmonic energy* of graph G. In Section 4, we obtain lower and upper bounds for the *Harmonic Estrada* index of graph G. All graphs considered in this paper are simple.

2 Preliminaries and known results

In this section, we shall list some previously known results that will be needed in the next section. We first calculate $tr(\mathcal{H}^2)$ and $tr(\mathcal{H}^3)$, where tr denotes the trace of the respective matrix.

Denote by N_k the k-th spectral moment of the *Harmonic* matrix $\mathcal{H}$, i. e.,

$$N_k = \sum_{i=1}^{n} (\gamma_i)^k \tag{4}$$

and recall that $N_k = tr(\mathcal{H}^k)$.

Lemma 1. *Let G be a graph with n vertices and Harmonic matrix $\mathcal{H}$. Then*

$$(1) \quad N_0 = \sum_{i=1}^{n} (\gamma_i)^0 = n, \tag{5}$$

$$(2) \quad N_1 = \sum_{i=1}^{n} \gamma_i = tr(\mathcal{H}) = 0, \tag{6}$$

$$(3) \quad N_2 = \sum_{i=1}^{n} (\gamma_i)^2 = tr(\mathcal{H}^2) = 8\chi_{-2}(G), \tag{7}$$

$$(4) \quad N_3 = \sum_{i=1}^{n} (\gamma_i)^3 = tr(\mathcal{H}^3) = 32\chi_{-2}(G)\Big(\sum_{k\sim i,\, k\sim j} \frac{1}{(d_k)^2}\Big), \tag{8}$$

$$(5) \quad N_4 = \sum_{i=1}^{n} (\gamma_i)^4 = tr(\mathcal{H}^4) = \sum_{i=1}^{n} \Big(\sum_{i\sim j} \frac{4}{(d_i+d_j)^2}\Big)^2 \tag{9}$$

$$+ \sum_{i\neq j} \frac{4}{(d_i+d_j)^2}\Big(\sum_{k\sim i,\, k\sim j} \frac{4}{(d_k)^2}\Big)^2. \tag{10}$$

where $\sum_{i\sim j}$ indicates summation over all pairs of adjacent vertices v_i, v_j and also

$$\sum_{k\sim i,\, k\sim j} \frac{1}{(d_k)^2} = \sum_{k\sim i,\, k\sim j} \frac{1}{(d_i+d_k)(d_k+d_j)}.$$

Nowadays, $\mathcal{H}$ is referred to as the *Harmonic* index.

Proof. By definition, the diagonal elements of $\mathcal{H}$ are equal to zero. Therefore the trace of $\mathcal{H}$ is zero.

Next, we calculate the matrix $\mathcal{H}^2$. For $i = j$

$$(\mathcal{H}^2)_{ii} = \sum_{j=1}^{n} \mathcal{H}_{ij}\mathcal{H}_{ji} = \sum_{j=1}^{n} (\mathcal{H}_{ij})^2 = \sum_{i\sim j} (\mathcal{H}_{ij})^2 = \sum_{i\sim j} \frac{4}{(d_i+d_j)^2},$$

whereas for $i \neq j$

$$(\mathcal{H}^2)_{ij} = \sum_{j=1}^{n} \mathcal{H}_{ij}\mathcal{H}_{ji} = \mathcal{H}_{ii}\mathcal{H}_{ij} + \mathcal{H}_{ij}\mathcal{H}_{jj} + \sum_{k\sim i,\, k\sim j} \mathcal{H}_{ik}\mathcal{H}_{kj} =$$

$$= \frac{2}{(d_i + d_j)} \sum_{k\sim i,\, k\sim j} \frac{4}{(d_k)^2}.$$

Therefore

$$tr(\mathcal{H}^2) = \sum_{i=1}^{n} \sum_{i\sim j} \frac{4}{(d_i + d_j)^2} = 8 \sum_{i\sim j} \frac{1}{(d_i + d_j)^2}.$$

Hence by equality (2), we have

$$tr(\mathcal{H}^2) = 8\chi_{-2}(G).$$

Since the diagonal elements of $\mathcal{H}^3$ are

$$(\mathcal{H}^3)_{ii} = \sum_{j=1}^{n} \mathcal{H}_{ij}(\mathcal{H}^2)_{jk} = \sum_{i\sim j} \frac{2}{(d_i + d_j)} (\mathcal{H}^2)_{ij} =$$

$$= \sum_{i\sim j} \frac{4}{(d_i + d_j)^2} \Big(\sum_{k\sim i,\, k\sim j} \frac{4}{(d_k)^2} \Big)$$

we obtain

$$tr(\mathcal{H}^3) = \sum_{i=1}^{n} \sum_{i\sim j} \frac{4}{(d_i + d_j)^2} \Big(\sum_{k\sim i,\, k\sim j} \frac{4}{(d_k)^2} \Big) =$$

$$= 32 \sum_{i\sim j} \frac{1}{(d_i + d_j)^2} \Big(\sum_{k\sim i,\, k\sim j} \frac{1}{(d_k)^2} \Big).$$

Hence by equality (2), we have

$$tr(\mathcal{H}^3) = 32\chi_{-2}(G) \Big(\sum_{k\sim i,\, k\sim j} \frac{1}{(d_k)^2} \Big).$$

We now calculate $tr(\mathcal{H}^4)$. Because $tr(\mathcal{H}^4) = \|\mathcal{H}^2\|_F^2$, where $\|\mathcal{H}^2\|_F^2$ denotes the *Frobenius norm* of $\mathcal{H}^2$, we obtain

$$\begin{aligned} tr(\mathcal{H}^4) &= \sum_{i,j=1}^{n} \mid (\mathcal{H}^2)_{ii} \mid^2 = \sum_{i=j} \mid (\mathcal{H}^2)_{ii} \mid^2 + \sum_{i \neq j} \mid (\mathcal{H}^2)_{ii} \mid^2 \\ &= \sum_{i=1}^{n} \Big(\sum_{i \sim j} \frac{4}{(d_i + d_j)^2} \Big)^2 + \sum_{i \neq j} \frac{4}{(d_i + d_j)^2} \Big(\sum_{k \sim i,\, k \sim j} \frac{4}{(d_k)^2} \Big)^2. \end{aligned}$$

□

Remark 1. *Recall that [8] for a graph with eigenvalues $\lambda_1, \lambda_2, \ldots, \lambda_n$, with m edges and t triangles,*

$$M_k = \sum_{i=1}^{n} (\lambda_i)^k.$$

$$M_0 = n, \quad M_1 = \sum_{i=1}^{n} (\lambda_i) = 0, \quad M_2 = \sum_{i=1}^{n} (\lambda_i)^2 = 2m,$$

$$M_3 = \sum_{i=1}^{n} (\lambda_i)^3 = 6t.$$

Lemma 2. *(RayleighRitz) [25] If $\boldsymbol{B}$ is a real symmetric $n \times n$ matrix with eigenvalues $\lambda_1(\boldsymbol{B}) \geqslant \lambda_2(\boldsymbol{B}) \leqslant \cdots \leqslant \lambda_n(\boldsymbol{B})$, then for any $\boldsymbol{X} \in \boldsymbol{R}^n$, $(\boldsymbol{X} \neq 0)$,*

$$\boldsymbol{X}^t \boldsymbol{B} \boldsymbol{X} \leqslant \lambda_1(\boldsymbol{B}) \boldsymbol{X}^t \boldsymbol{X}.$$

Equality holds if and only if $\boldsymbol{X}$ is an eigenvector of $\boldsymbol{B}$, corresponding to the largest eigenvalue $\lambda_1(\boldsymbol{B})$.

Theorem 1. *[11] Let G be a simple graph with the chromatic number $\chi'(G)$ and the Harmonic index $\mathcal{H}(G)$, then*

$$\chi'(G) \leqslant 2\mathcal{H}(G),$$

with equality if and only if G is a complete graph, possibly with some additional isolated vertices.

Lemma 3. *[36] Let G be a triangle-free graph with n vertices and m edges, then*

$$\mathcal{H}(G) \geqslant \frac{2m}{n}.$$

Lemma 4. *[8] Let G be a graph, where the number of eigenvalues greater than, less than, and equal to zero are p, q and r, respectively. Then*

$$\alpha \leqslant r + min\{p, q\},$$

where α is the independence number of G.

Remark 2. *For non-negative $x_1, x_2, \ldots, x_n$ and $k \geqslant 2$,*

$$\sum_{i=1}^{n}(x_i)^k \leqslant (\sum_{i=1}^{n} {x_i}^2)^{\frac{k}{2}}. \tag{11}$$

Lemma 5. *[6] For any real x, one has $e^x \geqslant 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!}$. Equality holds if and only if $x = 0$.*

3 Bounds for the Harmonic Energy of a graph

In this section, we obtain lower and upper bounds for the Harmonic energy of graph. The *Harmonic* energy of the graph G is defined in [27] as:

$$\mathcal{H}E(G) = \sum_{i=1}^{n} \mid \gamma_i \mid . \tag{12}$$

First, we prove the following theorem that will be needed for obtaining the bounds of Harmonic energy.

Theorem 2. *Let G be a connected graph with $n \geqslant 2$ vertices. Then the spectral radius of the Harmonic matrix is bounded from below as*

$$\lambda_1 \geqslant \frac{2\mathcal{H}(G)}{n}. \tag{13}$$

Proof. Let $\mathcal{H} = ||h_{ij}||$ be the Harmonic matrix corresponding to $\mathcal{H}$. By Lemma 2, for any vector $X = (x_1, x_2, \ldots, x_n)^t$,

$$\begin{aligned} X^t\mathcal{H}X &= \Big(\sum_{j,j\sim 1}^{n} x_j z_{j1}, \sum_{j,j\sim 2}^{n} x_j z_{j2}, \ldots, \sum_{j,j\sim n}^{n} x_j z_{jn}\Big)^t X \\ &= 2\sum_{i\sim j} z_{ij}x_i x_j \end{aligned} \tag{14}$$

because $h_{ij} = h_{ji}$. Also,

$$X^t X = \sum_{i=1}^{n} x_i^2. \tag{15}$$

Using Eqs. (14) and (15), by Lemma 2, we obtain

$$\gamma_1 \geqslant \frac{2\sum_{i\sim j} z_{ij}x_i x_j}{\sum_{i=1}^{n} x_i^2}. \tag{16}$$

Since (16) is true for any vector X, by putting $X = (1, 1, \ldots, 1)^t$, we have

$$\gamma_1 \geqslant \frac{2\mathcal{H}(G)}{n}.$$

□

Theorem 3. *Let G be a graph with n vertices. Then*

$$\mathcal{H}E(G) \leqslant \frac{8}{n}\sqrt{\chi_{-2}(G)} + \sqrt{(n-1)\Big(8\chi_{-2}(G) - \big(\frac{8}{n}\sqrt{8\chi_{-2}(G)}\big)^2\Big)}.$$

Proof. By applying the Cauchy-Schwartz inequality to the two $(n-1)$ vectors $(1, 1, \ldots, 1)$ and $(\mid \gamma_1 \mid, \mid \gamma_2 \mid, \ldots, \mid \gamma_n \mid)$, we have

$$\Big(\sum_{i=2}^{n} \mid \gamma_i \mid \Big)^2 \leqslant (n-1)\Big(\sum_{i=2}^{n} \gamma_i^2\Big).$$

By the define of Harmonic energy, we can get

$$\begin{aligned}(\mathcal{H}E(G)-\gamma_1)^2 &= \Big(\sum_{i=2}^{n} \mid \gamma_i \mid \Big)^2 \\ &\leqslant (n-1)\Big(\sum_{i=1}^{n}\gamma_i^2-\gamma_1^2\Big) \\ &= (n-1)\Big(8\chi_{-2}(G)-\gamma_1^2\Big), \qquad \text{(by Equality 7)}\end{aligned}$$

then

$$\mathcal{H}E(G) \leqslant \gamma_1+\sqrt{(n-1)\Big(8\chi_{-2}(G)-\gamma_1^2\Big)}. \tag{17}$$

Now let us define a function

$$f(x)=x+\sqrt{(n-1)\Big(8\chi_{-2}(G)-x^2\Big)}.$$

In fact, by keeping in mind $\gamma_1 \geqslant 1$, we set $\gamma_1 = x$. Using

$$\sum_{i=2}^{n}\gamma_i^2 = 8\chi_{-2}(G),$$

we get that

$$x^2=\gamma_1^2 \leqslant 8\chi_{-2}(G).$$

In other words, $x \leqslant \sqrt{8\chi_{-2}(G)}$, meanwhile $f'(x)=0$ implies that

$$x=\sqrt{\frac{8}{n}\chi_{-2}(G)}.$$

Therefore f is a decreasing function in the interval

$$\sqrt{\frac{8}{n}\chi_{-2}(G)} \leqslant x \leqslant 8\sqrt{\chi_{-2}(G)}$$

and

$$\sqrt{\frac{8}{n}\chi_{-2}(G)} \leqslant x \leqslant \frac{8}{n}\sqrt{\chi_{-2}(G)} \leqslant \gamma_1.$$

Hence

$$f(\gamma_1) \leqslant f\left(\frac{8}{n}\sqrt{\chi_{-2}(G)}\right).$$

Therefore

$$\mathcal{H}E(G) \leqslant \frac{8}{n}\sqrt{\chi_{-2}(G)} + \sqrt{(n-1)\left(8\chi_{-2}(G) - \left(\frac{8}{n}\sqrt{8\chi_{-2}(G)}\right)^2\right)}.$$

□

By Theorem 1 and Theorem 2, we establish the following result.

