Proceedings of the 5th
Conference on Mathematical Foundations of Informatics

3-6 July 2019, Iași, Romania

Daniela Gîfu ● Bogdan Aman ● Adrian Iftene ● Diana Trandabăț
(Editors)
Scientific references:

Svetlana Cojocaru Dr. habil., Professor at the Institute of Mathematics and Computer Science Academy of Sciences of Moldova and Corresponding Member of the Academy of Sciences of Moldova

Constantin Gaindric Dr. habil., Professor at the Institute of Mathematics and Computer Science Academy of Sciences of Moldova and Corresponding Member of the Academy of Sciences of Moldova

Coperta: Manuela Oboroceanu


© Editura Universității „Alexandru Ioan Cuza”, 2019
700109 – Iași, str. Pinului, nr. 1A, tel./fax: (0232) 314947
http:// www.editura.uaic.ro e-mail: editura@uaic.ro
Proceedings of the 5th Conference on Mathematical Foundations of Informatics

3-6 July 2019, Iași, Romania

Daniela Gîfu ● Bogdan Aman ● Adrian Iftene ● Diana Trandabăț
(Editors)
Table of Contents

Foreword ................................................................. 9

Invited Talks

Fuzzy Models in Deep Learning Networks - A Bridge between Symbolic and Connectionist Methods of AI. ....................... 13
   Alexei Averki

Several Definitions for Infinity in Finitely Supported Mathematics ................................................................. 15
   Andrei Alexandru and Gabriel Ciobanu

Applied Statistics in Information Security ............................ 31
   Emil Simion

Demystifying Deep Learning Breakthroughs for Computer Vision. 33
   Alexandru Lucian Ginsca

Algebra, Logic, and Cryptography

Boneh-Gentry-Hamburg’s Identity-based Encryption Schemes Revisited ................................................................. 45
   George Teșeleanu, Ferucio Laurențiu Țiplea, Sorin Iftene and Anca-Maria Nica

Four-dimensional Finite Non-commutative Associative Algebra and Its Application ................................................................. 59
   Alexandr Moldovyan, Dmitrii Moldovyan and Victor Shcherbacov

On Anonymization of Cocks’ Identity-based Encryption Scheme .... 75
   Anca-Maria Nica and Ferucio Laurențiu Țiplea

Attacks on Generalized Markovski Crypto-algorithm .................. 87
   Nadezhda Malyutina
On Weak Unpredictability of Legendre Sequences .......................... 103

Victor Pescaru

On Formalization Styles and SAD System ................................. 111

Alexander Lyaletski and Alexandre Lyaletsky

Approximations of Finitely Supported Sets ............................ 123

Andrei Alexandru and Gabriel Ciobanu

Compositions of Automata Defined on Finite Quasigroups ............ 137

Volodymyr G. Skobelev and Volodymyr V. Skobelev

On Some Groupoids with Bol-Moufang Type Identities ............... 149

Grigorii Horos, Victor Shcherbacov, Alexandr Tcachenco and Tatiana Yatsko

Maximal Slupecki Iterative Algebras of the 4-valued Diagonalizable Algebra and Related Algorithms ..................... 159

Andrei Rusu and Elena Rusu

Some Developments of Aggregate Theory ................................. 171

Ioachim Drugus and Volodymyr Skobelev

General Boolean Algebras, Boolean Rings and Extension Algebras ................................................................. 181

Eugene Kuznetsov and Elena Kuznetsova

Completeness of the Logic of Partial Quasiary Predicates with the Complement Composition .............................. 187

Mykola Nikitchenko, Oksana Shkilniak, Stepan Shkilniak and Tohrul Mamedov

Reliability of Erasure Coding Schemes for Distributed Storage Systems .............................................................. 203

Andrei Bogdan Baran, Adrian Iftene and Hugues Mercier
Mathematical Formalization of Local Sensitivity Analysis in Computational Models ........................................... 215
Julian Dimitrov

Computer Science in Applications

Recognition of Heterogeneous Documents: Problems and Challenges ................................................................. 231
Alexandru Colesnicov, Svetlana Cojocaru and Ludmila Malahov

Early Detection of Signs of Anorexia in Social Media .............. 247
Ciprian-Gabriel Cuşmuliuc, Lucia-Georgiana Coca and Adrian Iftene

Building a Diachronic Parallel Corpus for the Alignment of the Old Romanian Texts ........................................... 263
Tudor Bumbu

Iasi City Explorer ............................................................... 271
Camelia Miluţ, Adrian Iftene and Daniela Gifu

Pocket World AR – Let’s learn New Things in a Pleasant Way … 277
Roxana Pantea, Adrian Iftene and Daniela Gifu

Smart Museums with Augmented Reality ............................. 287
Andrei Porfireanu, Alexandru Ungurean, Andreea Dascălu,
Adrian Iftene and Daniela Gifu

Solar System Explorer ........................................................ 297
Diana Elena Guşă, Adrian Iftene and Daniela Gifu

Research Centered on Investigating the Ethical Attitudes Towards Information Technology ..................................... 309
Liliana Măţă and Ioana Boghian

Achievements – A Way to Engage Students in Educational Process ................................................................. 321
Olesea Caftanatov
AI Based Multilayered Approach for Management of Mass Casualty Situations ............................................. 331
    Constantin Gaindric, Svetlana Cojocaru, Stefan Pickl, Iulian Secrieru and Olga Popcova
Performing Cricothyroidotomy Incision Using Artificial .................. 337
    Catalin Crisan
User Recognition Based on Keystroke Dynamics................................. 347
    Constantin-Răzvan Cazuc, Adrian Iftene
Approaches in Formalization of a Distributed Transaction Compensation ......................................................... 355
    Ștefan Șaramet and Florin Olariu
Comparison between Modern JavaScript Frameworks Performance ................................................................. 367
    Ionuț Robert Iacob and Florin Olariu
Opportunities for Phone Account Recharging from Card Using the Possibilities Offered by Optical Character Recognition ......... 373
    Cristian Cosneanu and Inga Titchiev
SmartID - a PrivateSKY Based Platform for Medical Usage in Respect with GDPR .................................................. 379
    Mircea Hulea, Mihai Mart, Andrei Panu, Ștefan Romanescu, Ștefana Toader, Corneliu Munteanu and Andrei Scutelnicu
Foreword

At its inception, in 2015, the FOI Workshop (later MFOI Conference – Mathematical Foundations of Informatics) was an initiative of the Institute of Mathematics and Computer Science of the Academy of Sciences of Moldova, Taras Shevchenko National University of Kyiv, Ukraine, East Computers, SRL, in cooperation with Tiraspol State University and Information Society Development Institute.

The main aim of MFOI is to bring together at the same table mathematicians, computational linguists, computer scientists, PhD students and master students and all those that have a major interest in the study of the Mathematical Foundations of Informatics.

The organisers of the MFOI 2019 edition are: the Faculty of Computer Science of the “Alexandru Ioan Cuza” University of Iași, Romania; the “Vladimir Andrunachievici” Institute of Mathematics and Computer Science of Kishinev, Republic of Moldova; and Taras Shevchenko National University of Kyiv, Ukraine.

In 2019, the 5th MFOI Conference was brought in Iași, the beautiful city located in the North-Eastern Romania, in the heart of the province of Moldavia, a city full of students, and whose buildings and streets, stretching over seven hills (just like Rome), offer visitors memorable history lessons at their strolling through the town.

The volume includes 35 papers, covering topics in major areas of mathematics and informatics, which give also the names of the 3 chapters: (1) Invited Talks, (2) Algebra, Logic, and Cryptography, and (3) Computer Science in Applications.

Many people contributed to the success of this event, and we express our sincere thanks to all of them. The Scientific Committee made up of international experts in the area of Mathematics and
Computer Science, spent time carefully reviewing all the proposals submitted to MFOI-2019 to insure a qualitative improvement of the papers. The members of the Organizing Committee have enthusiastically assured the settings were appropriate for scientific networking.

July 2019

The editors
Invited Talks
Fuzzy models in deep learning networks - a bridge between symbolic and connectionist methods of AI

Alexei Averkin

This lesson introduces the terms and definitions of machine learning that are relevant to the context of extracting rules from classical and deep neural networks. It includes the problem of classification as a whole, as well as rule-based teaching methods and neural networks. Then, we will look at the current state of rule extraction from neural networks. Here we define the problem as well as the main approaches to its solution and present some of the existing rules extraction algorithms. The last part discusses specific problems when working with deep neural networks. At this stage, we also propose some algorithms that can successfully extract rules from these more complex neural networks.

Artificial Neural Networks (ANN) are widely known parallel computing models that exhibit excellent behavior in solving complex problems of artificial intelligence. However, many researchers refuse to use them due to their being a “black box”. This means that determining why a neural network makes a specific decision is a difficult task. This is a significant drawback, since it is difficult to trust the reliability of the network that solves real problems without the ability to make acceptable decisions. For example, this is the case in critical, in terms of safety, applications where hidden failure can lead to life-threatening actions or huge economic losses.

In addition, studying how neural networks extract, store and transform knowledge can be useful for future machine learning methods. For example, increasing the transparency of neural networks can help
detect the so-called hidden dependencies that are not present in the input data, but appear as a result of their integration into the neural network.

To overcome this lack of neural networks, researchers came up with the idea of extracting rules from neural networks, which can become a bridge between symbolic and connectionist models of knowledge representation in artificial intelligence.

Most authors focus on extracting the most understandable rules, and at the same time they should mimic the behavior of the neural network as precisely as possible, right up to an isomorphic representation of fuzzy rules in the form of a neuro-fuzzy system. Since 1992, since Chang’s doctoral thesis on neuro-fuzzy networks, much work has been done in this area, which ended with the creation of the direction of soft computing. Since then, many methods for extracting rules from neural networks have been developed and evaluated, and excellent results have been obtained for many approaches.

However, despite the fact that there are quite a few available algorithms, none of them has ever been explicitly tested in deep neural networks. In addition, most authors focus on networks with only a small number of hidden layers.

Only in the last few years pioneering work has appeared on the analysis of specific methods for extracting rules from deep neural networks and new algorithms are presented that are capable of performing this task.