Theorem 4. *Let G be a non-empty and non-singular graphs with n vertices and chromatic number χ'. Then*

$$\mathcal{H}E(G) \geqslant \frac{\chi'}{n} + ln \mid det\mathcal{H} \mid -ln\left(\frac{\chi'}{n}\right). \tag{18}$$

Proof. Since G is non-singular, it is $\mid \gamma_i \mid > 0, i = 1, 2, \ldots, n$. Consider a function

$$f_1(x) = x - 1 - lnx,$$

for $x > 0$. It is elementary to prove that $f_1(x)$ is increasing for $x \geqslant 1$ and decreasing for $0 < x \leqslant 1$. Consequently, $f_1(x) \geqslant f_1(1) = 0$, implying that $x \geqslant 1 + lnx$ for $x > 0$, with equality holding if and only if $x = 1$. Using the above result, we get

$$\begin{aligned}
\mathcal{H}E(G) &= \gamma_1 + \sum_{i=2}^{n} \mid \gamma_i \mid \\
&\geq \gamma_1 + n - 1 + \sum_{i=2}^{n} ln \mid \gamma_i \mid \\
&= \gamma_1 + n - 1 + ln\prod_{i=2}^{n} \mid \gamma_i \mid \\
&= \gamma_1 + n - 1 + ln \mid det\mathcal{H} \mid -ln\gamma_1. \qquad (19)
\end{aligned}$$

At this point, one has to recall that, by Lemma 2, $\gamma_1 \geqslant \frac{\chi'}{n}$.
Since $x \geqslant \frac{\chi'}{n} \geqslant 1$, we have that

$$g(x) = x + n - 1 + ln \mid det\mathcal{H} \mid -lnx,$$

is an increasing function on $1 \leqslant x \leqslant n$. So we conclude that

$$g(x) \geqslant g\left(\frac{\chi'}{n}\right) = \frac{\chi'}{n} + (n-1) + ln \mid det\mathcal{H} \mid -ln\left(\frac{\chi'}{n}\right).$$

Combining the above result with (19), we arrive at (18). □

Also, by Theorem 2 and Lemma 3, we establish the following result.

Remark 3. *Let G be a triangle-free graph with n vertices and m edges, then*

$$\mathcal{H}E(G) \geqslant \frac{4m}{n^2} + ln \mid det\mathcal{H} \mid -ln\left(\frac{4m}{n^2}\right).$$

Or

$$\mathcal{H}E(G) \leqslant \frac{4m}{n^2} + \sqrt{(n-1)(8\chi_{-2}(G) - \frac{4m}{n^2})}.$$

Theorem 5. *Let G be a connected graph with $n \geqslant 2$ vertices and independence number α. Then*

$$\mathcal{H}E(G) \leqslant 2\sqrt{(n-\alpha)\chi_{-2}(G)}.$$

Proof. Let $\gamma_1, \gamma_2, \ldots, \gamma_p$, be the p positive eigenvalues of G and let $\eta_1, \eta_2, \ldots, \eta_q$, be the q negative eigenvalues of G. Then G has $n-p-q$ eigenvalues which are equal to zero. From Lemma 4, we have

$$\alpha \leqslant (n-p-q) + min\{p, q\}.$$

Thus $\alpha \leqslant (n-p-q)+p$ and $\alpha \leqslant (n-p-q)+q$. Namely, $p \leqslant n-\alpha$ and $q \leqslant n-\alpha$. Since $\sum_{i=1}^{p} \gamma_i + \sum_{i=1}^{q} \eta_i = 0$, we have that

$$\mathcal{H}E(G) = 2\sum_{i=1}^{p} \gamma_i = 2\sum_{i=1}^{q} \mid \eta_i \mid .$$

From Cauchy - Schwarz inequality, we have that

$$\mathcal{H}E(G) = 2\sum_{i=1}^{p}\gamma_i \leqslant 2\sqrt{p\sum_{i=1}^{p}\gamma_i}.$$

Similarly, we have that

$$\mathcal{H}E(G) = 2\sum_{i=1}^{q}\eta_i \leqslant 2\sqrt{q\sum_{i=1}^{q}\eta_i}.$$

Therefore

$$\begin{aligned}\frac{\mathcal{H}E(G)^2}{2} = \frac{\mathcal{H}E(G)^2}{4} + \frac{\mathcal{H}E(G)^2}{4} &\leqslant p\sum_{i=1}^{p}\gamma_i^2 + q\sum_{i=1}^{q}\eta_i^2 \\ &\leqslant (n-\alpha)\sum_{i=1}^{p}\gamma_i^2 + (n-\alpha)\sum_{i=1}^{q}\eta_i^2 \\ &= (n-\alpha)\Big(\sum_{i=1}^{p}\gamma_i^2 + \sum_{i=1}^{q}\eta_i^2\Big) \\ &= 8(n-\alpha)\chi_{-2}(G).\end{aligned}$$

Hence

$$\mathcal{H}E(G) \leqslant 4\sqrt{(n-\alpha)\chi_{-2}(G)}.$$

□

Theorem 6. *If the graph G is regular of degree $r, r > 0$, then*

$$\mathcal{H}E(G) = \frac{1}{r}\mathcal{E}(G).$$

If, in addition $r = 0$, then $\mathcal{H}E(G) = 0$.

Proof. If $r = 0$, then G is the graph without edges. Then directly from the definition (3) it follows that $\mathcal{H}_{i,j} = 0$ for all $i, j = 1, 2, \ldots, n$, i. e., that $\mathcal{H}(G) = 0$. All eigenvalues of the zero matrix 0 are equal to zero. Therefore, $\mathcal{H}E(G) = 0$.

Suppose now that G is regular of degree $r > 0$, i. e., that $d_1 = d_2 = \cdots = d_n = r$. Then all non-zero terms in $\mathcal{H}(G)$ are equal to $\frac{1}{r}$, implying that $\mathcal{H}(G) = \frac{1}{r}A(G)$. Therefore, $\gamma_i = \frac{1}{r}\lambda_i$. Theorem 6 follows from the definitions of energy and Harmonic energy. □

Theorem 7. *Let G be a graph with n vertices. Then*

$$\mathcal{H}E(G) \leqslant \sqrt{8n\chi_{-2}(G) - \frac{n}{2}(\mid \gamma_1 \mid - \mid \gamma_n \mid)^2}. \tag{20}$$

Proof. From the Lagrange's identity (see for example [22]),

$$0 \leqslant 8n\chi_{-2}(G) - \mathcal{H}E(G)^2 = \sum_{i=1}^{n} \mid \gamma_i \mid^2 - \left(\sum_{i=1}^{n} \mid \gamma_i \mid \right)^2 =$$
$$= \sum_{1 \leqslant i \leqslant j \leqslant n} (\mid \gamma_i \mid - \mid \gamma_j \mid)^2,$$

the following inequality can be obtained

$$0 \leqslant 8n\chi_{-2}(G) - \mathcal{H}E(G)^2 \geqslant \sum_{i=2}^{n-1} \Big((\mid \gamma_1 \mid - \mid \gamma_i \mid)^2 + (\mid \gamma_i \mid - \mid \gamma_n \mid)^2)$$
$$+ (\mid \gamma_1 \mid - \mid \gamma_n \mid)^2 \Big).$$

On the other hand, according to the Jennsen's inequality (see [21]), from the above inequality it follows that

$$0 \leqslant 8n\chi_{-2}(G) - \mathcal{H}E(G)^2 \geqslant \frac{n-2}{2}(\mid \gamma_1 \mid - \mid \gamma_n \mid)^2 + (\mid \gamma_1 \mid - \mid \gamma_n \mid)^2$$
$$= \frac{n}{2}(\mid \gamma_1 \mid - \mid \gamma_n \mid)^2.$$

After rearranging the above inequality, the inequality (20) is obtained. □

Theorem 8. *Let G be a graph with $n \geqslant 2$ vertices. Then for each T with the property $\gamma_1 \geqslant T \geqslant \sqrt{\frac{8\chi_{-2}(G)}{n}}$, the following is valid*

$$\mathcal{H}E(G) \leqslant T + \sqrt{(n-1)(8\chi_{-2}(G) - T^2)}. \tag{21}$$

Proof. In [37] a class of real polynomials $P_n(x) = x^n + a_1x^{n-1} + a_2x^{n-2} + b_3x^{n-3} + \cdots + b_n$, denoted as $P_n(a_1, a_2)$, where a_1 and a_2 are fixed real numbers, was considered. For the roots $x_1 \geqslant x_2 \geqslant \cdots \geqslant x_n$ of an arbitrary polynomial $P_n(x)$ from this class, the following values were introduced

$$\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i, \tag{22}$$

$$\Delta = n\sum_{i=1}^{n} x_i^2 - \left(\sum_{i=1}^{n} x_i\right)^2. \tag{23}$$

Then upper and lower bounds for the polynomial roots, $x_i, i = 1, 2, \ldots, n$, were determined in terms of the introduced values

$$\bar{x} + \frac{1}{n}\sqrt{\frac{\Delta}{n-1}} \leqslant x_1 \leqslant \bar{x} + \frac{1}{n}\sqrt{(n-1)\Delta}, \tag{24}$$

$$\bar{x} - \frac{1}{n}\sqrt{\frac{i-1}{n-i+1}\Delta} \leqslant x_i \leqslant \bar{x} + \frac{1}{n}\sqrt{\frac{n-i}{i}\Delta}, \quad 2 \leqslant i \leqslant n-1, \tag{25}$$

$$\bar{x} - \frac{1}{n}\sqrt{(n-1)\Delta} \leqslant x_n \leqslant \bar{x} - \frac{1}{n}\sqrt{\frac{\Delta}{n-1}}. \tag{26}$$

Consider the polynomial

$$\psi_n(x) = \prod_{i=1}^{n}(x - \mid \gamma_i \mid) = x^n + a_1x^{n-1} + a_2x^{n-2} + b_3x^{n-3} + \cdots + b_n.$$

Since

$$a_1 = -\sum_{i=1}^{n} \mid \gamma_i \mid = -\mathcal{H}E$$

and

$$a_2 = \frac{1}{2}\left[\left(\sum_{i=1}^{n} \mid \gamma_i \mid\right)^2 - \sum_{i=1}^{n} \mid \gamma_i \mid^2\right] = \frac{1}{2}\mathcal{H}E^2 - 4\chi_{-2}(G),$$

the polynomial $\psi_n(x)$ belongs to a class of real polynomials $P_n(-\mathcal{H}E, \frac{1}{2}\mathcal{H}E^2 - 4\chi_{-2}(G))$. Based on the following equalities

$$\bar{x} = \frac{1}{n}\sum_{i=1}^{n} \mid \gamma_i \mid = \frac{\mathcal{H}E}{n}, \tag{27}$$

$$\Delta = n\sum_{i=1}^{n} \mid \gamma_i \mid^2 - \Big(\sum_{i=1}^{n} \mid \gamma_i \mid \Big)^2 = 8n\chi_{-2}(G) - \mathcal{H}E^2, \tag{28}$$

for $x_1 = \gamma_1$, according to (27), (28) and the right-hand side of the first inequality in (25), we get

$$\gamma_1 \leqslant \frac{\mathcal{H}E}{n} + \sqrt{(n-1)(8n\sum_{i\sim j}\frac{1}{(d_i+d_j)^2} - \mathcal{H}E^2)}. \tag{29}$$

Now, for each real T with the property $\gamma_1 \geqslant T \geqslant \sqrt{\frac{\chi_{-2}(G)}{n}}$ from (29) it follows that

$$T \leqslant \frac{\mathcal{H}E}{n} + \sqrt{(n-1)(8n\chi_{-2}(G) - \mathcal{H}E^2)}.$$

After rearranging the above inequality, the inequality (21) is obtained. □

Theorem 9. *Let G be a simple graph with $n \geqslant 2$ vertices. Then*

$$\frac{1}{n}\sqrt{\frac{8n\chi_{-2}(G)}{n-1}} \leqslant \gamma_1 \leqslant \frac{1}{n}\sqrt{8n(n-1)\chi_{-2}(G)},$$

$$-\frac{1}{n}\sqrt{\frac{i-1}{n-i+1}8n\chi_{-2}(G)} \leqslant \gamma_i \leqslant \frac{1}{n}\sqrt{\frac{n-i}{i}8n\chi_{-2}(G)}$$

$$\textit{for} \quad 2 \leqslant i \leqslant n-1$$

$$-\frac{1}{n}\sqrt{8n(n-1)\chi_{-2}(G)} \leqslant \gamma_n \leqslant -\frac{1}{n}\sqrt{\frac{8n\chi_{-2}(G)}{n-1}}.$$

Proof. Let the characteristic polynomial of a graph G be the following:

$$\varphi_n(x) = \prod_{i=1}^{n}(x-\gamma_i) = x^n + a_1x^{n-1} + a_2x^{n-2} + b_3x^{n-3} + \cdots + b_n.$$

Since

$$a_1 = -\sum_{i=1}^{n} \gamma_i = 0$$

and

$$a_2 = \frac{1}{2}\left[\left(\sum_{i=1}^{n} \gamma_i\right)^2 - \sum_{i=1}^{n} \gamma_i^2\right] = -4\chi_{-2}(G),$$

the polynomial $\varphi_n(x)$ belongs to a class of real polynomials $P_n(0, -4\chi_{-2}(G))$, by the equalities

$$\bar{x} = \frac{1}{n}\sum_{i=1}^{n} \gamma_i = 0$$

and

$$\Delta = n\sum_{i=1}^{n} \gamma_i^2 - \left(\sum_{i=1}^{n} \gamma_i\right)^2 = 8n\chi_{-2}(G)$$

and inequalities (24), (25), (26), the proof is completed. □

Theorem 10. *Let G be a graph with n vertices and $|\gamma_1| \geqslant |\gamma_2| \geqslant \cdots \geqslant |\gamma_n|$ be a non-increasing arrangement of eigenvalues of G. Then, the following inequality is valid*

$$\mathcal{H}E(G) \geqslant \sqrt{8n\chi_{-2}(G) - \theta(n)(|\gamma_1| - |\gamma_n|)^2}. \tag{30}$$

where $\theta(n) = n[\frac{n}{2}](1 - \frac{1}{n}[\frac{n}{2}])$, while $[x]$ denotes integer part of a real number x.

Proof. Let $a_1, a_2, \ldots, a_n$ and $b_1, b_2, \ldots, b_n$ be real numbers for which there exist real constants a, b, A and B, so that for each $i, i = 1, 2, \ldots, n$, $a \leqslant a_i \leqslant A$ and $b \leqslant b_i \leqslant B$. Then the following inequality is valid (see [7])

$$\left| n\sum_{i=1}^{n} a_i b_i - \sum_{i=1}^{n} a_i \sum_{i=1}^{n} b_i \right| \leqslant \theta(n)(A-a)(B-b). \tag{31}$$

Equality in (31) holds if and only if $a_1 = a_2 = \cdots = a_n$ and $b_1 = b_2 = \cdots = b_n$.

For $a_i := |\gamma_i|$, $b_i := |\gamma_i|$, $a = b := |\gamma_n|$ and $A = B := |\gamma_1|$, $i = 1, 2, \dots n$ inequality (31) becomes

$$\left| n \sum_{i=1}^{n} |\gamma_i|^2 - \left(\sum_{i=1}^{n} |\gamma_i| \right)^2 \right| \leqslant \theta(n)(|\gamma_1| - |\gamma_n|)^2.$$

Therefore, the above inequality becomes

$$8n\chi_{-2}(G) - \mathcal{HE}(G)^2 \leqslant \theta(n)(|\gamma_1| - |\gamma_n|)^2,$$

wherefrom the statement of Theorem 10 follows. Since equality in (31) holds if and only if $a_1 = a_2 = \cdots = a_n$ and $b_1 = b_2 = \cdots = b_n$, equality in (30) holds if and only if $|\gamma_1| = |\gamma_2| = \cdots = |\gamma_n|$. □

Theorem 11. *Let G be a graph with n vertices and $|\gamma_1| \geqslant |\gamma_2| \geqslant \cdots \geqslant |\gamma_n|$ be a non-increasing arrangement of eigenvalues of G. Then, the following inequality is valid*

$$\mathcal{HE}(G) \geqslant \frac{|\gamma_1||\gamma_n|n + 8\chi_{-2}(G)}{|\gamma_1| + |\gamma_n|}. \tag{32}$$

Equality in (32) holds if and only if $G \cong \bar{K}_n$.

Proof. Let $a_1, a_2, \dots, a_n$ and $b_1, b_2, \dots, b_n$ be real numbers for which there exist real constants R and r, so that for each $i, i = 1, 2, \dots, n$ there holds $ra_i \leqslant b_i \leqslant Ra_i$. Then the following inequality is valid (see [14])

$$\sum_{i=1}^{n} b_i^2 + rR \sum_{i=1}^{n} a_i^2 \leqslant (r + R) \sum_{i=1}^{n} a_i b_i. \tag{33}$$

Equality in (33) holds if and only if for at least one $i, 1 \leqslant i \leqslant n$ there holds $ra_i = bi = Ra_i$.

For $a_i := 1$, $b_i := |\gamma_i|$, $r := |\gamma_n|$ and $R := |\gamma_1|$, $i = 1, 2, \dots n$, inequality (31) becomes

$$\sum_{i=1}^{n} |\gamma_i|^2 + |\gamma_1||\gamma_n| \sum_{i=1}^{n} 1 \leqslant (|\gamma_n| + |\gamma_1|) \sum_{i=1}^{n} |\gamma_i|.$$

Therefore, the above inequality becomes

$$8n\chi_{-2}(G) + n|\gamma_1||\gamma_n| \leqslant (|\gamma_n| + |\gamma_1|)\mathcal{H}E(G).$$

If for some i there holds that $ra_i = b_i = Ra_i$, then for the same i the following equality also holds: $b_i = r = R$. This means that for each $j, j \neq i$ there holds $|\gamma_i| \leqslant |\gamma_j| \leqslant |\gamma_i|$. Therefore equality in (33) holds if and only if $|\gamma_1| = |\gamma_2| = \cdots = |\gamma_n|$. □

Theorem 12. *Let G be a non-empty graph with n vertices. Then*

$$\mathcal{H}E(G) \geqslant \frac{(N_2)^2}{N_4}.$$

Proof. We start with the Hölder inequality

$$\sum_{i=1}^{n} a_i b_i \leqslant \left(\sum_{i=1}^{n} a_i^p\right)^{\frac{1}{p}} \left(\sum_{i=1}^{n} b_i^q\right)^{\frac{1}{q}}, \tag{34}$$

which holds for non-negative real numbers $a_1, a_2, \ldots, a_n$ and $b_1, b_2, \ldots, b_n$. Setting $a_i = |\gamma_i|^{\frac{1}{2}}$, $b_i = |\gamma_i|^{\frac{3}{2}}$, $p = 2$ and $q = 2$, from (34), we obtain

$$\sum_{i=1}^{n} |\gamma_i|^2 = \sum_{i=1}^{n} |\gamma_i|^{\frac{1}{2}} \left(|\gamma_i|^3\right)^{\frac{1}{2}} \leqslant \left(\sum_{i=1}^{n} |\gamma_i|\right)^{\frac{1}{2}} \left(\sum_{i=1}^{n} |\gamma_i|^3\right)^{\frac{1}{2}}. \tag{35}$$

Then $\sum_{i=1}^{n} |\gamma_i|^3 \neq 0$ and (35) can be written as the following

$$\sum_{i=1}^{n} |\gamma_i| \geqslant \frac{\left(\sum_{i=1}^{n} |\gamma_i^2|\right)^2}{\sum_{i=1}^{n} |\gamma_i|^3}.$$

Hence by equalities (12), (7) and (8), we have

$$\mathcal{H}E(G) \geqslant \frac{(N_2)^2}{N_4}.$$

□

Theorem 13. *Let G be a non-empty graph with n vertices. Then*

$$\mathcal{H}E(G) \geqslant \frac{\sqrt{32n\chi_{-2}(G)\big(|\gamma_1|\gamma_n|\big)}}{|\gamma_1|+|\gamma_n|}.$$

Proof. Let $a_1, a_2, \ldots, a_n$ and $b_1, b_2, \ldots, b_n$ be real numbers for which there exist real constants m_1, m_2, M_1 and M_2, so that for each $i, i = 1, 2, \ldots, n$, $m_1 \leqslant a_i \leqslant M_1$ and $m_2 \leqslant b_i \leqslant M_2$. Then the following inequality is valid by the Hölder inequality (see [26], p. 135)

$$\left[\sum_{i=1}^{n}(a_i)^2\right]\left[\sum_{i=1}^{n}(b_i)^2\right] \leqslant \frac{1}{4}\left(\sqrt{\frac{M_1M_2}{m_1m_2}}+\sqrt{\frac{m_1m_2}{M_1M_2}}\right)^2\left(\sum_{i=1}^{n}a_ib_i\right)^2, \quad (36)$$

where the equality holds if and only if $a_1 = a_2 = \cdots = a_n$, $b_1 = b_2 = \ldots = b_n$, $m_1 = M_1 = a_1$, $m_2 = M_2 = b_1$.