Alexei Averkin

Federal Research Center “Computer Science and Control” of Russian Academy of Sciences
E-mail: averkin2003@inbox.ru
Several Definitions for Infinity in Finitely Supported Mathematics
Andrei Alexandru, Gabriel Ciobanu

Abstract
The theory of finitely supported algebraic structures represents a reformulation of Zermelo-Fraenkel set theory obtained by requiring that every set construction is finitely supported according to a certain action of a group of permutations of some basic elements named atoms. In this paper we introduce and compare different definitions for ‘infinite finitely supported set’. We present relevant examples of atomic sets that satisfy some forms of infinity, while do not satisfy other forms of infinity.

Keywords: finitely supported structures, various infinities, Dedekind infinite, Mostowski infinite, Tarski infinite.

1 Introduction
The approach regarding finitely supported algebraic structures, which is known in the literature under the name ‘Finitely Supported Mathematics’ (in pure set theoretical papers related to the foundations of mathematics) [1], or ‘nominal sets’ (when dealing with computer science applications regarding managing names in syntax) [5], represents an alternative framework for working with infinite structures, hierarchically constructed by involving an infinite family of basic elements (called atoms), by dealing only with finitely many entities that form their supports. Finitely supported sets are also related to the recent development of Fraenkel-Mostowski axiomatic set theory which represents an axiomatization of Fraenkel Basic Model for Zermelo-Fraenkel
set theory with atoms (originally presented to prove the independence of the axiom of choice and other axioms of set theory with atoms). Nominal sets are actually a Zermelo-Fraenkel (ZF) alternative to this non-standard set theory, since nominal sets are defined by involving group actions without being necessary to modify the ZF axioms of extensionality or foundation.

A nominal set is defined as a usual ZF set endowed with a group action of the group of (finitary) permutations over a certain fixed countable ZF set $A$ (also called the set of atoms by analogy with the Fraenkel and Mostowski models of set theory with atoms) formed by elements whose internal structure is irrelevant (i.e. by elements that can be checked only for equality), satisfying a certain finite support requirement. This requirement states that for any element $x$ in a nominal set there should exist a finite set $S \subseteq A$ such that any permutation of $A$ fixing $S$ pointwise also leaves the element $x$ invariant under the related group action. Nominal sets represent a categorical mathematical theory of names studying scope, binding, freshness and renaming in formal languages based upon symmetry. Inductively defined finitely supported sets (that are finitely supported elements in the powerset of a nominal set) involving the name-abstraction together with Cartesian product and disjoint union can encode syntax modulo renaming of bound variables. In this way, the standard theory of algebraic data types can be extended to include signatures involving binding operators. Particularly, there is an associated notion of structural recursion for defining syntax-manipulating functions and a notion of proof by structural induction. Generalizations of nominal sets are involved in the study of automata or programming languages over infinite alphabets; for this a relaxed notion of finiteness, called ‘orbit finiteness’, was defined and means ‘having a finite number of orbits under a certain group action’.

Finitely Supported Mathematics (shortly, FSM) is an alternative name for nominal algebraic structures, used in theoretical papers focused on the foundations of set theory (rather than on applications in computer science). In order to describe FSM as a theory of finitely
Several Definitions for Infinity in Finitely Supported Mathematics

supported algebraic structures (that are finitely supported sets together with finitely supported internal algebraic operation/laws), we use nominal sets (without the requirement that the set $A$ of atoms is countable) which by now on will be called invariant sets motivated by Tarski’s approach regarding logicality (i.e. a logical notion is defined by Tarski as one that is invariant under the permutations of the universe of discourse). FSM contains the family of ‘non-atomic’ (ordinary) ZF sets (which are proved to be trivial FSM sets, i.e. their elements are left unchanged under the effect of the canonical permutation action) and the family of ‘atomic’ sets with finite supports (hierarchically constructed from the empty set and the fixed ZF set $A$, meaning that they contain at least one atom somewhere in their structure). One main task now is to analyze whether a classical ZF result (obtained in the framework of non-atomic sets) can be adequately reformulated by replacing ‘non-atomic element/set’ with ‘atomic finitely supported element/set’ (according to the canonical actions of the group of permutations of $A$) in order to be valid also for atomic sets with finite supports. The (non-atomic) ZF results cannot be directly translated into the framework of atomic finitely supported sets, unless we are able to reprove their new formulations internally in FSM, i.e. by involving only finitely supported structures even in the intermediate steps of the proof. This is because the family of finitely supported sets is not closed under subset constructions (there are subsets of finitely supported sets failing to be themselves finitely supported, for example the simultaneously infinite and coinfinitely subsets of $A$), and we cannot use a construction outside FSM in order to prove something in FSM.

The meta-theoretical techniques for the translation of a result from non-atomic structures to atomic structures are based on a refinement of the finite support principle form [5] called “$S$-finite supports principle” claiming that for any finite set $S$ of atoms, anything that is definable in higher order logic from $S$-supported structures using $S$-supported constructions is also $S$-supported. The formal involvement of the $S$-finite support principles actually implies a hierarchical constructive method for defining the support of a structure by employing the supports of
the sub-structures of a related structure.

Formally, in order to describe FSM, we fix an infinite ZF set $A$ (however, despite classical set theory with atoms, we do not need to modify the ZF axioms of extensionality and foundation). A finite set (without any other considerations) is referred to a set of the form $\{x_1, \ldots, x_n\}$. An infinite set (without any other considerations) is referred to a set which is not finite. The elements of $A$ (atoms) are entities whose internal structure is considered to be irrelevant, and which are considered as basic for a higher-order construction. An invariant set $(X, \cdot)$ is defined as a ZF set $X$ equipped with an action $\cdot$ on $X$ of the group of all permutations of $A$, having the additional property that any element $x \in X$ is finitely supported. In a pair $(X, \cdot)$ formed by a ZF set $X$ and a group action $\cdot$ on $X$ of the group of all permutations of $A$, an arbitrary element $x \in X$ is finitely supported if and only if there exists a finite family $S \subseteq A$ such that any permutation of $A$ that fixes $S$ pointwise also leaves $x$ invariant under the group action $\cdot$. The least set of atoms supporting a finitely supported element $x$ is called the support of $x$ (denoted by $\text{supp}(x)$), and it is defined as the intersection of all finite sets supporting $x$. An empty supported element $x \in X$ is called equivariant. If there exists an action $\cdot$ of the group of permutations of $A$ on a set $X$, then there is a canonical action $\otimes$ of the group of permutations of $A$ on $\varphi(X) = \{Y \mid Y \subseteq X\}$, defined by $(\pi, Y) \mapsto \pi \otimes Y := \{\pi \cdot y \mid y \in Y\}$ for all permutations $\pi$ of $A$ and all $Y \subseteq X$. The set $\varphi_{fs}(X)$ represents the family of those finitely supported subsets of $X$ as elements in $\varphi(X)$ with respect to the action $\otimes$; $\varphi_{fs}(X)$ is an invariant set whenever $X$ is an invariant set. A subset of an invariant set having the property that all its elements are supported by the same set of atoms is called uniformly supported. The Cartesian product of two invariant sets $(X, \cdot)$ and $(Y, \diamond)$ is an invariant set with the canonical action $\otimes$ defined as $(\pi, (x, y)) \mapsto (\pi \cdot x, \pi \cdot y)$. Generally, an FSM set is a finitely supported subset of an invariant set. The class of FSM sets is closed under powerset and Cartesian constructions. The disjoint union of two invariant sets $(X, \cdot)$ and $(Y, \diamond)$, $X + Y = \{(0, x) \mid x \in X\} \cup \{(1, y) \mid y \in Y\}$, is an invariant set with the canonical action $\otimes$ defined by $\pi \otimes z = (0, \pi \cdot x)$.
Several Definitions for Infinity in Finitely Supported Mathematics

if \( z = (0, x) \) and \( \pi \star z = (1, \pi \diamond y) \) if \( z = (1, y) \). The sets \( \wp_{\text{fin}}(X) \) of all finite subsets of \( X \), \( \wp_{\text{cofin}}(X) \) of all subsets of \( X \) having finite complements, \( T_{\text{fin}}(X) \) of all finite injective tuples from \( X \), and \( T_{\delta_{\text{fin}}}^{\delta}(X) \) of all finite tuple from \( X \) are FSM sets whenever \( X \) is FSM set. Particularly, we have \( \wp_{\text{fs}}(A) = \wp_{\text{fin}}(A) \cup \wp_{\text{cofin}}(A) \). If \( X \in \wp_{\text{fin}}(A) \), then \( \text{supp}(X) = X \). If \( X \in \wp_{\text{cofin}}(A) \), then \( \text{supp}(X) = A \setminus X \). A relation (or, particularly, a function) between two invariant sets is finitely supported/equivariant if it is finitely supported/equivariant as a subset of the Cartesian product of those two invariant sets. Particularly, a function between two FSM sets \((X, \cdot)\) and \((Y, \diamond)\) is supported by a finite set \( S \) if and only if \( f(\pi \cdot x) = \pi \diamond f(x) \), \( \pi \cdot x \in X \) and \( \pi \diamond f(x) \in Y \) for all \( x \in X \) and all permutations \( \pi \) that fix \( S \) pointwise. The set of all functions from the FSM set \((X, \cdot)\) to the FSM set \((Y, \diamond)\) is an FSM set denoted by \( Y_{\text{fs}}^{X} \). Whenever an element \( a \in A \) appears in (the construction of) an FSM set \((X, \cdot)\) (particularly if \( X = A \)), the effect of a permutation of atoms \( \pi \) on \( a \), under \( \cdot \), is \( \pi(a) \). Therefore, canonical actions in FSM are defined from the action \( a \mapsto \pi(a) \) of the group of permutations of atoms on \( A \), using the rules of constructing actions for powersets, disjoint unions and Cartesian products.

2 Several Forms of Infinity in FSM

The equivalence of various definitions of infiniteness is provable in ZF under the consideration of the axiom of choice. Since in FSM the axiom of choice fails, our goal is to study various FSM forms of infinite and to provide several relations between them. We present the results without proofs. The related proofs (that are quite technical and exceeds the purpose of this conference paper) can be found in the technical paper [2].

The notion of infinity appear at a variety of levels in computer science. Actually computers are able to work over discrete models, meaning that the ‘infinity’ can be at most approximated, but not effectively computed. The newly developed theory of finitely supported sets allows the computational study of structures which are very large,
possibly infinite, but containing enough symmetries such that they can be concisely represented and manipulated. Our goal in this paper is to describe the notion of infinity within finitely supported structures. We emphasize the connections and differences between various ZF definitions of infinity internally in this new framework.

**Definition 2.1** Let $X$ be a finitely supported subset of an invariant set $Y$.

1. $X$ is called FSM classical infinite if $X$ does not correspond one-to-one and onto to a finite ordinal, i.e. if $X$ cannot be represented as $\{x_1, \ldots, x_n\}$ for some $n \in \mathbb{N}$. We simply call an FSM classical infinite set as infinite, and a set that is not FSM classical infinite as finite.

2. $X$ is FSM covering infinite if there is a finitely supported directed family $F$ of finitely supported subsets of $Y$ with the property that $X$ is contained in the union of the members of $F$, but there does not exist $Z \in F$ such that $X \subseteq Z$;

3. $X$ is called FSM Tarski I infinite (TI i) if there exists a finitely supported one-to-one mapping of $X$ onto $X \times X$.

4. $X$ is called FSM Tarski II infinite (TII i) if there exists a finitely supported family of finitely supported subsets of $X$, totally ordered by inclusion, having no maximal element.

5. $X$ is called FSM Tarski III infinite (TIII i) if $|X| = 2|X|$.

6. $X$ is called FSM Mostowski infinite (Mi) if there exists an infinite finitely supported totally ordered subset of $X$.

7. $X$ is called FSM Dedekind infinite (Di) if there exist a finitely supported one-to-one mapping of $X$ onto a finitely supported proper subset of $X$.

8. $X$ is FSM ascending infinite (Ai) if there is a finitely supported increasing countable chain of finitely supported sets $X_0 \subseteq X_1 \subseteq$
Several Definitions for Infinity in Finitely Supported Mathematics

\[ \cdots \subseteq X_n \subseteq \cdots \text{ with } X \subseteq \bigcup X_n, \text{ but there does not exist } n \in \mathbb{N} \text{ such that } X \subseteq X_n; \]

9. \( X \) is called FSM non-amorphous (N-am) if \( X \) contains two disjoint, infinite, finitely supported supported subsets.

Note that in the definition of FSM Tarski II infiniteness for a certain \( X \), the existence of a finitely supported family of finitely supported subsets of \( X \) is required, while in the definition of FSM ascending infiniteness for \( X \), the related family of finitely supported subsets of \( X \) has to be FSM countable (i.e. the mapping \( n \mapsto X_n \) should be finitely supported). It is immediate that if \( X \) is FSM ascending infinite, then it is also FSM Tarski II infinite.

**Theorem 2.2** Let \( X \) be a finitely supported subset of an invariant set \( Y \).

1. \( X \) is FSM classical infinite if and only if \( X \) is FSM covering infinite.

2. \( X \) is FSM classical infinite if and only if there exists a non-empty finitely supported family of finitely supported subsets of \( X \) having no maximal element under inclusion.

**Theorem 2.3** The following properties of FSM Dedekind infinite sets hold.

1. Let \( X \) be a finitely supported subset of an invariant set \( Y \). Then \( X \) is FSM Dedekind infinite if and only if there exists a finitely supported one-to-one mapping \( f : \mathbb{N} \rightarrow X \). As a consequence, an FSM superset of an FSM Dedekind infinite set is FSM Dedekind infinite, and an FSM subset of an FSM set that is not Dedekind infinite is also not FSM Dedekind infinite.

2. Let \( X \) be an infinite finitely supported subset of an invariant set \( Y \). Then the sets \( \wp_{fs}(\wp_{fin}(X)) \) and \( \wp_{fs}(T_{fin}(X)) \) are FSM Dedekind infinite.

21
3. Let $X$ be an infinite finitely supported subset of an invariant set $Y$. Then the set $\wp_{\text{fs}}(\wp_{\text{fs}}(X))$ is FSM Dedekind infinite.

4. Let $X$ be a finitely supported subset of an invariant set $Y$ such that $X$ does not contain an infinite subset $Z$ with the property that all the elements of $Z$ are supported by the same set of atoms. Then $X$ is not FSM Dedekind infinite.

5. Let $X$ be a finitely supported subset of an invariant set $Y$ such that $X$ does not contain an infinite subset $Z$ with the property that all the elements of $Z$ are supported by the same set of atoms. Then $\wp_{\text{fin}}(X)$ is not FSM Dedekind infinite.

6. Let $X$ and $Y$ be two finitely supported subsets of an invariant set $Z$. If neither $X$ nor $Y$ is FSM Dedekind infinite, then $X \times Y$ is not FSM Dedekind infinite.

7. Let $X$ and $Y$ be two finitely supported subsets of an invariant set $Z$. If neither $X$ nor $Y$ is FSM Dedekind infinite, then $X + Y$ is not FSM Dedekind infinite.

8. Let $X$ be a finitely supported subset of an invariant set $Y$. Then $\wp_{\text{fs}}(X)$ is FSM Dedekind infinite if and only if $X$ is FSM ascending infinite.

9. Let $X$ be a finitely supported subset of an invariant set $Y$. If $X$ is FSM Dedekind infinite, then $X$ is FSM ascending infinite. The reverse implication is not valid.

**Corollary 2.4** The following sets and all of their FSM classical infinite subsets are FSM classical infinite, but they are not FSM Dedekind infinite.

1. The invariant set $A$ of atoms.

2. The powerset $\wp_{\text{fs}}(A)$ of the set of atoms.

3. The set $T_{\text{fin}}(A)$ of all finite injective tuples of atoms.
4. The invariant set $A_{fs}^A$ of all finitely supported functions from $A$ to $A$.

5. The invariant set of all finitely supported functions $f : A \rightarrow A^n$, where $n \in \mathbb{N}$ and $A^n$ is the $n$-times Cartesian product of $A$.

6. The invariant set of all finitely supported functions $f : A \rightarrow T_{fin}(A)$.

7. The invariant set of all finitely supported functions $f : A \rightarrow \varphi_{fs}(A)$.

8. The sets $\varphi_{fin}(A)$, $\varphi_{cofin}(A)$, $\varphi_{fin}(\varphi_{fs}(A))$, or $\varphi_{fin}(A_{fs}^A)$.

9. Any construction of finite powersets of the following forms $\varphi_{fin}(\ldots \varphi_{fin}(A))$, $\varphi_{fin}(\ldots \varphi_{fin}(A_{fs}^A))$, or $\varphi_{fin}(\ldots \varphi_{fin}(\varphi_{fs}(A)))$.

10. Every finite Cartesian combination between the set $A$, $\varphi_{fin}(A)$, $\varphi_{cofin}(A)$, $\varphi_{fs}(A)$ and $A_{fs}^A$.

11. The disjoint unions $A + A_{fs}^A$, $A + \varphi_{fs}(A)$, $\varphi_{fs}(A) + A_{fs}^A$ and $A + \varphi_{fs}(A) + A_{fs}^A$ and all finite disjoint unions between $A$, $A_{fs}^A$ and $\varphi_{fs}(A)$.

**Corollary 2.5** The following sets and all of their supersets, their powersets and the families of their finite subsets, are both FSM classical infinite and FSM Dedekind infinite.

1. The invariant sets $\varphi_{fs}(\varphi_{fs}(A))$, $\varphi_{fs}(\varphi_{fin}(A))$ and $\mathbb{N}$.

2. The set of all finitely supported mappings from $X$ to $Y$, and the set of all finitely supported mappings from $Y$ to $X$, where $X$ is a finitely supported subset of an invariant set with at least two elements, and $Y$ is an FSM Dedekind infinite set.

3. The set of all finitely supported functions $f : \varphi_{fin}(Y) \rightarrow X$ and the set of all finitely supported functions $f : \varphi_{fs}(Y) \rightarrow X$, where $Y$ is an infinite finitely supported subset of an invariant
set, and $X$ is a finitely supported subset of an invariant set with at least two elements.

4. The set $T_{\text{fin}}^\delta(A) = \bigcup_{n \in \mathbb{N}} A^n$ of all finite tuples of atoms (not necessarily injective).

By definition, if a set $X$ is not FSM Dedekind infinite (which is equivalent with the non-existence of a finitely supported injection from $\mathbb{N}$ into $X$), then every finitely supported injection from $X$ into $X$ is also surjective. However, there may exist a finitely supported surjection from $X$ onto $X$ which is not injective. For example, let us consider $X = T_{\text{fin}}(A)$ the set of all finite injective tuples of atoms which is not FSM Dedekind infinite. We can define an equivariant surjective function $f : T_{\text{fin}}(A) \to T_{\text{fin}}(A)$ by

$$f(y) = \begin{cases} \emptyset, & \text{if } y \text{ is a tuple with exactly one element, or } y = \emptyset; \\ y', & \text{otherwise}; \end{cases}$$

where $y'$ is a new tuple of atoms formed by deleting the first element in the tuple $y$. Clearly, $g$ is not injective because every one-element tuple of atoms is mapped into the empty tuple. Despite of this, whenever $\varphi_{fs}(X)$ is not FSM Dedekind infinite we are able to prove the equivalence between injectivity and surjectivity for finitely supported self-mappings of $X$. This follows from Proposition 2.6, and from the fact that whenever $\varphi_{fs}(X)$ is not FSM Dedekind infinite we also have that $X$ is not FSM Dedekind infinite, and so the finitely supported injective self-mappings of $X$ should be surjective.

**Proposition 2.6** Let $X$ be a finitely supported subset of an invariant set. If $\varphi_{fs}(X)$ is not FSM Dedekind infinite, then each finitely supported surjective mapping $f : X \to X$ should be injective. The converse does not hold.

**Proposition 2.7** 1. Let $X$ be a finitely supported subset of an invariant set. If $\varphi_{\text{fin}}(X)$ is FSM Dedekind infinite, then $X$ should be FSM non-uniformly amorphous, meaning that $X$ should contain two disjoint, infinite, uniformly supported subsets.
Several Definitions for Infinity in Finitely Supported Mathematics

2. Let $X$ be a finitely supported subset of an invariant set. If $\mathcal{P}_{fs}(X)$ is FSM Dedekind infinite, then $X$ should be FSM non-amorphous, meaning that $X$ should contain two disjoint, infinite, finitely supported supported subsets. The reverse implication is not valid.