For $a_i := |\gamma_i|$, $b_i := 1$, $m_1 := |\gamma_n|$, $M_1 := |\gamma_1|$, $M_2 = m_2 := 1$, $i = 1, 2, \ldots n$, inequality (36) becomes

$$\left[\sum_{i=1}^{n}(|\gamma_i|)^2\right]\left[\sum_{i=1}^{n}(1)^2\right] \leqslant \frac{1}{4}\left(\sqrt{\frac{|\gamma_1|}{|\gamma_n|}}+\sqrt{\frac{|\gamma_n|}{|\gamma_1|}}\right)^2\left(\sum_{i=1}^{n}|\gamma_i|\right)^2. \quad (37)$$

Hence by equalities (12), (7), we have

$$8n\chi_{-2}(G) \leqslant \frac{1}{4}\left(\sqrt{\frac{|\gamma_1|}{|\gamma_n|}}+\sqrt{\frac{|\gamma_n|}{|\gamma_1|}}\right)^2\Big(\mathcal{H}E(G)\Big)^2.$$

Therefore

$$\mathcal{H}E(G) \geqslant \frac{\sqrt{32n\chi_{-2}(G)(|\gamma_1|\gamma_n|)}}{|\gamma_1|+|\gamma_n|}.$$

□

Theorem 14. *Let G be a graph with n vertices. Then*

$$\mathcal{H}E(G) \leqslant \sqrt[3]{n^2}\sqrt{8\chi_{-2}(G)}.$$

Proof. Let $a_1, a_2, \ldots, a_n,\ b_1, b_2, \ldots, b_n$ and $c_1, c_2, \ldots, c_n,$ be positive real numbers, $i = 1, 2, \ldots, n$. Then the following inequality is valid by the Hölder inequality (see [26], p. 137)

$$\left(\sum_{i=1}^{n} a_i b_i c_i\right)^3 \leqslant \left[\sum_{i=1}^{n}(a_i)^3\right]\left[\sum_{i=1}^{n}(b_i)^3\right]\left[\sum_{i=1}^{n}(c_i)^3\right], \qquad (38)$$

where equality holds if and only if $a_i = b_i = c_i$, $i = 1, 2, \ldots, n$. For $a_i := |\gamma_i|$, $b_i := 1$, $c_i := 1$, $i = 1, 2, \ldots n$ inequality (38) becomes

$$\begin{aligned}\left(\sum_{i=1}^{n}|\gamma_i|\right)^3 &\leqslant \left[\sum_{i=1}^{n}(|\gamma_i|)^3\right]\left[\sum_{i=1}^{n}(1)^3\right]\left[\sum_{i=1}^{n}(1)^3\right]\\ &= n^2\left[\sum_{i=1}^{n}(|\gamma_i|)^3\right]\\ &\leqslant n^2\left[\sum_{i=1}^{n}(|\gamma_i|)^2\right]^{\frac{3}{2}}, \quad \text{by Inequality (11)}\\ &= n^2\left[\sum_{i=1}^{n}(\gamma_i)^2\right]^{\frac{3}{2}}\\ &= n^2\left[8\chi_{-2}(G)\right]^{\frac{3}{2}}, \quad \text{by Equality (7).}\end{aligned}$$

Therefore

$$\mathcal{HE}(G) \leqslant \sqrt[3]{n^2}\sqrt{8\chi_{-2}(G)}.$$

□

Theorem 15. *Let G be a graph with n vertices. Then*

$$\mathcal{HE}(G) \leqslant 8\chi_{-2}(G).$$

Proof. Let $a_1, a_2, \ldots, a_n$ and $b_1, b_2, \ldots, b_n$ be real numbers for which there exist real constants r and s, such that $r + s = 1$, $r,\ s \neq 0, 1$. Then the following inequality is valid by the Hölder inequality (see [26], p. 135)

$$\sum_{i=1}^{n} a_i b_i \geqslant \left[\sum_{i=1}^{n}(a_i)^{\frac{1}{r}}\right]^r \left[\sum_{i=1}^{n}(b_i)^{\frac{1}{s}}\right]^s \quad \text{for} \quad r > 1. \qquad (39)$$

For $a_i := |\gamma_i|^{\frac{1}{2}}$, $b_i := |\gamma_i|^{\frac{1}{2}}$, $r := \frac{1}{2}$, $s := \frac{1}{2}$ inequality (39) becomes

$$\sum_{i=1}^{n} |\gamma_i|^{\frac{1}{2}} |\gamma_i|^{\frac{1}{2}} \geqslant \left[\sum_{i=1}^{n} (|\gamma_i|^{\frac{1}{2}})^2\right]^{\frac{1}{2}} \left[\sum_{i=1}^{n} (|\gamma_i|^{\frac{1}{2}})^2\right]^{\frac{1}{2}}$$

$$\sum_{i=1}^{n} |\gamma_i| \geqslant \left[\sum_{i=1}^{n} |\gamma_i|\right]^{\frac{1}{2}} \left[\sum_{i=1}^{n} |\gamma_i|\right]^{\frac{1}{2}}$$

$$\sum_{i=1}^{n} |\gamma_i|^2 \geqslant \left[\sum_{i=1}^{n} |\gamma_i|\right]^{\frac{1}{2}} \left[\sum_{i=1}^{n} |\gamma_i|\right]^{\frac{1}{2}}.$$

Hence by equalities (12) and (7), we have

$$8\chi_{-2}(G) \geqslant \mathcal{H}(G).$$

□

4 Bounds on the Harmonic Estrada index of a graph

In this section, we obtain lower and upper bounds for the Harmonic Estrada index of graphs. We first recall that the Estrada index of a graph G is defined by

$$EE = EE(G) = \sum_{i=1}^{n} e^{\lambda_i}.$$

Denoting by $M_k = M_k(G)$ to the *k-th moment* of the graph G, we get

$$M_k = M_k(G) = \sum_{i=1}^{n} (\lambda_i)^k.$$

and recalling the power-series *expansion* of e^x, we have

$$EE = \sum_{i=1}^{\infty} \frac{M_k(G)}{k!}.$$

It is well known that [18] $M_k(G)$ is equal to the number of *closed walks* of length k of the graph G. In fact *Estrada index* of graphs has an important role in *Chemistry* and *Physics* and there exists a vast *literature* that studies this special index. In addition to the Estrada's papers mentioned above, we may also refer the reader to ([12], [13], [20], [29], [30], [31]) for the detailed information, such as lower and upper bounds for *Estrada index* in terms of the number of vertices and edges, and some inequalities between *Estrada index* and the energy of G.

Let thus G be a graph of order n whose Harmonic eigenvalues are $\gamma_1 \geqslant \gamma_2 \geqslant \cdots \geqslant \gamma_n$. Then the Harmonic Estrada index of G, denoted by $\mathcal{H}EE(G)$, is defined as [35]

$$\mathcal{H}EE = \mathcal{H}EE(G) = \sum_{i=1}^{n} e^{\gamma_i}.$$

Recalling Eq. (4), it follows that

$$\mathcal{H}EE(G) = \sum_{i=1}^{\infty} \frac{N_k}{k!}.$$

Theorem 16. *Let G be a graph with n vertices. Then the Harmonic Estrada index of G is bounded as*

$$\sqrt{n^2 + 16\chi_{-2}(G)} \leqslant \mathcal{H}EE(G) \leqslant n - 1 + e^{\sqrt{8\chi_{-2}(G)}}. \qquad (40)$$

Proof. **Lower bound**. Directly from the definition of the Harmonic Estrada index, we get

$$\mathcal{H}EE(G)^2 = \sum_{i=1}^{n} e^{2\gamma_i} + 2\sum_{i<j} e^{\gamma_i} e^{\gamma_j}. \qquad (41)$$

In view of the inequality between the arithmetic and geometric means,

$$\begin{aligned} 2\sum_{i<j} e^{\gamma_i}e^{\gamma_j} &\geqslant n(n-1)\Big(\prod_{i<j} e^{\gamma_i}e^{\gamma_j}\Big)^{\frac{2}{n(n-1)}} = \\ &= n(n-1)\left[\Big(\prod_{i=1}^{n} e^{\gamma_i}\Big)^{n-1}\right]^{\frac{2}{n(n-1)}} = \\ &= n(n-1)\Big(e^{\sum_{i=1}^{n}\gamma_i}\Big)^{\frac{2}{n}}, \quad \text{by} \sum_{i=1}^{n}\gamma_i = 0 \\ &= n(n-1). \end{aligned} \tag{42}$$

By means of a power-series expansion, and bearing in mind the properties of N_0, N_1 and N_2, we get

$$\sum_{i=1}^{n} e^{2\gamma_i} = \sum_{i=1}^{n}\sum_{k\geqslant 0}\frac{(2\gamma_i)^k}{k!} = n + 16\chi_{-2}(G) + \sum_{i=1}^{n}\sum_{k\geqslant 3}\frac{(2\gamma_i)^k}{k!}.$$

Because we are aiming at a (as good as possible) lower bound, it may look plausible to replace $\sum_{k\geqslant 3}\frac{(2\gamma_i)^k}{k!}$ by $8\sum_{k\geqslant 3}\frac{(\gamma_i)^k}{k!}$. However, instead of $8 = 2^3$ we shall use a multiplier $\omega \in [0, 8]$, so as to arrive at

$$\begin{aligned} \sum_{i=1}^{n} e^{2\gamma_i} &\geqslant n + 16\chi_{-2}(G) + \omega\sum_{i=1}^{n}\sum_{k\geqslant 3}\frac{(\gamma_i)^k}{k!} \\ &= n + 16\chi_{-2}(G) - \omega n - 4\omega\chi_{-2}(G) + \omega\sum_{i=1}^{n}\sum_{k\geqslant 0}\frac{(\gamma_i)^k}{k!}, \end{aligned}$$

i.e.,

$$\sum_{i=1}^{n} e^{2\gamma_i} \geqslant (1-\omega)n + 4(4-\omega)\chi_{-2}(G) + \omega\mathcal{HEE}(G). \tag{43}$$

By substituting (42) and (43) back into (41), and solving for $\mathcal{HEE}$ we obtain

$$\mathcal{HEE} \geqslant \frac{\omega}{2} + \sqrt{(n-\frac{\omega}{2})^2 + 4(4-\omega)\chi_{-2}(G)}. \tag{44}$$

It is elementary to show that for $n \geqslant 2$ and $4\chi_{-2}(G) \geqslant 1$ the function

$$f(x) := \frac{x}{2} + \sqrt{(n - \frac{x}{2})^2 + (4 - x)4\chi_{-2}(G)}$$

monotonically decreases in the interval $[0, 8]$. Consequently, the best lower bound for $\mathcal{HEE}$ is attained not for $\omega = 8$, but for $\omega = 0$. Setting $\omega = 0$ into (44) we arrive at the first half of Theorem 16.
Upper bound. By definition of the Harmonic Estrada index, we have

$$\begin{aligned}\mathcal{HEE} &= n + \sum_{i=1}^{n}\sum_{k\geqslant 1}\frac{(\gamma_i)^k}{k!} \leqslant n + \sum_{i=1}^{n}\sum_{k\geqslant 1}\frac{(\mid \gamma_i \mid)^k}{k!} \\ &= n + \sum_{k\geqslant 1}\frac{1}{k!}\sum_{i=1}^{n}\left[(\gamma_i)^2\right]^{\frac{k}{2}} \leqslant n + \sum_{k\geqslant 1}\frac{1}{k!}\left[\sum_{i=1}^{n}(\gamma_i)^2\right]^{\frac{k}{2}} \\ &= n + \sum_{k\geqslant 1}\frac{1}{k!}\Big(8\chi_{-2}(G)\Big)^{\frac{k}{2}} = n - 1 + \sum_{k\geqslant 0}\frac{\left(\sqrt{8\chi_{-2}(G)}\right)^k}{k!},\end{aligned}$$

which directly leads to the right-hand side inequality in (40). By this the proof of Theorem 16 is completed. □

Theorem 17. *Let G be a graph with n vertices.*

$$\mathcal{HEE}(G) \leqslant n - 1 + e^{\sqrt[4]{N_4}}.$$

Proof. By definition of the Harmonic Estrada index, we have

$$\begin{aligned}\mathcal{HEE}(G) = \sum_{i=1}^{n} e^{\gamma_i} = \sum_{i=1}^{n}\sum_{k=0}^{\infty}\frac{\gamma_i^k}{k!} &\leqslant n + \sum_{i=1}^{n}\sum_{k=1}^{\infty}\frac{\mid \gamma_i \mid^k}{k!} = \\ &= n + \sum_{k=1}^{\infty}\frac{1}{k!}\sum_{i=1}^{n}(\gamma_i^4)^{\frac{k}{4}} \\ &\leqslant n + \sum_{k=1}^{\infty}\frac{1}{k!}\Big(\sum_{i=1}^{n}\gamma_i^4\Big)^{\frac{k}{4}} = \\ &= n + \sum_{k=1}^{\infty}\frac{1}{k!}H^{\frac{k}{4}} = \\ &= n - 1 + \sum_{k=0}^{\infty}\frac{\sqrt[4]{N_4^k}}{k!} = \\ &= n - 1 + e^{\sqrt[4]{N_4}}.\end{aligned}$$

□

Theorem 18. *Let G be a graph with n vertices. Then*

$$\mathcal{HEE}(G) \leqslant e^{\sqrt{8\chi - 2(G)}}. \tag{45}$$

Proof. By definition of Harmonic Estrada index, we have

$$\begin{aligned}\mathcal{HEE}(G) = \sum_{i=1}^{n} e^{\gamma_i} \leqslant \sum_{i=1}^{n} e^{|\gamma_i|} &= \sum_{i=1}^{n}\sum_{k\geqslant 0} \frac{(\mid \gamma_i \mid)^k}{k!} = \sum_{k\geqslant 0}\frac{1}{k!}\sum_{i=1}^{n}(\mid \gamma_i \mid)^k \\ &\leq \sum_{k\geqslant 0}\frac{1}{k!}(\sum_{i=1}^{n}(\mid \gamma_i \mid)^2)^{\frac{k}{2}} \qquad \text{(by Inequality 11)} \\ &= \sum_{k\geqslant 0}\frac{1}{k!}(\sum_{i=1}^{n}(\gamma_i)^2)^{\frac{k}{2}} \\ &= \sum_{k\geqslant 0}\frac{1}{k!}(8\chi_{-2}(G))^{\frac{k}{2}} \qquad (\text{ by Equality 7}) \\ &= \sum_{k\geqslant 0}\frac{1}{k!}(\sqrt{8\chi_{-2}(G)})^k = e^{\sqrt{8\chi_{-2}(G)}}.\end{aligned}$$