**Corollary 2.8** Let $X$ be a finitely supported amorphous subset of an invariant set (i.e. any finitely supported subset of $X$ is either finite or cofinite). Then each finitely supported surjective mapping $f : X \to X$ should be injective.

**Proposition 2.9**
1. Let $X$ be an FSM Dedekind infinite set. Then there exists a finitely supported surjection $j : X \to \mathbb{N}$. The reverse implication is not valid.

2. If $X$ is a finitely supported subset of an invariant set such that there exists a finitely supported surjection $j : X \to \mathbb{N}$, then $\mathcal{P}_{fs}(X)$ is FSM Dedekind infinite. The reverse implication is also valid.

The sets $A$ and $\mathcal{P}_{fin}(A)$ are both FSM classical infinite and none of them is FSM Dedekind infinite. We prove below that $A$ is not FSM ascending infinite, while $\mathcal{P}_{fin}(A)$ is FSM ascending infinite.

**Proposition 2.10**
- The set $A$ is not FSM ascending infinite.
- Let $X$ be a finitely supported subset of an invariant set $U$. If $X$ is FSM classical infinite, then the set $\mathcal{P}_{fin}(X)$ is FSM ascending infinite.

**Theorem 2.11** Let $X$ be a finitely supported subset of an invariant set $(Z, \cdot)$.

1. If $X$ is FSM Dedekind infinite, then $X$ is FSM Mostowski infinite.

2. If $X$ is FSM Mostowski infinite, then $X$ is FSM Tarski II infinite. The reverse implication is not valid.
**Proposition 2.12** Let $X$ be a finitely supported subset of an invariant set $(Z, \cdot)$. If $X$ is FSM Mostowski infinite, then $X$ is non-amorphous meaning that $X$ can be expressed as a disjoint union of two infinite finitely supported subsets. The reverse implication is not valid.

**Theorem 2.13** Let $X$ be a finitely supported subset of an invariant set $(Z, \cdot)$. If $X$ contains no infinite uniformly supported subset, then $X$ is not FSM Mostowski infinite.

Looking to the proof of Proposition 2.12 the following result follows directly.

**Corollary 2.14** Let $X$ be a finitely supported subset of an invariant set $(Z, \cdot)$. If $X$ is FSM Mostowski infinite, then $X$ is non-uniformly amorphous meaning that $X$ has two disjoint, infinite, uniformly supported subsets.

**Remark 2.15** In a permutation model of set theory with atoms, a set can be well-ordered if and only if there is a one-to-one mapping of the related set into the kernel of the model. Also it is noted that axiom of choice is valid in the kernel of the model [3]. Although FSM/nominal is somehow related to (has connections with) permutation models of set theory with atoms, it is independently developed over ZF without being necessary to relax the axioms of extensionality or foundation. FSM sets are ZF sets together with group actions, and such a theory makes sense over ZF without being necessary to require the validity of the axiom of choice on ZF sets. Thus, FSM is the entire ZF together with atomic sets with finite support (where the set of atoms is a fixed ZF formed by element whose internal structure is ignored and which are basic in the higher order construction). There may exist infinite ZF sets that do not contain infinite countable subsets, and as well there may exist infinite uniformly supported FSM sets (particularly such ZF sets) that do not contain infinite countable, uniformly supported subsets.

**Corollary 2.16** 1. The sets $A$, $A + A$ and $A \times A$ are FSM classical infinite, but there are not FSM Mostowski infinite, nor FSM Tarski II infinite.
2. None of the sets $\wp_{\text{fin}}(A)$, $\wp_{\text{cofin}}(A)$, $\wp_{fs}(A)$ and $\wp_{\text{fin}}(\wp_{fs}(A))$ is Mostowsky infinite in FSM.

3. None of the sets $A_{fs}^A$, $T_{\text{fin}}(A)_{fs}^A$ and $\wp_{fs}(A)_{fs}^A$ is FSM Mostowsky infinite.

**Corollary 2.17** Let $X$ be a finitely supported subset of an invariant set $Y$ such that $X$ does not contain an infinite uniformly supported subset. Then the set $\wp_{\text{fin}}(X)$ is not FSM Mostowsky infinite.

**Theorem 2.18** Let $X$ be a finitely supported subset of an invariant set $(Y, \cdot)$.

1. If $X$ is FSM Tarski I infinite, then $X$ is FSM Tarski III infinite. The converse does not hold. However if $X$ is FSM Tarski III infinite, then $\wp_{fs}(X)$ is FSM Tarski I infinite.

2. If $X$ is FSM Tarski III infinite, then $X$ is FSM Dedekind infinite. The converse does not hold. However if $X$ is FSM Dedekind infinite, then $\wp_{fs}(X)$ is FSM Tarski III infinite.

**Corollary 2.19** The following sets are FSM classical infinite, but they are not FSM Tarski I infinite, nor FSM Tarski III finite.

1. The invariant sets $A$ and $\wp_{fs}(A)$.

2. The set $\wp_{\text{fin}}(X)$ where $X$ is a finitely supported subset of an invariant set containing no infinite uniformly supported subset.

**Corollary 2.20** Let $X$ be an infinite finitely supported subset of an invariant set. Then $\wp_{fs}(\wp_{fs}(\wp_{fs}(X)))$ is FSM Tarski III infinite and consequently, $\wp_{fs}(\wp_{fs}(\wp_{fs}(\wp_{fs}(X))))$ is FSM Tarski I infinite.

In a future work we intend to prove an even stronger result claiming that $\wp_{fs}(\wp_{fs}(X))$ is FSM Tarski III infinite and, consequently, $\wp_{fs}(\wp_{fs}(\wp_{fs}(\wp_{fs}(X))))$ is FSM Tarski I infinite, whenever $X$ is an infinite finitely supported subset of an invariant set.
Corollary 2.21 The sets $A_f^{N}$ and $N_f^A$ are FSM Tarski I infinite, and so they are also Tarski III infinite.

Proposition 2.22 1. The set $\mathbb{N} \times A$ is FSM Tarski III infinite, but it is not FSM Tarski I infinite.

2. The set $A \cup \mathbb{N}$ is Dedekind infinite, but it is not FSM Tarski III infinite.

Theorem 2.23 Let $X$ be a finitely supported subset of an invariant set $(Y, \cdot)$. If $\varphi_{fs}(X)$ is FSM Tarski I infinite, then $\varphi_{fs}(X)$ is FSM Tarski III infinite. The converse does not hold.

Proposition 2.24 Let $X$ be a finitely supported subset of an invariant set $(Y, \cdot)$. If $X$ is FSM Tarski III infinite, then there exists a finitely supported bijection $g : \mathbb{N} \times X \rightarrow X$. The reverse implication is also valid.

3 Conclusion

The idea of presenting various approaches regarding ‘infinity’ belongs to Tarski whoformulates several definitions of infinite in [6]. The independence of these definitions was later proved in set theory with atoms in [4]. Such independence results can be transferred into classical ZF set theory by employing Jech-Sochor’s embedding theorem stating that permutation models of set theory with atoms can be embedded into symmetric models of ZF, and so a statement which holds in a given permutation model of set theory with atoms and whose validity depend only on a certain fragment of that model, also holds in some well-founded model of ZF. In this paper we reformulate the definitions of (in)finiteness from [6] internally into FSM, in terms of finitely supported structures. The related definitions for ‘FSM infinite’ are introduced in Section 2. We were able to establish strong comparison results between them and to present relevant examples of FSM sets that satisfy certain specific infiniteness properties. These comparison
results are proved internally in FSM, by employing only finitely supported constructions. Some of the results are obtained by using the classical translation technique from ZF into FSM involving the S-finite support principle, while many other properties (especially those revealing uniform supports) are specific to FSM. We also provide connections with FSM (uniformly) amorphous sets.

In the table below we present the forms of infinity satisfied by the classical infinite sets in Finitely Supported Mathematics:

<table>
<thead>
<tr>
<th>Set</th>
<th>T I</th>
<th>T III</th>
<th>D I</th>
<th>M I</th>
<th>A I</th>
<th>T II</th>
<th>N-am</th>
</tr>
</thead>
<tbody>
<tr>
<td>$A$</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>$A + A$</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>$A \times A$</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>$\emptyset_{fin}(A)$</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>$T_{fin}(A)$</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>$\emptyset_{fs}(A)$</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>$\emptyset_{fin}(\emptyset_{fs}(A))$</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>$A_{fs}^A$</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>$T_{fin}(A)^A_{fs}$</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>$\emptyset_{fs}(A)^A_{fs}$</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>$A \cup N$</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>$A \times N$</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>$\emptyset_{fs}(A \cup N)$</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>$\emptyset_{fs}(\emptyset_{fs}(A))$</td>
<td>?</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>$A_{fs}^N$ and $N_{fs}^A$</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
</tbody>
</table>

References


Applied Statistics in Information

Security

Emil Simion

Abstract

This talk is about the role of statistical tests in information security, especially the way of validating cryptographic primitives. We shall discuss the validation the FIPS 198 standard (Advanced Encryption Standard) through the NIST 800-22 statistical test suite. Also there are presented a series of open issues related to NIST 800-22 standard, such as the independence of the 15 statistical tests that are part of it, as well as the interpretation of the results from the second order error (the probability that a hypothesis will be accepted based on the sample false). At the same time, there are several examples where the deviation from randomness can lead to the exposure of cryptographic keys (secrets used for encryption or private use used for signing): Sony play station console signing key exposure and trapdoors introduced by NSA in dual elliptic curve deterministic random bit generator (Dual_EC_DRBG) used by NIST SP 800-90A.

Keywords: (pseudo)randomness, statistical testing, cryptographic evaluation, type I and type II errors (false positive and false negative results).