□

Theorem 19. *Let G be a graph with n vertices. Then*

$$\mathcal{HEE}(G) \geqslant \sqrt{n^2 + 8n\chi_{-2}(G) + \frac{32n\chi_{-2}(G)\Big(\sum\limits_{k\sim i,\, k\sim j}\frac{1}{(d_k)^2}\Big)}{3}}. \quad (46)$$

Proof. Suppose that $\gamma_1, \gamma_2, \ldots, \gamma_n$ is the spectrum of G. Using the definition of the Harmonic Estrada index and Lemma 5 we have

$$\begin{aligned}\mathcal{HEE}(G)^2 &= \sum_{i=1}^{n}\sum_{j=1}^{n} e^{\gamma_i+\gamma_j} \\ &\geqslant \sum_{i=1}^{n}\sum_{j=1}^{n}\left(1+\gamma_i+\gamma_j+\frac{(\gamma_i+\gamma_j)^2}{2}+\frac{(\gamma_i+\gamma_j)^3}{6}\right) \\ &= \sum_{i=1}^{n}\sum_{j=1}^{n}\Big(1+\gamma_i+\gamma_j+\frac{\gamma_i^2}{2}+\frac{\gamma_j^2}{2}+\gamma_i\gamma_j+ \\ &+\frac{\gamma_i^3}{6}+\frac{\gamma_j^3}{6}+\frac{\gamma_i^2\gamma_j}{2}+\frac{\gamma_i\gamma_j^2}{2}\Big).\end{aligned}$$

Now, by Equality (6), $\sum_{i=1}^{n}\sum_{j=1}^{n}(\gamma_i+\gamma_j) = n\sum_{i=1}^{n}\gamma_i + n\sum_{j=1}^{n}\gamma_j = 0$, $\sum_{i=1}^{n}\sum_{j=1}^{n}\gamma_i\gamma_j = (\sum_{i=1}^{n}\gamma_i)^2 = 0$.
By Equality (7),

$$\sum_{i=1}^{n}\sum_{j=1}^{n}(\frac{\gamma_i^2}{2}+\frac{\gamma_j^2}{2}) = \frac{n}{2}\sum_{i=1}^{n}\gamma_i^2 + \frac{n}{2}\sum_{j=1}^{n}\gamma_j^2 = 8n\chi_{-2}(G).$$

Similarly by Equality (8),

$$\sum_{i=1}^{n}\sum_{j=1}^{n}(\frac{\gamma_i^3}{6}+\frac{\gamma_j^3}{6}) = \frac{n}{6}\sum_{i=1}^{n}\gamma_i^3+\frac{n}{6}\sum_{j=1}^{n}\gamma_j^3 = \frac{32n\chi_{-2}(G)\Big(\sum\limits_{k\sim i,\, k\sim j}\frac{1}{(d_k)^2}\Big)}{3}.$$

By Equality (6),

$$\sum_{i=1}^{n}\sum_{j=1}^{n}\frac{\gamma_i\gamma_j^2}{2} = \frac{1}{2}\sum_{i=1}^{n}\gamma_i\sum_{j=1}^{n}\gamma_j^2 = 0,$$

$$\sum_{i=1}^{n}\sum_{j=1}^{n}\frac{\gamma_i^2\gamma_j}{2} = \frac{1}{2}\sum_{i=1}^{n}\gamma_i^2\sum_{j=1}^{n}\gamma_j = 0.$$

Combining the above relations, the proof is completed. □

5 Summary and conclusions

For a graph of order n, the Harmonic matrix is defined as the square matrix whose (i,j)- element is equal to the sum $\frac{2}{d(u)+d(v)}$ of degrees of adjacent vertices u and v , and zero otherwise. In this paper we obtain some new bounds for the Harmonic Energy and Harmonic Estrada index of graphs.

Acknowledgment

The author is very grateful to the referees for helpful comments and suggestions, which improved the presentation of the original manuscript.

References

[1] C. Adiga and R. K. Zaferani, "Upper bounds for energy of a graph," *Adv. Stud. Cont. Math.*, vol. 16, pp. 279–285, 2008.

[2] N. Alawiah, N. J. Rad, A. Jahanbani, and H. Kamarulhaili, "New Upper Bounds on the Energy of a Graph," *MATCH Commun. Math. Comput. Chem.*, vol.79, pp. 287–301, 2018.

[3] J. Amalorpava Jerline and L. Benedict Michaelraj, "On a Conjecture of Harmonic Index and Diameter of Graphs," *Kragujevac Journal of Mathematics*, vol. 40, pp. 73–78, 2016.

[4] J. Amalorpava Jerline and L. Benedict Michaelraj, "On harmonic index and diameter of unicyclic graphs," *Iranian Journal of Mathematical Sciences and Informatics*, vol. 11, pp. 115–122, 2016.

[5] J. Amalorpava Jerline, L. Benedict Michaelraj, K. Dhanalakshmi, and P. Syamala, "Harmonic index of graphs with more than one cut-vertex," *Ars Combinatoria*, to be published.

[6] H. Bamdad,"New lower bounds for estrada endex," *Bull. Malays. Math. Sci. Soc.*, vol. 39, pp. 683–688, 2016.

[7] M. Biernacki, H. Pidek, and C. R. Nardzewski, "Sur une inègalitè entre des intègrales dèfinies," *Annales Universitatis Mariae Curie-Sklodowska*, vol. 4, pp. 1–4, 1950.

[8] D. Cvetkovič, M. Doob, and H. Sachs, *Spectra of Graphs-Theory and Application*, Johann Ambrosius Barth, 1995.

[9] H. Deng, S. Balachandran, S .K.Ayyaswamy, and Y. B. Venkatakrishnan, "On harmonic indices of trees, unicyclic graphs and bicyclic graphs," *Ars Combinatoria*, to be published.

[10] H. Deng, S.Balachandran, S .K.Ayyaswamy, and Y. B.Venkatakrishnan,"Sharp bounds of the harmonic index of a graph with given girths," *Filomat*, to be published.

[11] H. Deng, S. Balachandran, S. K.Ayyaswamy, and Y. B.Venkatakrishnan, "On the harmonic index and the chromatic number of a graph," *Discrete Appl. Math.*, vol. 161, pp. 2740–2744, 2013.

[12] J. A. De la Peña, I. Gutman, and J. Rada, "Estimating the Estrada Index," *Lin. Algebra Appl.*, vol. 427, pp. 70–76, 2007.

[13] H. Deng, S. Radenkovič, and I. Gutman, "The Estrada Index," in *Applications of Graph Spectra*, D. Cvetkovič and I. Gutman, Eds. Belgrade: Math. Inst., 2009, pp. 123–140.

[14] J. B. Diaz and F. T. Metcalf, "Stronger forms of a class of inequalities of G. Pòlya - G. Szego, L. V. Kantorovich," *Bull. Amer. Math. Soc.*, vol. 69, pp. 415–418, 1963.

[15] Z. Du and B. Zhou, "On sum-connectivity index of bicyclic graphs," *Bull. Malays. Math. Sci. Soc.*, to be published.

[16] Z. Du, B. Zhou and N. Trinajstič, " Minimum general sum-connectivity index of unicyclic graphs," *J. Math. Chem.*, vol. 48, pp. 697–703, 2010.

[17] Z. Du, B. Zhou and N. Trinajstič, "Minimum sum-connectivity indices of trees and unicyclic graphs of a given matching number," *J. Math. Chem.*, vol. 47, pp. 842–855, 2010.

[18] E. Estrada and J. A. Rodrìguez-Velàzguez, "Subgraph Centrality in Complex Networks," *Phys. Rev. E.* vol 71, pp. 056103–056103-9, 2005.

[19] O. Favaron, M. Mahò, and J. F. Saclè, "Some Eigenvalue Properties in Graphs (Conjectures of Graffiti-II)," *Discrete Mathematics*, vol. 111, pp. 197–220, 1993.

[20] D. Gungor and S. B. Bozkurt, "On the distance Estrada index of graphs," *Hacettepe Journal of Mathematics and Statistics*, vol. 38, pp. 277–283, 2009.

[21] I. Gutman and O. E. Polansky, *Mathematical Concepts in Organic Chemistry*, Berlin: Springer, 1986.

[22] I. Gutman, M. Milun, and N. Trinajstič, "Comment on the paper: Properties of the latent roots of a matrix. Estimation of π-electron energies by B. J. McClelland," *J. Chem. Phys.*, vol.59, pp. 2772–2774, 1973.

[23] I. Gutman, "The energy of a graph: old and new results," in *Algebraic Combinatorics and Applications*, A. Betten, A. Kohnert, R. Laue, and A. Wassermann, Eds. Berlin: Springer-Verlag, 2001, pp. 196–211.

[24] I. Gutman, "The energy of a graph," *Ber. Math. Stat. Sckt. Forschungzentrum Graz*, vol. 103, pp. 1–22, 1978.

[25] R. A. Horn and C. R. Johnson, *Matrix Analysis*, New York: Cambridge Univ. Press, 1985.

[26] L. C. Hsu and X. H. Wang, *The Methods and Examples of Mathematical Analysis*, Higher Education Press, 1988.

[27] S. M. Hosamani, B. B. Kulkarni, R. G. Boli, and V. M. Gadag, "QSPR analysis of certain graph theoretical matrices and their corresponding energy," *Appl. Math. Nonlin. Sci.*, vol.2, pp. 131–150, 2017.

[28] A. Ilic, "Note on the harmonic index of a graph," *Applied Mathematics Letters.*, vol. 25, pp. 561–566, 2012.

[29] N. Jafari Rad, A. Jahanbani, and D. A. Mojdeh, "Tetracyclic Graphs with Maximal Estrada Index," *Discrete Mathematics, Algorithms and Applications,* vol. 7, pp. 1750041, 2017.

[30] N. Jafari Rad, A. Jahanbani, and R. Hasni, "Pentacyclic Graphs with Maximal Estrada Index," *Ars Combin.*, vol. 133, pp. 133–145, 2017.

[31] N. J. Rad, A. Jahanbani, and I. Gutman, "Zagreb Energy and Zagreb Estrada Index of Graphs," *MATCH Commun. Math. Comput. Chem.*, vol. 79, pp. 371–386, 2018.

[32] A. Jahanbani, "Upper bounds for the energy of graphs," *MATCH Commun. Math. Comput. Chem.*, vol. 79, pp. 275–286, 2018.

[33] A. Jahanbani, "Some new lower bounds for energy of graphs," *Applied Mathematics and Computation*, vol. 296, no. C, pp. 233–238, 2017.

[34] A. Jahanbani, "Lower bounds for the energy of graphs," *AKCE International Journal of Graphs and Combinatorics*, vol. 15, pp. 88–96, 2018.

[35] A. Jahanbani and H. H. Raz, "On the Harmonic energy and theHarmonic Estrada index of graphs," *MATI.*, vol. 1, no. 1, pp. 1–20, 2019. Retrieved from http://dergipark.gov.tr/mati/issue/38227/425047.

[36] J. Liu, "On the Harmonic index of triangle-free graphs," *Applied Mathematics*, vol. 4, pp. 1204–1206, 2013.

[37] A. Lupas, "Inequalities for the roots of a class of polynomials," *Publ. Elektrotehn. Fak. Ser. Math. Fiz.*, vol. 594, pp. 79–85, 1977.

[38] R. Wu, Z. Tang, and H. Deng, "A Lower Bound for the Harmonic Index of a Graph with Minimum Degree at Least Two," *Filomat.*, vol. 27, pp. 51–55, 2013.