References


Emil Simion

Dept. of Mathematical Methods and Models, University POLITEHNICA of Bucharest, Romania
E-mail: emil.simion@upb.ro
Demystifying Deep Learning
Breakthroughs for Computer Vision

Alexandru Lucian Ginsca

Abstract

The recent rise of computer vision that has gained mainstream attention is mainly due to emergence of deep learning real fueled real world applications. While most of the theoretical concepts have been laid out in the past decades, the availability of large volumes of data, increasing computational power coupled with recent breakthroughs has led to deep learning being the main machine learning paradigm, especially for computer vision. In this extended abstract, we give an overview of the main mathematical concepts behind deep learning and outline the main talking points around the building blocks of deep learning applications and briefly introduce its main uses.

Keywords: deep learning, computer vision, convolutional neural networks

1 Introduction

For a long time the vision community has been striving for the quest of creating human-like intelligent vision systems. Recently the resurgence of neural networks [7] has first led to a revolution in computer vision, for example [2, 8], and then quickly provoked to other areas including reinforcement learning [12], speech recognition [6], and natural language processing [11]. Here, we focus on the building blocks of deep learning for computer vision and some of its applications.
2 Convolutional Neural Networks

The most successful deep learning model is probably convolutional neural networks (CNN) [4] where its hidden layers transform raw input images into highly discriminative features once the network is trained from millions of labeled images. Importantly, the generalization of convnet goes beyond the general assumption of training and test data. For example the AlexNet [8] trained on 1.2 million ImageNet images [15] has been used as an off-the-shelf feature detector for object detection [3], image segmentation [10], and image retrieval [1].

2.1 Building blocks

The simplest way to view convolutional neural networks is as a composition of functions and optimizers. Next, we briefly present the main building blocks that are used in CNNs but also in many of the other deep learning architectures.

Convolution. In purely mathematical terms, convolution is a function derived from two given functions by integration which expresses how the shape of one is modified by the other.

Pooling layer The pooling layer replaces the output of the network at certain locations by deriving a summary statistic of the nearby outputs. This helps in reducing the spatial size of the representation, which decreases the required amount of computation and weights. The pooling operation is processed on every slice of the representation individually. There are several pooling functions such as the average of the rectangular neighborhood, L2 norm of the rectangular neighborhood, and a weighted average based on the distance from the central pixel. However, the most popular process is max pooling, which reports the maximum output from the neighborhood.

Fully connected layer. Neurons in this layer have full connectivity with all neurons in the preceding and succeeding layer as seen in regular FCNN. This is why it can be computed as usual by a matrix multiplication followed by a bias effect. The most common approach used in pooling is max pooling.
**Sigmoid layer.** The input to the function is transformed into a value between 0.0 and 1.0. Inputs that are much larger than 1.0 are transformed to the value 1.0, similarly, values much smaller than 0.0 are snapped to 0.0. The shape of the function for all possible inputs is an S-shape from zero up through 0.5 to 1.0.

**Tanh.** The hyperbolic tangent function, or tanh for short, is a similar shaped nonlinear activation function that outputs values between -1.0 and 1.0.

**Rectified linear activation function.** It is a piecewise linear function that will output the input directly if is positive, otherwise, it will output zero. It has become the default activation function for many types of neural networks because a model that uses it is easier to train and often achieves better performance.

**Backpropagation.** It is an algorithm that uses gradient descent to minimize error in the predictions of Machine Learning models. This will calculate the gradient of the error function of any given error function and an artificial neural network while taking into account the different weights within that neural network.

**Loss functions.** Given an input and a target, the loss functions calculate the loss, such as the difference between output and target variable. The regressive loss functions are used in case of regressive problems, that is when the target variable is continuous. Most widely used regressive loss function is Mean Square Error. For classification problems, the output variable is usually a probability value $f(x)$, called the score for the input $x$. Generally, the magnitude of the score represents the confidence of our prediction. The target variable $y$, is a binary variable, 1 for true and -1 for false. On an example $(x, y)$, the margin is defined as $yf(x)$. The margin is a measure of how correct the prediction is. Most classification losses mainly aim to maximize the margin.

**Optimisation Algorithms.** They are used to update weights and biases i.e. the internal parameters of a model to reduce the error.
2.2 Object detection

One of the most common uses of CNNs is for object detection. The task of object detection implies drawing a bounding box around the objects found in an image and predicting their class. An object detector is a mapping from an image input space $X$ into the set of object sequences $Y = (R^4 \times C)^N$ with a finite number of elements: that is, such that for each $y \in Y$, there exists $m > 0$ such that for all $p \geq m$, $y_p = (0,0)$. $C = 0, 1, \ldots, n_c$ is a finite set where each element corresponds to an object class.

Finding a mapping that minimizes a given objective function is a task that can not be solved by the same architectures as image classifiers as the number of outputs changes depending on the input. A naive approach would rely on first extracting a set of regions of interest then process each part of the image bounded by a box and classify the presence of an object. Girshick et al. [5] proposed R-CNN, a model taking as input a set of 2,000 region proposals extracted with Selective Search, which extracts a representation of each region by passing them through a pre-trained CNN, and taking the output of a dense layer, then classifying each one using binary support vector machines. Furthermore, a linear regression is trained to predict more accurate bounding box coordinates.

Another popular object detector is You Only Look Once (YOLO) changes the paradigm of object detection by removing the need of region proposals. Instead, the image is divided into a grid of fixed dimensions and the network predicts two bounding boxes at each grid location with a confidence score and class probabilities at this point, independently from the bounding box and in only one forward pass.

2.3 Visual relations detection

More than being able to detect objects, visual relationship detection aims to describe the interactions between pairs of objects. Different from individual object learning tasks, the number of possible relationships are much larger, which makes it hard to explore only based on
the visual appearance of objects. Visual relationships capture a wide variety of interactions between pairs of objects in images, as the example from the model proposed by [9], as it can be seen in Figure 2.

![Illustration of visual relation extraction.](image)

Figure 1. Illustration of visual relation extraction.

## 3 Transfer Learning

Transfer learning refers to the process of using image descriptors from neural networks and using them in different tasks. Regardless on the network’s architecture (e.g. the number of convolutional of fully-connected layers, the size of of the sliding window used for the convolution operation, the image transformations or the use of regularization techniques, such as dropout), the global idea of this work is that the internal layers of the CNN can act as a generic extractor of mid-level image representations. The network can be pre-trained on one dataset (the source task, most common ImageNet) and then re-used on other target tasks, as illustrated in Figure 2.

CNN-based descriptors have shown major improvements over conventional ones in many non-trivial computer vision tasks, including image retrieval. Early works employ fully connected layers of CNNs as black-box extractors for global features that have high-level semantic meaning [16].

A different way to obtain image representations is through siamese networks. This architecture takes two images as input. Both the im-
Figure 2. Transferring parameters of a CNN. Adaptation of the framework depicted in [13]. First, the network is trained on the source task (ImageNet classification, top row) with a large amount of available labeled images. Pre-trained parameters of the internal layers of the network are then used as descriptors for images in different tasks.

Images are fed to a single Convolutional Neural Network (CNN). The last layer of the CNN produces a fixed size vector (embedding of the image). Since two images are fed, two embeddings are obtained. Then, the absolute distance between the vectors is calculated. The values pass through a sigmoid function and a similarity score is produced. Another difference with a CNN used for classification is the employment of a contrastive loss. We next briefly describe the use of a contrastive loss. Let $a$ be an anchor image, $p$ be a positive image that is relevant to the anchor, $n$ be a negative irrelevant image, $f(\cdot)$ be the function transforming an image to a new vector space, the classical contrastive loss is defined as:

$$L(a, p, n) = \max(0, m + d(f(a), f(p))^2 - d(f(a), f(p))^2)$$  \hspace{1cm} (1)$$

where $d(\cdot, \cdot)$ is the distance between 2 vectors in the embedding space and $m$ is a scalar controlling the margin. By minimizing 1, we want to maximizing the distance in the embedding space between a dissimilar
pair while minimizing that between a similar pair. We want the dis-
tance between a given image and any relevant images to be smaller than
that with respect to any irrelevant images. Ideally, the difference be-
tween 2 distances should be greater than the margin, in which case the
loss is 0. Siamese networks are commonly used in person identification
or image search tasks.

4 Generative Adversarial Networks

Generative Adversarial Networks (GANs) are composed of two neural
networks, each one trying to outcompete the other. The discriminator
tries to figure out whether a given image is real or synthetically gener-
ated by the other neural network. The generator attempts to output
images that are indistinguishable from real ones in an attempt to fool
the discriminator. The hope is that the generator learns enough to cre-
ate realistic images that can fool humans. Besides CNNs, GANs [14]

Figure 3. Illustration of GAN capabilities for portrait generation over
the years.

are represent the architecture that found itself the most in the public’s
eye in recent years. One of the uses for GANs is the generation of fake
life looking human portraits. In Figure 3, we illustrate the progress of
GANs between the years 2014 and 2018.
5 Conclusion

We presented in this extended abstract a brief overview of the building blocks of deep learning architectures for computer vision followed by some of the most high impact applications and breakthroughs.

References


Algebra, Logic, and Cryptography
Boneh-Gentry-Hamburg’s Identity-based Encryption Scheme Revisited

George Teșeleanu, Ferucio Laurențiu Țiplea, Sorin Iftene, Anca-Maria Nica

Abstract

BasicIBE and AnonIBE are two space-efficient identity-based encryption (IBE) schemes based on quadratic residues, proposed by Boneh, Gentry, and Hamburg, and closely related to Cocks’ IBE scheme. BasicIBE is secure in the random oracle model under the quadratic residuosity assumption, while AnonIBE is secure in the standard model under the interactive quadratic residuosity assumption. In this paper we revise the BasicIBE scheme and we show that if the requirements for the deterministic algorithms used to output encryption and decryption polynomials are slightly changed, then the scheme’s security margin can be slightly improved.