[39] L. Zhong, "The harmonic index for graphs," *Appl. Math. Lett.*, vol. 25, pp. 561–566, 2012.

[40] L. Zhong, "The harmonic index on unicyclic graphs," *Ars Combinatoria*, vol. 104, pp. 261–269, 2012.

[41] L. Zhong and K. Xu, "The harmonic index on bicyclic graphs," *Utilitas Mathematica*, vol. 90, pp. 23–32, 2013.

[42] B. Zhou and N. Trinajstiċ, "On a novel connectivity index," *J. Math. Chem.*, vol. 46, pp. 1252–1270, 2009.

[43] B. Zhou and N. Trinajstiċ, "On general sum-connectivity index," *J. Math. Chem.*, vol. 47, pp. 210–218, 2010.

Akbar Jahanbani

Received June 10, 2018

Department of Mathematics,
Shahrood University of Technology,
Shahrood, Iran

E–mail: akbar.jahanbani92@gmail.com

Computer Science Journal of Moldova, vol.**26**, no.3(78), 2018

Post-quantum signature algorithms based on the hidden discrete logarithm problem

A.A. Moldovyan N.A. Moldovyan

Abstract

New options of the hidden discrete logarithm problem are proposed as cryptographic primitive of the post-quantum signature algorithms. Two signature schemes using computations in finite non-commutative algebras with associative multiplication operation are introduced. The main feature of the proposed signature algorithms consists in using locally invertible elements of algebras. Two different types of algebras are used: i) containing global bi-side unit and ii) containing a large set of global right-side units.

Keywords: finite associative algebra, non-commutative algebra, global unit, local unit, right-side units, local invertibility, discrete logarithm problem, public-key cryptoscheme, digital signature, post-quantum cryptography.

MSC 2000: 94A60, 16Z05, 14G50, 11T71, 16S50.

1 Introduction

Development of the post-quantum public-key cryptographic algorithms and protocols is a current challenge of the applied and theoretic cryptography [1], [2]. A well-known response to this challenge is the competition for the development of the post-quantum public-key cryptoschemes, announced by NIST in 2016 [3]. The current outcome of this competition is a set of post-quantum cryptoschemes, chosen as candidates for the adoption of post-quantum cryptographic standards on their basis [4]. At the next stage it is expected to initiate an all-round discussion of candidates from the wide cryptographic community. It is

assumed that the main point at this three-year stage will be research on the resistance of selected candidates to attacks using a hypothetical quantum computer.

The problem of discrete logarithm in a hidden cyclic group [5] remained outside the attention of NIST participants, although it seems to be an interesting primitive for constructing practical post-quantum cryptoschemes. Apparently this is due to the fact that in the literature this problem, which can be called the hidden discrete logarithm problem (HDLP), is little illuminated. Another reason is a relatively small number of known carriers of this problem, which are finite noncommutative associative algebras (FNAAs). The urgency of the problem of finding new carriers of the HDLP is underlined in the paper [6].

The purpose of this work is to attract the attention of cryptographic community to the HDLP as a post-quantum cryptographic primitive. To achieve this goal, new types of algebras are being developed (Section 2), new variants of the HDLP are introduced (Section 3), and algorithms for digital signature based on the proposed options of the HDLP are being developed for the first time (Section 4). In the concluding Section 5 we estimate that the application of the HDLP for the design of the post-quantum public-key algorithms and protocols is a promising direction.

2 New carriers of the HDLP

The known option of the HDLP [5] is formulated in a finite noncommutative group Γ as follows. Suppose the group Γ contains elements Q and G having large prime order q and satisfying the condition $G \circ Q \neq Q \circ G$. In [5] it is proposed to compute a public key Y as follows $Y = G^w \circ Q^x \circ G^{-w}$, where the pair of integers (w, x) represents the private key. Computing the values $w < q$ and $x < q$, while the elements Y, Q, and G are known, is called HDLP. The paper [5] decribes the public key-agreement protocol, the public encryption algorithm, and the commutative cipher based on the HDLP formulated in the finite algebra of quaternions. No proposals for digital signature schemes are known in the literature.

While considering new FNAAs of the dimensions $m = 4$ and $m = 6$ the following common description is used. The m-dimensional vector space defined over the finite field $GF(p)$ becomes the m-dimensional finite algebra after the operation for multiplying arbitrary two vectors is defined, which is distributive relatively the addition operation. To set the multiplication operation one can use the notion of formal basis vectors denoted as $\mathbf{e}_0 = (1, 0, 0 \ldots, 0)$, $\mathbf{e}_1 = (0, 1, 0 \ldots, 0)$, ... $\mathbf{e}_{m-1} = (0, 0 \ldots, 0, 1)$ and representation of some vector $A = (a_0, a_1, \ldots a_{m-1})$ in the form of the following summ of the single component vectors $a_i\mathbf{e}_i$: $A = \sum_{i=0}^{m-1} a_i\mathbf{e}_i$.

The multiplication operation $\circ$ of the m-dimensional vectors A and $B = \sum_{j=0}^{m-1} b_j\mathbf{e}_j$ is defined by the following formula

$$A \circ B = \left(\sum_{i=0}^{m-1} a_i\mathbf{e}_i\right) \circ \left(\sum_{j=0}^{m-1} b_j\mathbf{e}_j\right) = \sum_{j=0}^{m-1}\sum_{i=0}^{m-1} a_i b_j \left(\mathbf{e}_i \circ \mathbf{e}_j\right), \qquad (1)$$

where product $\mathbf{e}_i \circ \mathbf{e}_j$ for all possible pairs of the integers i and j is to be replaced by some single-component vector $\lambda\mathbf{e}_k$ indicated by so called basis vector multiplication table (BVMT). In formula (1) it is ussumed that the intersection of the ith row and the jth column defines the cell which contains the value $\lambda\mathbf{e}_k = \mathbf{e}_i \circ \mathbf{e}_j$. If the coordinate $\lambda \neq 1$, then λ is called structural coefficient. To build a FNAA we should compose and use some BVMT defining non-commutative associative multiplication operation.

2.1 The 4-dimensional FNAA

In this subsection we summarize in brief some results of the paper [7] relating to the case of the 4-dimensional non-commutative algebra. The FNAA defined by Table 1, where $\tau \neq 1$, contains the global bi-side unit

$$E = \left(\frac{1}{1-\tau}, \frac{1}{1-\tau}, \frac{\tau}{\tau-1}, \frac{1}{\tau-1}\right)$$

for which for every element of the algebra the formulas $A \circ E = A$ and $E \circ A = A$ hold. Every vector A of the algebra, coordinates of which

Table 1. The BVMT for defining the 4-dimensional algebra with the global unit ($\tau \neq 1$).

$\circ$	$\mathbf{e}_0$	$\mathbf{e}_1$	$\mathbf{e}_2$	$\mathbf{e}_3$
$\mathbf{e}_0$	$\mathbf{e}_0$	$\mathbf{e}_3$	$\mathbf{e}_0$	$\mathbf{e}_3$
$\mathbf{e}_1$	$\tau\mathbf{e}_2$	$\mathbf{e}_1$	$\mathbf{e}_2$	$\tau\mathbf{e}_1$
$\mathbf{e}_2$	$\mathbf{e}_2$	$\mathbf{e}_1$	$\mathbf{e}_2$	$\mathbf{e}_1$
$\mathbf{e}_3$	$\tau\mathbf{e}_0$	$\mathbf{e}_3$	$\mathbf{e}_0$	$\tau\mathbf{e}_3$

satisfy the condition $a_0a_1 \neq a_2a_3$, is invertible relatively the global unit E, i.e., for arbitrary vector of such a kind there exists the vector A^{-1} such that $A \circ A^{-1} = A^{-1} \circ A = E$ holds. The vectors B satisfying the condition $b_0b_1 = b_2b_3$ are non-invertible relatively the global unit E. However, the majority of vectors B are locally invertible, i.e., invertible relatively some local bi-side unit E_B acting as unit element in some subset of the algebra elements, which includes the vector B. Evidently, this subset is a finite group with the group operation $\circ$. There exists the single local bi-side unit in the subset of the non-invertible (globally) algebra elements. However, for some fixed non-invertible vector B there exists a large set of the vectors E'_B satisfying the condition $E'_B \circ B = B$. This set of the vectors E'_B can be called the set of the left-side units of the vector B and is described by the following formula:

$$E'_B = \left(d,\ \frac{b_2}{b_0+b_2} - \frac{b_0+b_2}{\tau b_0+b_2}h,\ h,\ \frac{b_0}{\tau b_0+b_2} - \frac{b_0+b_2}{\tau b_0+b_2}d \right), \qquad (2)$$

where $h, d = 0, 1, \ldots p-1$. Analogously, for the vector B there exists a large set of the vectors E''_B satisfying the condition $B \circ E''_B = B$. The last set can be called the set of the right-side units of the vector B and is described by the following formula:

$$E''_B = \left(d,\ \frac{b_3}{b_0+b_3} - \frac{b_0+\tau b_3}{b_0+b_3}h,\ \frac{b_0}{b_0+b_3} - \frac{b_0+\tau b_3}{b_0+b_3}d,\ h \right), \qquad (3)$$

where $h, d = 0, 1, \ldots p-1$.

The existence of many local units associated with a given non-invertible vector is an essential point in setting a new form of the HDLP proposed in Section 3.1. Earlier [8] the use of the non-invertible elements had been proposed in frame of the known form of the HDLP [5], however in that proposal there are not exploited the local units related to the used non-invertible element B.

2.2 The 6-dimensional FNAA

If the structural coefficients λ, μ, and τ in Table 2 satisfy the following two conditions $\mu \neq \tau$ and $\mu \neq \lambda\tau$, then this BVMT defines the multiplication operation in the 6-dimensional FNAAs containing a large set of the global right-side units. For some right-side unit X acting on the vector A the following vector equation holds:

$$A \circ X = A. \tag{4}$$

Using Table 2 one can represent (4) in the form of the following system of six linear equations with coordinates of the right operand $x_0, x_1, \ldots, x_5$ as the unknown values:

$$\begin{cases} a_0x_0 + \tau a_0x_2 + a_0x_4 + \lambda a_5x_0 + \mu a_5x_2 + a_5x_4 = a_0; \\ a_1x_1 + \mu a_1x_3 + \lambda a_1x_5 + a_4x_1 + \tau a_4x_3 + a_4x_5 = a_1; \\ a_2x_0 + \tau a_2x_2 + a_2x_4 + \lambda a_3x_0 + \mu a_3x_2 + a_3x_4 = a_2; \\ a_2x_1 + \tau a_2x_3 + a_2x_5 + a_3x_1 + \mu a_3x_3 + \lambda a_3x_5 = a_3; \\ a_0x_0 + \tau a_0x_2 + a_0x_4 + \lambda a_5x_0 + \mu a_5x_2 + a_5x_4 = a_4; \\ a_0x_1 + \tau a_0x_3 + a_0x_5 + a_5x_1 + \mu a_5x_3 + \lambda a_5x_5 = a_5. \end{cases} \tag{5}$$

The system (5) can be rewritten as the following system of six equations with four unknowns: $z_1 = x_0 + \tau x_2 + x_4$; $z_2 = \lambda x_0 + \mu x_2 + x_4$; $z_3 =$

$x_1 + \tau x_3 + x_5$; $z_4 = x_1 + \mu x_3 + \lambda x_5$:

$$\begin{cases} a_0 z_1 + a_5 z_2 = a_0; \\ a_1 z_4 + a_4 z_3 = a_1; \\ a_2 z_1 + a_3 z_2 = a_2; \\ a_2 z_3 + a_3 z_4 = a_3; \\ a_1 z_2 + a_4 z_1 = a_4; \\ a_0 z_3 + a_5 z_4 = a_5. \end{cases} \tag{6}$$

The system (6) has the following single solution:

$$z_1 = 1; \quad z_2 = 0; \quad z_3 = 0; \quad z_4 = 1. \tag{7}$$

Using (7) one can write the following two independent systems:

$$\begin{cases} x_0 + \tau x_2 + x_4 = 1; \\ \lambda x_0 + \mu x_2 + x_4 = 0; \end{cases} \tag{8}$$

$$\begin{cases} x_1 + \mu x_3 + \lambda x_5 = 1; \\ x_1 + \tau x_3 + x_5 = 0. \end{cases} \tag{9}$$

Solution of the systems (8) and (9) defines all solutions of the system (5). The lasts can be described by the following formula:

$$R = \left(x_0, \quad x_1, \quad \frac{1 + (\lambda - 1)x_0}{\tau - \mu}, \quad \frac{1 + (\lambda - 1)x_1}{\mu - \lambda\tau}, \right. \\ \left. \frac{(\mu - \lambda\tau)x_0 - \mu}{\tau - \mu}, \quad \frac{(\tau - \mu)x_1 - \tau)}{\mu - \lambda\tau} \right), \tag{10}$$

where $x_0, x_1 = 0, 1, \ldots p - 1$. Thus, the solutions of the system (5) do not depend on the value A, therefore formula (10) describes the full set of the global right-side units in the considered FNAA.