Keywords: identity-based encryption, quadratic residue, indistinguishability

1 Introduction

Identity-based cryptography was proposed in 1984 by Adi Shamir [5] who formulated its basic principles. The first identity-based encryption (IBE) scheme was proposed by Boneh and Franklin [2] and is based on bilinear maps. Shortly, Cocks [4] proposed another IBE scheme based on the standard quadratic residuosity problem modulo an RSA composite $n$. Cocks’ scheme encrypts a bit by two integers modulo $n$ such that the bit is recovered as the Jacobi symbol of one of these two integers together
with the private key. Although the scheme is very elegant and quite fast, its main disadvantage is ciphertext expansion: a bit of message requires $2 \log n$ bits of ciphertext. In [3], Boneh, Gentry, and Hamburg proposed two abstract IBE schemes with short ciphertexts that are related to Cocks’ scheme. The first scheme, named BasicIBE, is IND-ID-CPA\(^1\) secure in the random oracle model (ROM) under the quadratic residuosity assumption, while the second one, named AnonIBE, is ANON-IND-ID-CPA\(^1\) secure in the standard model under the interactive quadratic residuosity assumption\(^1\). Both security results are obtained by providing upper bounds on the advantage of an efficient adversary against the corresponding IBE scheme.

In order to provide a tighter upper bound for the BasicIBE scheme we slightly change the requirements for the deterministic algorithms $Q$ used to output encryption and decryption polynomials. The concrete instantiation of $Q$ provided in [3] satisfies the new set of restrictions. Thus, without changing the instantiation of BasicIBE we obtain a marginally better security margin.

Structure of the paper. We introduce notations and definitions used throughout the paper in Section 2. In Section 3 we describe the BasicIBE scheme and we provide the security margin proved in [3]. We reassess BasicIBE’s security proof in Section 4. We conclude in Section 5.

2 Preliminaries

Notations. Throughout the paper, $\lambda$ will denote a security parameter. The action of selecting a random element $x$ from a sample space $X$ is denoted by $x \leftarrow_{\$} X$. We denote by $x \leftarrow y$ the assignment of value $y$ to variable $x$. The probability that event $E$ happens is denoted by $Pr[E]$. The Jacobi symbol of an integer $a$ modulo an integer $n$ is denoted by $(a \mid n)$. $J_n$ stands for the set of integers in $\mathbb{Z}_n^*$ whose Jacobi symbol is 1, $QR_n$\(^1\). We refer the reader to Section 2 for a definition of the concept.
denotes the set of quadratic residues in \( \mathbb{Z}_n^* \) and \( SQRT_n(a) \) is the set of square roots of \( a \) modulo \( n \). \( \mathbb{Z}_n[x] \) is the ring of polynomials over \( \mathbb{Z}_n \). We denote by PPT algorithm a probabilistic polynomial-time algorithm. By RSAGen(\( \lambda \)), we understand a PPT algorithm that generates two equal size primes \( p \) and \( q \) larger then \( 2^\lambda \).

### 2.1 Security Assumptions

**Definition 1** (Pseudorandom Function - prf). A function \( F : \{0,1\}^n \times \{0,1\}^\lambda \rightarrow \{0,1\}^m \) is a prf if:

- Given a key \( K \in \{0,1\}^\lambda \) and an input \( X \in \{0,1\}^n \) there is an efficient algorithm to compute \( F_K(X) = F(X,K) \).

- For any algorithm \( A \), the prf-advantage of \( A \), defined as

\[
\text{PRFAdv}_{A,F}(\lambda) = \left| Pr[A^{F_K(\cdot)} = 1|K \leftarrow \{0,1\}^\lambda] - Pr[A^{F(\cdot)} = 1|F \leftarrow \mathcal{F}] \right|,
\]

is negligible for any PPT algorithm \( A \), where \( \mathcal{F} = \{F : \{0,1\}^n \rightarrow \{0,1\}^m\} \).

**Definition 2** (Quadratic Residuosity - QR). Let \( n = pq \), where \( (p,q) \leftarrow \text{RSAGen}(\lambda) \) and let \( A \) be a PPT algorithm which returns 1 on input \( (x,n) \) if \( x \) is a quadratic residue modulo \( n \). We define the advantage

\[
\text{QRAdv}_{A,\text{RSAGen}}(\lambda) = \left| Pr[A(x,n) = 1|x \leftarrow QR_n] - Pr[A(x,n) = 1|x \leftarrow J_n \setminus QR_n] \right|
\]

The *Quadratic Residuosity* assumption states that for any efficient PPT algorithm \( A \) the advantage \( \text{QRAdv}_{A,\text{RSAGen}}(\lambda) \) is negligible.
2.2 Identity-based encryption

An IBE scheme consists of four PPT algorithms: \textit{Setup}, \textit{Extract}, \textit{Encrypt}, and \textit{Decrypt}. The first one takes as input a security parameter and outputs the system public parameters together with a master key. The \textit{Extract} algorithm takes as input an identity \textit{ID} together with the public parameters and the master key and outputs a private key associated to \textit{ID}. The \textit{Encrypt} algorithm, starting with a message \textit{m}, an identity \textit{ID}, and the public parameters, encrypts \textit{m} into some ciphertext \textit{c} (the encryption key is \textit{ID} or some binary string derived from \textit{ID}). The last algorithm decrypts \textit{c} into \textit{m} by using the private key associated to \textit{ID}.

\textbf{Definition 3} (Anonymity and Indistinguishability under Selective Identity and Chosen Plaintext Attacks - ANON-IND-ID-CPA). The ANON-IND-ID-CPA security of an IBE scheme \textit{S} is formulated by means of the following game between a challenger \textit{C} and an adversary \textit{A}:

\textit{Setup}(\lambda): The challenger \textit{C} generates the public parameters \textit{PP} and sends them to adversary \textit{A}, while keeping the master key \textit{msk} to himself.

\textit{Queries}: The adversary issues a finite number of adaptive queries. A query can be one of the following types:

- Private key query. When \textit{A} requests a query for an identity, the challenger runs the \textit{Extract} algorithm and returns the resulting private key to \textit{A}.

- Encryption query. Adversary \textit{A} can issue only one query of this type. He sends \textit{C} two pairs \((\textit{ID}_0, \textit{m}_0)\) and \((\textit{ID}_1, \textit{m}_1)\) consisting of two equal length plaintexts \textit{m}_0 and \textit{m}_1 and two identities \textit{ID}_0 and \textit{ID}_1. The challenger flips a coin \(b \in \{0, 1\}\) and encrypts \textit{m}_b using \textit{ID}_b. The resulting ciphertext \textit{c} is sent to the adversary. The following restrictions are in place: private key queries for \textit{ID}_0 and \textit{ID}_1 must never be issued.
Guess: In this phase, the adversary outputs a guess $b' \in \{0, 1\}$. He wins the game, if $b' = b$.

The advantage of an adversary $A$ attacking an IBE scheme is defined as

$$\text{IBEAdv}_{A,S}(\lambda) = \left| \Pr[b = b'] - \frac{1}{2} \right|$$

where the probability is computed over the random bits used by $C$ and $A$. An IBE scheme is ANON-IND-ID-CPA secure, if for any PPT adversary $A$ the advantage $\text{IBEAdv}_{A,S}(\lambda)$ is negligible. If we consider $ID_0 = ID_1$ in the above game, we obtain the concept of IND-ID-CPA security.

3 Boneh-Gentry-Hamburg’s BasicIBE Scheme

Let $n$ be an RSA composite and $r$ the private key of the recipient. Cocks’ IBE scheme [4] encrypts a bit $m \in \{\pm 1\}$ by two integers $c_1$ and $c_2$ such that either the Jacobi symbol of $(c_1 + 2r)$ or the Jacobi symbol of $(c_2 + 2r)$ modulo $n$ is $m$. The scheme is IND-ID-CPA secure in ROM under the QR assumption.

Despite its elegance, Cocks’ scheme produces large ciphertext: $2 \log n$ bits are used to encrypt just one bit. Moreover, it is not anonymous [1,6]. In 2007, Boneh, Gentry and Hamburg proposed two space efficient IBE schemes related to Cocks’ scheme, whose security is also based on the QR problem [3]. Additionally, one of the schemes is anonymous. Below we describe the BasicIBE scheme.

Setup$(\lambda)$: Let $n = pq$, where $(p, q) \leftarrow \text{RSAGen}(\lambda)$. Generate $u \in J_n \setminus QR_n$, and choose a hash function $h : \{0, 1\}^* \times \{1, \ldots, \ell\} \to J_n$, for some integer $\ell \geq 1$. Choose a deterministic algorithm $Q$, that takes as input $n$ and any $R, S \in \mathbb{Z}_n^*$ and outputs two polynomials $f, g \in \mathbb{Z}_n[x]$. Output the public parameters $PP = (n, u, h, \ell, Q)$. The master key is $msk = (p, q, K)$, where $K$ is a random key for a pseudorandom function $F_K : \{0, 1\}^* \times \{1, \ldots, \ell\} \to \{0, 1, 2, 3\}$. 

49
Extract\((msk, ID, \ell)\): For each \(j \in \{1, \ldots, \ell\}\), let \(R_j = h(ID, j)\) and \(i_j = F_K(ID, j)\). If \(r_0, r_1, r_2, r_3\) is a fixed total ordering of the square roots of \(R_j\) or \(uR_j\) (depending on which of them is a quadratic residue), then the private key is \(r = (r_{i_1}, \ldots, r_{i_\ell})\).

Encrypt\((PP, ID, m)\): Let \(m = m_1 \ldots m_\ell \in \{\pm 1\}^\ell\) be an \(\ell\)-bit sequence. Generate at random \(s \in \mathbb{Z}_n^*\) and set \(S = s^2 \mod n\). For each \(j \in \{1, \ldots, \ell\}\), let \(R_j = h(ID, j)\). Use algorithm \(Q\) to compute the polynomials \((f_j, g_j) \leftarrow Q(n, R_j, S)\) and \((\bar{f}_j, \bar{g}_j) \leftarrow Q(n, uR_j, S)\). Set \(c_j \leftarrow m_j \cdot \left(\frac{g_j(s)}{n}\right)\) and \(\bar{c}_j \leftarrow m_j \cdot \left(\frac{\bar{g}_j(s)}{n}\right)\). Output \((c, \bar{c}, S)\), where \(c = c_1 \cdots c_\ell\) and \(\bar{c} = \bar{c}_1 \cdots \bar{c}_\ell\).