One can easily prove the following propositions:

Proposition 1. For arbitrary global right-side unit R_i and arbitrary integer n the following equation holds $R_i^n = R_i$.

Propossition 2. For arbitrary two 6-dimensional vectors U and T such that $U \circ T = R_i$, where R_i is a global right-side unit, and arbitrary integer n the following equation holds $U^n \circ T^n = R_i$.

Propossition 3. For arbitrary global right-side unit R_i, arbitrary 6-dimensional vector U, and arbitrary integer n the following equation holds $(R_i \circ U)^n = R_i \circ U^n$.

Proposition 4. Arbitrary global right-side unit R_i is simultaneously the single local bi-side unit E_A for the vector $R_i \circ A$, where A is an arbitrary non-zero vector.

Table 2. The BVMT defining the 6-dimensional FNAA with p^2 different global right-side units ($\lambda \neq 1$, $\mu \neq 1$, $\tau \neq 1$, $\tau \neq \mu$, $\lambda\tau \neq \mu$)

$\circ$	$\mathbf{e}_0$	$\mathbf{e}_1$	$\mathbf{e}_2$	$\mathbf{e}_3$	$\mathbf{e}_4$	$\mathbf{e}_5$
$\mathbf{e}_0$	$\mathbf{e}_0$	$\mathbf{e}_5$	$\tau\mathbf{e}_0$	$\tau\mathbf{e}_5$	$\mathbf{e}_0$	$\mathbf{e}_5$
$\mathbf{e}_1$	$\lambda\mathbf{e}_4$	$\mathbf{e}_1$	$\mu\mathbf{e}_4$	$\mu\mathbf{e}_1$	$\mathbf{e}_4$	$\lambda\mathbf{e}_1$
$\mathbf{e}_2$	$\mathbf{e}_2$	$\mathbf{e}_3$	$\tau\mathbf{e}_2$	$\tau\mathbf{e}_3$	$\mathbf{e}_2$	$\mathbf{e}_3$
$\mathbf{e}_3$	$\lambda\mathbf{e}_2$	$\mathbf{e}_3$	$\mu\mathbf{e}_2$	$\mu\mathbf{e}_3$	$\mathbf{e}_2$	$\lambda\mathbf{e}_3$
$\mathbf{e}_4$	$\mathbf{e}_4$	$\mathbf{e}_1$	$\tau\mathbf{e}_4$	$\tau\mathbf{e}_1$	$\mathbf{e}_4$	$\mathbf{e}_1$
$\mathbf{e}_5$	$\lambda\mathbf{e}_0$	$\mathbf{e}_5$	$\mu\mathbf{e}_0$	$\mu\mathbf{e}_5$	$\mathbf{e}_0$	$\lambda\mathbf{e}_5$

Computation of the local bi-side unit E_A relating to the vector A can be executed as finding the local left-side unit of the vector A from the vector equation $X \circ A = A$, i.e., from the following system of six equations with six unknowns:

$$\begin{cases} (a_0 + \tau a_2 + a_4)\, x_0 + (\lambda a_0 + \mu a_2 + a_4)\, x_5 = a_0; \\ (a_1 + \mu a_3 + \lambda a_5)\, x_1 + (a_1 + \tau a_3 + a_5)\, x_4 = a_1; \\ (a_0 + \tau a_2 + a_4)\, x_2 + (\lambda a_0 + \mu a_2 + a_4)\, x_3 = a_2; \\ (a_1 + \tau a_3 + a_5)\, x_2 + (a_1 + \mu a_3 + \lambda a_5)\, x_3 = a_3; \\ (\lambda a_0 + \mu a_2 + a_4)\, x_1 + (a_0 + \tau a_2 + a_4)\, x_4 = a_4; \\ (a_1 + \tau a_3 + a_5)\, x_0 + (a_1 + \mu a_3 + \lambda a_5)\, x_5 = a_5. \end{cases} \tag{11}$$

It is easy to see that for the vectors A satisfying the condition

$$\begin{aligned}\Delta_A = (a_4a_5 - a_0a_1)(\lambda - 1) + (a_3a_4 - a_1a_2)(\mu - \tau) + \\ + (a_2a_5 - a_0a_3)(\lambda\tau - \mu) \neq 0\end{aligned} \tag{12}$$

the system (11) has the single solution, i.e., the single local left-side unit $L_A \neq (0, 0, \ldots, 0)$ relates to every such vector A.

Proposition 5. All vectors A satisfying the condition (12) and relating to a fixed local bi-side unit E_A compose a finite group (with the unit E_A and the group operation $\circ$) contained in the considered 6-dimensional FNAA.

Earlier [9] the analogous 6-dimensional FNAA containing p^4 different global right-side units have been considered, however units of such type had not been used at defining the HDLP.

3 Novel forms of defining the HDLP

3.1 The HDLP in FNAA containing the global unit

Over a FNAA with the global unit (for example, over the 4-dimensional algebra considered in Subsection 2.1) one can define the HDLP as follows. Suppose the vector B is a non-invertible one and has sufficiently large prime local order ω, the invertible vectors G and H are such that the following conditions $G \circ B \neq B \circ G$, $G \circ H \neq H \circ G$, and $H \circ B \neq B \circ H$ hold. Then one can select at random an integer $x < \omega$ and an invertible vector E from the sets of the local single-side units of the vector B, i.e., from the sets (2) and (3) in the case of considered 4-dimensional FNAA, and compute the vectors Z, Y, and T:

$$Z = H \circ B \circ H^{-1}; \quad Y = G \circ B^x \circ G^{-1}; \quad T = G \circ E \circ H^{-1}. \tag{13}$$

The triple (Z, Y, T) can be used as public key to which the private key representing the set of values x, B, H, and G corresponds. The value E is also secret, however it is used only at step of computing the public-key element T. Computationally difficult problem consists in finding the private key or alternative four values x', B', H', and G' with which the public key can be expressed in accordance with the formulas (13).

3.2 The HDLP in FNAA containing large set of the global single-side units

Over a FNAA with the set of the single-side global units (for example, over the 6-dimensional algebra considered in Subsection 2.2, which contains a large set of the global right-side units) an option of the HDLP can be defined as follows. Suppose the vector A satisfying the condition (12) has sufficiently large prime local order ω and the vectors G, P, H, and Q are selected so that the following conditions $G \circ A \neq A \circ G$, $H \circ A \neq A \circ H$, $P \circ G = R_1$, and $Q \circ H = R_2$, where R_1 and $R_2 \neq R_1$ are arbitrary global right-side units, hold. Then one can select at random an integer $x < \omega$ and a global right-side unit R_3, such that $R_3 \neq R_2$ and $R_3 \neq R_1$, and compute the triple of the vectors Z, Y, and T satisfying the following equations:

$$Z = H \circ A \circ Q; \quad Y = G \circ A^x \circ P; \quad P \circ T \circ H = R_3. \qquad (14)$$

The triple (Z, Y, T) represents a public key connected with the private key representing the set of values x, G, A, and Q. The values P, H, and R_3 are also secret, however they are needed to the owner of the public key only in frame of the process of computing the values Z, Y, and T.

Finding the private key or some alternative four values x', G', A', and Q', with which the public key can be expressed in accordance with the formulas (14), represents a difficult computation problem. The last is called HDLP due to using the exponentiation operation performed in the finite cyclic group generated by the vector A, which is hidden in the FNAA. The used exponentiation operation contributes significantly to the difficulty of the considered variants of the HDLP.

4 Digital signature algorithms

In the case of using the HDLP introduced in Subsection 3.1 and the 4-dimensional FNAA described in Subsection 2.1 and defined over the field $GF(p)$ with 512-bit prime p one can propose the following signa-

ture generation algorithm in which some specified hash function F_h is used:

1. Generate a uniformly random value $k < \omega$ and compute the vector $V = G \circ B^k \circ H^{-1}$.

2. Compute the first signature element $e = F_h(M, V)$, where M is the electronic document to be signed.

3. While interpreting the bit string e as a binary number, compute the second signature element $s = k - xe \bmod \omega$.

The respective signature verification algorithm is performed as follows:

1. Using the signature (e, s) to the document M, compute the vector V': $V' = Y^e \circ T \circ Z^s$.

2. Compute the hash value $e' = F_h(M, V')$.

3. If $e' = e$, then the signature is accepted as genuine. Otherwise the signature is rejected.

Correctness proof of the proposed signature scheme is as follows:

$$V' = \left(G \circ B^x \circ G^{-1}\right)^e \circ T \circ \left(H \circ B \circ H^{-1}\right)^{(k-xe)} =$$
$$G \circ B^{xe} \circ G^{-1} \circ T \circ H \circ B^{k-xe} \circ H^{-1} = G \circ B^{xe} \circ E \circ B^{k-xe} \circ H^{-1} =$$
$$G \circ B^{xe+k-xe} \circ H^{-1} = G \circ B^k \circ H^{-1} = V \;\Rightarrow\; e' = e.$$

While using the HDLP, set in the 6-dimensional FNAA defined over the field $GF(p)$ with 384-bit characteristic p (see Subsection 3.2), the following algorithm can be proposed for generating a signature to document M:

1. Select at random an integer $k < \omega$ and compute the vector $V = G \circ A^k \circ Q$.

2. Compute the first signature element $v = F_h(V)$, where F_h is the used hash function.

3. Compute the hash function value e from the document M and the second signature element s: $e = F_h(M)$ and $s = ke - xv \bmod \omega$.

Verification of the signature (v, s) to the document M is to be performed with the following algorithm:

1. Compute the hash value e from the document $e = F_h(M)$.

2. Compute the vector $V' : V' = Y^{ve^{-1}} \circ T \circ Z^{se^{-1}}$.

3. Compute the hash value v' from the vector V': $v' = F_h(V')$.

4. If $v' = v$, then the signature is accepted as genuine. Otherwise the signature is rejected as false one.

Proof of the correctness of the last signature scheme is as follows:

$$V' = (G \circ A^x \circ P)^{ve^{-1}} \circ T \circ (H \circ A \circ Q)^{se^{-1}} =$$
$$G \circ A^{xve^{-1}} \circ P \circ T \circ H \circ A^{se^{-1}} \circ Q =$$
$$G \circ A^{xve^{-1}} \circ R_3 \circ A^{(ke-xv)e^{-1}} \circ Q =$$
$$G \circ A^{xve^{-1}+k-xve^{-1}} \circ Q = G \circ A^k \circ Q = V \Rightarrow v' = v.$$

Like in the Schnorr digital signature protocol [10] and in the discrete logarithm based standards [11], in the described signature schemes there is used some cyclic group of the prime order. However, in the proposed signature algorithms the used cyclic group is hidden in a FNAA. The public part of the introduced new signature algorithms is the used FNAA and three its elements Y, Z, and T that are connected with the hidden cyclic group generated by powers of the hidden-group generator (the vector B in the first signature scheme and the vector A in the second scheme) that is an element of the private key.

5 Conclusion

In this paper, two new FNAAs have been introduced as carries of the HDLP defined in two novel forms. One should note that the proposed two variants of the HDLP suit well to design digital signature schemes, however it is not evident, how they can be used for designing the public key agreement protocols. The known form of the HDLP [5] suits well to design the last type of protocols, however at present no proposals for signature schemes on its base are known. In the compared forms of the HDLP there are used different mechanisms for hiding a cyclic group.