Decrypt\((c, \bar{c}, S, r, PP)\): For each \(j \in \{1, \ldots, \ell\}\), let \(R_j = h(ID, j)\). If \(R_j = r_{j_2}^2\), then \((f'_j, g'_j) = Q(n, R_j, S)\), else \((f'_j, g'_j) = Q(n, uR_j, S)\). Compute \(m_j = c_j \cdot \left(\frac{f'(r_{i_j})}{n}\right)\). Output the message \(m = m_1 \cdots m_\ell\).

Before stating the security result from [3], we first need to introduce some properties that need to be satisfied by \(Q\).

**Definition 4** (IBE Compatible). Let \(n\) be a positive integer and \(R, S \in \mathbb{Z}_n^*\). Let \(Q\) be a deterministic algorithm that takes as input \(n, R, S\) and outputs two polynomials \(f, g \in \mathbb{Z}_n[x]\). Algorithm \(Q\) is *IBE Compatible* if the following conditions are satisfied

1. If \(R, S \in QR_n\), then \(f(r)g(s) \in QR_n\), where \(r \in \text{SQRT}_n(R)\) and \(s \in \text{SQRT}_n(S)\).

2. If \(R \in QR_n\) and \(S \in J_n \setminus QR_n\), then \(\left(\frac{f(r)}{n}\right)\) is uniformly distributed in \(\{\pm 1\}\), where \(r \overset{\$}{\leftarrow} \text{SQRT}_n(R)\).

Definition 4 is introduced in [3] in another form. If \(R \in QR_n\), the authors require that \(f(r)f(-r)S \in QR_n\) for all \(r \in \text{SQRT}_n(R)\). In [3, Lemma 3.3] it is proven that if \(r \overset{\$}{\leftarrow} \text{SQRT}_n(R)\) and \(S \in J_n \setminus QR_n\), then \(f(r)f(-r)S \in QR_n\) implies Definition 4, Condition 2.
In the security proof of BasicIBE, Boneh, Gentry and Hamburg [3] use that \( \frac{f(r)}{n} \) is uniformly distributed in \( \{\pm 1\} \) and not that \( f(r)f(-r)S \in QR_n \). Thus, Definition 4, Condition 2 captures the exact security requirement for \( Q \).

If \( Q \) is IBE compatible, then Definition 4, Condition 1 guarantees the soundness of decryption. As with respect to security, the following result is proven in [3].

**Theorem 1.** If the \( QR \) assumption holds and \( Q \) is IBE compatible, then the BasicIBE scheme is IND-ID-CPA secure in ROM. Formally, for any efficient PPT adversary \( A \) there exist efficient PPT algorithms \( B_1 \) and \( B_2 \) such that

\[
\text{IBEAdv}_{A,\text{BasicIBE}}(\lambda) \leq \text{PRFAdv}_{B_1,F}(\lambda) + 2\text{QRAdv}_{B_2,\text{RSA}\text{Gen}}(\lambda).
\]

A few words are in place about Theorem 1. The proof of this theorem as it is in [3], exploits the fact that \( h \) outputs truly random elements from \( J_n \) and replaces \( h \) with \( H(ID,j) = ua_jv_j^2 \), where \( a_j \xleftarrow{} \{0,1\} \), \( v_j \xleftarrow{} Z_n^* \). The initial IND-ID-CPA game is successively changed into another game where the challenge ciphertext is created by decrypting the message (that is, by encrypting it by \( f's \) instead of \( g's \)). In order to have \( R_j, uR_j \in QR_n \) and \( S \in J_n \setminus QR_n \), the \( QR \) assumption is used two times, which gives rise to the factor \( 2\text{QRAdv}_{B_2,\text{RSA}\text{Gen}}(\lambda) \). Moreover, to ensure that \( r_j \) is a random square root of \( R_j \) or \( uR_j \), \( F_K \) is replaced by a truly random function, and this gives rise to the factor \( \text{PRFAdv}_{B_1,F}(\lambda) \).

**Remark 1.** We emphasize that the BasicIBE scheme is an abstract IBE scheme because no concrete algorithm \( Q \) is presented. In [3], the method proposed to construct the polynomials \( f \) and \( g \) is based on the congruence given by

\[
Rx^2 + Sy^2 \equiv 1 \mod n,
\]

where \( n \) is an integer and \( R,S \in Z_n^* \).

Any solution \((x_0,y_0)\) to the above equation gives rise to two polynomials.
George Teşeleanu et al.

\[
f(r) = x_0r + 1 \mod n \quad g(s) = 2(y_0s + 1) \mod n
\]

that satisfy the conditions from Definition 4.

If we instantiate \( Q \) as in Remark 1, then the encryptor must find solutions to \( 2\ell \) congruences, while the decryptor must find solutions to \( \ell \) congruences. Boneh, Gentry, and Hamburg [3] have proposed the following Combining Lemma in order to reduce the number of congruences to be solved by the encryptor.

**Lemma 1.** If \((x_1, y_1)\) is a solution to the congruence \( R_1x^2 + Sy^2 \equiv 1 \mod n \) and \((x_2, y_2)\) is a solution to the congruence \( R_2x^2 + Sy^2 \equiv 1 \mod n \), then \((x_{1,2}, y_{1,2})\) is a solution to the congruence \( R_1R_2x^2 + Sy^2 \equiv 1 \mod n \), where

\[
x_{1,2} = \frac{x_1x_2}{Sy_1y_2 + 1} \mod n \quad \text{and} \quad y_{1,2} = \frac{y_1 + y_2}{Sy_1y_2 + 1} \mod n,
\]

provided that \((Sy_1y_2 + 1, n) = 1\).

Using this result, the encryptor first finds solutions to \( ux^2 + Sy^2 \equiv 1 \mod n \) and \( R_jx^2 + Sy^2 \equiv 1 \mod n \), for all \( 1 \leq j \leq \ell \), and then combines them to obtain solutions to \( uR_jx^2 + Sy^2 \equiv 1 \mod n \), for all \( 1 \leq j \leq \ell \). Therefore, the encryptor needs to find solutions to \( \ell + 1 \) congruences, instead of \( 2\ell \). It is to be remarked that, the proposed method does not affect the security of the scheme.

4 A New Security Analysis for BasicIBE

In [3], the authors only require \( Q \) to be IBE compatible, but the algorithm proposed by them satisfies one more condition, that is captured in the following definition. It is easy to see that \( g \) satisfies the Condition 3, since \( Rx^2 + Sy^2 = 1 \) is symmetric.

**Definition 5** (Extended IBE Compatible). Let \( n \) be a positive integer and \( R, S \in \mathbb{Z}_n^* \). Let \( Q \) be a deterministic algorithm that takes as
input $n, R, S$ and outputs two polynomials $f, g \in \mathbb{Z}_n[x]$. Algorithm $Q$ is *Extended IBE Compatible* if it is *IBE compatible* and the following condition is satisfied

3. If $R \in J_n \setminus QR_n$ and $S \in QR_n$, then $g(s)/n$ is uniformly distributed in $\{\pm 1\}$, where $s \leftarrow \text{SQRT}_n(S)$.

Using the extra property satisfied by $Q^2$, we slightly modify the security proof of Theorem 1 and we obtain a tighter upper bound.

**Theorem 2.** *If the QR assumption holds and $Q$ is extended IBE compatible, then the BasicIBE scheme is IND-ID-CPA secure in ROM. Formally, for any efficient PPT adversary $A$ there exist efficient PPT algorithms $B_1$ and $B_2$ such that*

$$\text{IBEAdv}_{A,\text{BasicIBE}}(\lambda) \leq \text{PRFAdv}_{B_1,F}(\lambda) + \text{QRAdv}_{B_2,\text{RSAGen}}(\lambda).$$

**Proof.** Let $A$ be an IND-ID-CPA adversary for BasicIBE. We prove that his advantage is negligible. We present the proof as a sequence of games. Let $W_i$ be the event that $A$ wins game $i$.

**Game 0.** The first game is identical to the IND-ID-CPA game$^3$. Thus, we have

$$|P[W_0] - 1/2| = \text{IBEAdv}_{A,\text{BasicIBE}}(\lambda) \quad (1)$$

**Game 1.** In this game, $F_K$ is replaced by a truly random function $f : \{0, 1\}^* \times \{1, \ldots, \ell\} \to \{0, 1, 2, 3\}$. Adversary $A$ will not notice the difference, since $F_K$ is a pseudorandom function. Formally, there exists an algorithm $B_1$ such that

$$|Pr[W_0] - Pr[W_1]| = \text{PRFAdv}_{B_1,F}(\lambda). \quad (2)$$

**Game 2.** We slightly modify how the challenger answers hash queries. Thus, for a query $h(ID, j)$ it first chooses $a_j \leftarrow \{0, 1\}$ and $v_j \leftarrow \mathbb{Z}_{n^*}$.

$^2$captured in Definition 5

$^3$as in Definition 3
Then, it outputs \( h(ID, j) = u^a_j v_j^2 \). It is easy to see that the challenger implements a random function.

Let \( R_j = h(ID, j) \) for some \((ID, j)\). The challenger also has to answer private key extraction queries. Thus, when an extraction query for \( ID \) is received, the challenger answers with \( R_j^{1/2} = v_j \) if \( a_j = 0 \) or \((uR_j)^{1/2} = uv_j \) if \( a_j = 1 \), where \( 1 \leq j \leq \ell \). Since \( a_j \) and \( v_j \) are random elements, the challenger outputs a random square root of either \( R_j \) or \( uR_j \). Thus, Game 1 and Game 2 are identical from \( A \)'s point of view. Note that in this case the challenger can answer private key queries without knowing the factorization of \( n \). Formally, we have

\[
Pr[W_1] = Pr[W_2]. \tag{3}
\]

**Game 3.** By maintaining a list with all the hash queries, the challenger can decide if \( R_j \in QR_n \) or \( uR_j \in QR_n \). We change the encryption algorithm as follows

- If \( R_j \in QR_n \), then \( c_j \leftarrow m_j \cdot \left( f_j(r_{ij}) / n \right) \) and
  \[\bar{c}_j \leftarrow \{\pm 1\};\]
- Else \( c_j \leftarrow \{\pm 1\} \) and \( \bar{c}_j \leftarrow m_j \cdot \left( \bar{f}_j(r_{ij}) / n \right) \).

We will show that the ciphertext has the same distribution as in Game 2. First, recall that \( S \in QR_n \).