Estimation of the security of the proposed signature algorithms to quantum attacks is connected with estimation of the computational difficulty of the reduction of the used HDLP to the discrete logarithm

problem in $GF(p)$. Consideration of this item represents an individual task. If the polynomial-time algorithms for such reduction will not be found for several years after publication of the proposed forms of the HDLP and signature schemes on their base, then one can hope the attractive candidates for post-quantum signature standards will be available. Significant advantage of the proposed signature schemes relatively the candidates selected in frame of the NIST PQCrypto project [3] is smaller signature size (384 to 512 bits in the case of 128-bit security) and higher performance of the signature generation and verification procedures.

Besides analysis of the resistance to quantum attacks, future development of the performed research can be also related to justification of the parameters of the FNAAs applied as carriers of the HDLP used as cryptographic primitive as well as to justification of the parameters of the HDLP.

References

[1] *Proceedings of the 7th International Workshop on Post-Quantum Cryptography, PQCrypto 2016, Fukuoka, Japan, February 24-26, 2016*, (Lecture Notes in Computer Science, vol. 9606), 2016, 270 p.

[2] *Post-Quantum Cryptography. 9th International Conference, PQCrypto 2018 Proceedings, Fort Lauderdale, FL, USA, April 9-11, 2018*, (Lecture Notes in Computer Science, vol. 10786), 2018.

[3] Federal Information Security Management Act (FISMA) of 2002. *Public Law 107347. Submission Requirements and Evaluation Criteria for the Post-Quantum Cryptography Standardization Process. NIST PQCrypto project.* [Online]. Available: https://csrc.nist.gov/CSRC/media/Projects/Post-Quantum-Cryptography/documents/call-for-proposals-final-dec-2016.pdf

[4] "First NIST standardization conference," April 11–13, 2018. [Online]. Available: http://prometheuscrypt.gforge.inria.fr/2018-04-18.pqc2018.html

[5] D.N. Moldovyan, "Non-Commutative Finite Groups as Primitive of Public-Key Cryptoschemes," *Quasigroups and Related Systems*, vol. 18, no. 2, pp. 165–176, 2010.

[6] A. S. Kuzmin, V. T. Markov, A. A. Mikhalev, A. V. Mikhalev, A. A. Nechaev, "Cryptographic Algorithms on Groups and Algebras," *Journal of Mathematical Sciences*, vol. 223, no. 5, pp. 629–641, 2017.

[7] A. A. Moldovyan, N. A. Moldovyan, V. A. Shcherbacov, "Non-commutative finite rings with several mutually associative multiplication operations," in *Proceedings of The Fourth Conference of Mathematical Society of the Republic of Moldova dedicated to the centenary of Vladimir Andrunachievici (1917-1997), June 28 - July 2, 2017*, (Chisinau), 2017, pp. 133–136.

[8] D. N. Moldovyan, N. A. Moldovyan, "Cryptoschemes over hidden conjugacy search problem and attacks using homomorphisms," *Quasigroups Related Systems*, Vol. 18, no. 2, pp. 177–186, 2010.

[9] A. A. Moldovyan, N. A. Moldovyan, V. A. Shcherbacov, "Non-commutative 6-dimensional associative algebras of two different types," in *Proceedings of the Workshop on Foundations of Informatics, July 2-6, 2018*, (Chisinau), 2018, pp. 154–163.

[10] C. P. Schnorr, "Efficient signature generation by smart cards," *J. Cryptology*, vol. 4, pp. 161–174, 1991.

[11] *Information technology – Security techniques – Digital Signatures with appendix – Part 3: Discrete logarithm based mechanisms*, International Standard ISO/IEC 14888-3:2006(E), 2006.

A. A. Moldovyan, N. A. Moldovyan,

Received September 5, 2018

St. Petersburg Institute for Informatics and Automation of
Russian Academy of Sciences
14 Liniya, 39, St.Petersburg, 199178
Russia
E–mail: nmold@mail.ru

Computer Science Journal of Moldova, vol.**26**, no.3(78), 2018

About Workshop in the framework of NIMSPPS project

Svetlana Cojocaru, Constantin Gaindric, Galina Magariu,
Olga Popcova, Tatiana Verlan

In the period of September 18-19, 2018 the Workshop in the framework of NIMSPPS project "Network for informational methods in supporting persons predisposed to preventable strokes using common devices", took place in Chisinau, Moldova.

The previous NIMSPPS Workshop was organized in Mainz, Germany on May 29-30, 2018 by Johannes Gutenberg University Mainz, where Cooperation Agreement was signed.

The September NIMSPPS Workshop was organized by the Institute of Mathematics and Computer Science. 22 scientists from 5 countries (Austria, Germany, Moldova, Romania and Ukraine) attended this workshop.

The Workshop consisted of 12 following presentations:

- **Stefan Kramer, NIMSPPS Project Director (Johannes Gutenberg University Mainz, Germany)** opened the Workshop with the presentation **"**Towards a Data-Complete Approach to Stroke Prevention (Under Privacy Constraints)". He told about the NIMSPPS Background and Two Phases of Network

Creation. He emphasized that the result of the project work should be a sustainable network of mathematicians, computer scientists, physicians, policy makers in healthcare / stroke prevention, developers of medical software and equipment, SMEs, insurance companies etc. The following approaches should be applyed: Towards a data-complete approach to stroke prevention under privacy constraints; Ideas / concept for putting everything together; Some challenges addressed and potential solutions (for potential future proposals).

- **Daniela Efremova** in her joint report with **Eremei Zota** (Clinic of Neurology and Neurosurgery, Chisinau) presented the results of the project *Specific risk factors for stroke in moldovan population, strategies of primary and secondary prevention (Stroke risk factors among patients hospitalized with stroke: preliminary results from a cross-sectional study in the population of Republic of Moldova*) and argued that Stroke is a major health and social problem. Therefore the need to strengthen all efforts in stroke prevention to reduce its impact on society becomes obvious. She emphasized that many prevalent risk factors are modifiable through lifestyle changes and/or medical therapy. Lack of public awareness about stroke warning signs and risk factors must be addressed as one important contribution to reducing mortality and morbidity from stroke.
- **Tatiana Verlan** with **Svetlana Cojocaru, Constantin Gaindric, Galina Magariu** (The Vladimir Andrunakievich Institute of Mathematics and Computer Science, Chisinau) presented the report which considers the problem of "Analysis and preparation of data from Stroke.md database when creating a stroke prediction model". She accentuated that data preparation when creating a prediction model, especially in medicine, is the most time-consuming and labour-intensive stage. To the most degree the model performance depends on data quality. In the report the applied approaches to the available data preparation were described, and also the main steps used for prediction models creation were presented. Preliminary evaluations of performance

of the models created at different steps were analyzed as well.

- **Silvia Miksch** (Vienna University of Technology, Austria) made a report "Visual Analytics Meet Medicine: How Can We Tackle the Challenges?", in which she presented a short introduction to Visual Analytics as "... the science of analytical reasoning facilitated by interactive visual interfaces" and its goals. Then she presented the selected Challenges for Visual Analytics in Health Care, which include: Scale and Complexity of Time-oriented Data; Intertwining Patient Condition with Treatment Processes; Scalable Analysis from Single Patients to Cohorts; Data Quality and Uncertainty; Interaction, User Interfaces, and the Role of Users; Evaluation; Guidance.
- **Pankratova Nataliya** (Institute for Applied System Analysis of the National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute " (IASA)) in her report told about the institute, its history, structure and aims: embodiment of conception of science and education integration, having the aim of carrying out the pioneer scientific research in the country and training of special purpose of highly qualified specialists for NAS of Ukraine and Ministry of Education of Ukraine. The main directions of scientific activity, especially in medicine, were described. She accentuated about the importance of solving strategic problems of medicine of the future on the basis of the foresight methodology.
- **Iury Timoshenko** (Institute for Applied System Analysis, National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute") made a report "Systems medicine approach to stroke prevention through wellness" in which P4-medicine (Predictive, Preventive, Personalized, Participatory) as a new Health-Care conception was described. He told about the need of cheap, accessible diagnostic testing, and an intelligent mobile application Cardio4U, which allows one to implement personalized preventive heart diagnosis without going to a doctor, was exemplified. Then he presented a system based on PPG (photoplethysmogram) – Getting a heartbeat signal from changing the skin's tone. The speaker invited participants of the Workshop to take part in ex-

periment and demonstrated the work of the system (see Fig1).

Figure 1.

- **Adriana Albu** (Polytechnic University of *Timisoara, Romania*) presented the report "Medical Condition – Predictions and Diagnosis using Artificial Intelligence Methods". One of the aims of the presentation: "to show our concerns and results in the field of AI used in decision-making related to medical domain (prediction and diagnosis)". She also told about several created in the University medical decision-making systems, using: Logical inference (knowledge-based system) for hepatitis; Probabilistic reasoning (Bayes theorem) for hepatitis B; Artificial Neural Networks for therapy selection – hepatitis C, diagnosis using images – liver diseases, diagnosis using laboratory data – skin diseases, predictions – risk of stroke.
- **Victor Cojocaru** with **Teodor Fedorisin** and **Rihart Galus** (D. Ghitu Institute of Electronic Engineering and Nanotechnologies, Chisinau) made a report "Dynamic method of brain cooling", in which they told about mechanisms to protect the brain

during hypothermia. Their work was aimed to design, build and test a new version of a hypothermic therapy device used for medical purposes using the Peltier elements. The authors presented a hypothermia system able to drive independently a large number of Peltier elements and to monitor the skin temperature under them. They also demonstrated the work of the current version of hypothermia control device which can be used for people after brain surgery and for patients in serious trouble (see Fig 2).

Figure 2.

- **Maurice Dann** (Johannes Gutenberg University Mainz) made the report "Application for Supporting Persons Predisposed to Stroke" (joint with Patrick Müller), where he presented the developed by the authors Prototype Android App. The goal of the application is to Help/Support people with a high risk of stroke. The applied approaches are: Development of native Android application; Collect useful information from local/external sensors; Save information in local DB; Synchronize local DB with external DB. The following types of data are used: Sensors (Accelerom-

eter, Magnetometer, Gyroscope, GPS); Weather data (Current weather information, 3 hours forecast weather information); Interpreted data (Daily steps, Hourly steps).

- **Elena Zamsha** (The Vladimir Andrunakievich Institute of Mathematics and Computer Science, Chisinau) presented the report about the current state of the developed "Information System StrokeMD". She described this application as a stroke clustering and prediction system for associated medical data visualization and management for neurologists. The goal of the system is to facilitate efficient visual data introduction and knowledge extraction based on a predictive model implementation. Its main functions include: Identify and register acute stroke cases; Record information related to symptom onset, diagnostic evaluation, acute treatments, discharge status and plan; Develop mechanism for timely transmission of registry data elements; Conduct process evaluation; Prediction; Provide care providers tools needed for prevention, treatment and rehab.
- **Julian Vexler** (Johannes Gutenberg University Mainz) made a report "An Overview of ICT-Architectures for better Patient Treatment", where he gave a comparative analysis of several different devices - support systems for medical diagnosis, patient-centered smart health service platform, etc. Afterwards he told about the concept of the device in the framework of NIMSPPS, described its architecture and main principles of work.
- **Vladimir Popukaylo** (Tiraspol University; The Vladimir Andrunakievich Institute of Mathematics and Computer Science, Chisinau) presented the report "Predicting the occurrence of strokes using the language R". The purpose of this study was to build a stroke prediction model. The transformation and analysis of the data was carried out in the language R using the caret package. He also considered the problem of data preprocessing (preparation and transformation) and selecting predictors.

Representatives from the Polytechnic University of Timisoara, Romania and the Institute for Applied System Analysis of the National

Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute" have expressed a desire to take part in the international consortium for applying to specific calls within EU funding programmes.

The workshops that have taken place contributed to the adjustment and validation of national methodology, technology and knowledge bases in the teams participating in the project.

S. Cojocaru, C. Gaindric, G. Magariu,
O. Popcova, T. Verlan

Received October 24, 2018

Institute of Mathematics and Computer Science
5 Academiei str., MD-2028, Chisinau, Republic of Moldova
E–mails: svetlana.cojocaru@math.md, constantin.gaindric@math.md, galina.magariu@math.md, oleapopcova@yahoo.com, tatiana.verlan@math.md