If \( R_j \in QR_n \), then according to Definition 4, Condition 1 \( c_j \) has the same value as in Game 2. Since \( uR_j \in J_n \setminus QR_n \), Definition 5, Condition 3 assures us that \( \left( \bar{g}_j(s) / n \right) \) is uniformly distributed in \( \{\pm 1\} \). Thus \( \bar{c}_j \) chosen in this game has the same distribution as in Game 2.

A similar discussion for the case \( R_j \in J_n \setminus QR_n \) shows that \( c_j \) and \( \bar{c}_j \) have the same distribution as in Game 2.

Since these are the only changes between Game 2 and Game 3, \( A \) will not notice the difference assuming \( Q \) is extended IBE compatible. Formally, this means that

\[
Pr[W_2] = Pr[W_3]. \tag{4}
\]
**Game 4.** In this game the challenger chooses $S \in J_n \setminus QR_n$. Since this is the only change between Game 3 and Game 4, $A$ will not notice the difference assuming the QR assumption holds. Formally, this means that there exists an algorithm $B_2$ such that

$$|Pr[W_3] - Pr[W_4]| = QRAdv_{B_2,RSAGen}(\lambda). \tag{5}$$

**Game 5.** To make the challenge ciphertext independent of the challenge bit $b$, we slightly change Game 4. Thus, the challenger chooses $c_j \overset{\$}{\leftarrow} \{\pm 1\}$, for $j \in \{1, \ldots, \ell\}$.

We will show that Game 4 and Game 5 are identical. To do that, we prove that the ciphertext distribution remains the same as in Game 4. Due to the change made in Game 4, we have that $S \in J_n \setminus QR_n$. If $R_j \in QR_n$, then according to Definition 4, Condition 2 $\left(\frac{f_j(r_{ij})}{n}\right)$ is uniformly distributed in $\{\pm 1\}$. Thus, $c_j$ has the same distribution in both games. The case $uR_j \in QR_n$ is treated in a similar way.

Since these are the only changes between Game 4 and Game 5, $A$ will not notice the difference assuming $Q$ is extended IBE compatible. Formally, this means that

$$Pr[W_4] = Pr[W_5]. \tag{6}$$

In this game, $c$ and $\bar{c}$ are independent of the challenge bit. Thus, we have

$$Pr[W_5] = 1/2. \tag{7}$$

Finally, the statement is proven by combining the equalities $(1)-(7)$. \hfill \Box

**Remark 2.** By tweaking the proof of Theorem 2, we obtain the same upper bound for the ANON-IND-ID-CPA security$^{4}$ of the AnonIBE scheme. Note that the deterministic algorithm used by the AnonIBE scheme also satisfies the extra property stated in Definition 5.

$^{4}$in ROM
5 Conclusions

Boneh, Gentry, and Hamburg have proposed in [3] two IBE schemes related to Cocks’ IBE scheme, called BasicIBE and AnonIBE. Compared to BasicIBE and Cocks’ IBE scheme, AnonIBE also provides anonymity of identity. These two schemes are more space efficient than Cocks’ IBE scheme, but the concrete instantiations are less time efficient.

In this paper we have revisited the BasicIBE abstract scheme, by slightly modifying the requirements for the deterministic algorithm. These requirements are already fulfilled by the concrete instantiation proposed in [3] (however, the authors of [3] did not use them). By using a different proof approach, we managed to obtain a tighter security margin for BasicIBE.

References


George Teșeleanu, Ferucio Laurențiu Țiplea, Sorin Iftene, Anca-Maria Nica
Department of Computer Science, “Al.I.Cuza” University of Iași
Email: {george.teseleanu,ferucio.tiplea,siftene,anca.nica}@info.uaic.ro
Four-dimensional Finite Non-commutative Associative Algebra and Its Application

Alexandr Moldovyan, Dmitrii Moldovyan, Victor Shcherbacov

Abstract

The paper considers the structure of a 4-dimensional finite non-commutative associative algebra defined over the ground finite field $GF(p)$, which represent significant interest as potential carrier of the hidden discrete logarithm problem and post-quantum public-key cryptoschemes. The investigated algebra has been used as algebraic carrier of two post-quantum public-key cryptoschemes.

Keywords: post-quantum cryptography, hidden logarithm problem, finite associative algebra, non-commutative algebra, left-sided unit, global unit.

MSC: 94A60, 16Z05, 14G50, 11T71, 16S50

1 Introduction

Development of the practical post-quantum public-key cryptoschemes represents a current challenge in the area of applied and theoretic cryptography [1, 2]. Post-quantum cryptographic algorithms and protocols should be based on some computational problems that are different from the factorization and discrete logarithm problems (DLP) that can be solved using polynomial algorithms developed for hypothetic quantum computer [3, 4].

For the development of the post-quantum public key-agreement schemes the hidden discrete logarithm problem (HDLP) was proposed
earlier as the base cryptographic primitive \cite{5, 6}. Recently \cite{7, 8} several new forms of the HDLP were introduced as the base primitive of the digital signature schemes. Different types of the finite non-commutative associative algebras (FNAAs) are used as the algebraic carrier of the post-quantum cryptoschemes based on the HDLP.

For designing the cryptoschemes based on the HDLP and for estimation of their security one should know the structure of the used algebraic carriers.

This paper considers some properties and structure of a particular 4-dimensional FNAA representing interest as the algebraic carrier of the HDLP and post-quantum cryptoschemes.

2 Preliminaries

The DLP is defined in a finite cyclic group in the form of finding solution of the equation $Y' = G^x$ relatively the unknown non-negative integer $x$. In the HDLP one of the values $G$ and $Y'$ or both of them are masked, i.e. instead of the values $G$ and $Y'$ there are given the values $Z$ and $Y$, respectively, which are not elements of the base cyclic group generated by the value $G$. The hiding is possible if the cyclic group represents a subset of some set of algebraic elements. The FNAAs suites very well as carriers of different forms of the HDLP.

In some finite $m$-dimensional vector space defined over the Galois field $GF(p)$ one can define additionally the vector multiplication operation, i.e., operation for multiplying arbitrary two vectors, which is distributive relatively the addition operation. The finite vector space complemented with the mentioned multiplication operation (denoted as $\circ$) is called finite algebra. If the multiplication operation is non-commutative and associative, then we have some FNAA. Suppose $e_0, e_1, \ldots, e_{m-1}$ are the basis vectors. The vector $A$ is denoted in the following two forms: $A = (a_0, a_1, \ldots, a_{m-1})$ and $A = a_0e_0 + a_1e_1 + \cdots + a_{m-1}e_{m-1}$, where $a_0, a_1, \ldots, a_{m-1} \in GF(p)$.

Usually the multiplication operation of two vectors $A$ and
$B = \sum_{i=0}^{m-1} b_i e_i$ is defined with the formula

$$A \circ B = \sum_{j=0}^{m-1} \sum_{i=0}^{m-1} a_i b_j (e_i \circ e_j),$$

(1)

where every of the products of different pairs of basis vectors $e_i \circ e_j$ is to be substituted by a vector $\mu e_k$ indicated in the so called basis vector multiplication table (BVMT). The value $\mu \in GF(p)$ is called structural constant. It is usually assumed that the intersection of the $i$th row and $j$th column defines the cell indicating the value of the product $e_i \circ e_j$.

3 The Studied 4-dimensional Algebra

3.1 Global and local units

The BVMT described by Table 1 defines the 4-dimensional FNAA, containing $p^2$ different global left-sided units. To obtain the formula describing the set of such units one should consider the vector equation

$$X \circ A = A,$$

(2)

where $A = (a_0, a_1, a_2, a_3)$ is a fixed 4-dimensional vector and $X = (x_0, x_1, x_2, x_3)$ is the unknown. Using the formula (1) and Table 1 one can represent the vector equation (2) in the form of the following system of four linear equations:

$$\begin{cases}
(x_0 + x_2) a_0 + \mu (x_1 + x_3) a_1 = a_0; \\
(x_0 + x_2) a_1 + (x_1 + x_3) a_0 = a_1; \\
(x_0 + x_2) a_2 + \mu (x_1 + x_3) a_3 = a_2; \\
(x_0 + x_2) a_3 + (x_1 + x_3) a_2 = a_3.
\end{cases}$$

(3)

Performing the variable substitution $u_1 = x_0 + x_2$ and $u_2 = x_1 + x_3$, the system (3) takes on the form of the following two independent systems
of two equations:

\[
\begin{align*}
    u_1a_0 + \mu u_2a_1 &= a_0; \\
    u_1a_1 + u_2a_0 &= a_1; \\
    u_1a_2 + \mu u_2a_3 &= a_2; \\
    u_1a_3 + u_2a_2 &= a_3.
\end{align*}
\]

(4)

(5)

For the vector \( A \) satisfying the conditions \( a_0^2 \neq a_1^2 \) and \( a_2^2 \neq a_3^2 \) each of the systems (4) and (5) has the same unique solution \( u_1 = 1 \) and \( u_2 = 0 \). The indicated solution satisfies the systems (4) and (5) for all vectors in the considered FNAA (in the cases \( a_0^2 = a_1^2 \) and \( a_2^2 = a_3^2 \) there exists some other (local) solutions). The solution \((u_1, u_2) = (1, 0)\) defines the set of the global left-sided units \( X \) coordinates of which satisfy the conditions \( x_0 + x_2 = u_1 = 1 \) and \( x_1 + x_3 = u_2 = 0 \). These units are called global, since every of them acts as unit on every vector in the FNAA. The set of \( p^2 \) global left-sided units \( L \) is described as follows:

\[
L = (l_0, l_1, l_2, l_3) = (h, k, 1 - h, -k),
\]

(6)

where \( h, k = 0, 1, 2, \ldots p - 1 \).

Table 1. The BVMT for defining the 4-dimensional FNAA (\( \mu \neq 1 \)).

<table>
<thead>
<tr>
<th>( \circ )</th>
<th>( e_0 )</th>
<th>( e_1 )</th>
<th>( e_2 )</th>
<th>( e_3 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( e_0 )</td>
<td>( e_0 )</td>
<td>( e_1 )</td>
<td>( e_2 )</td>
<td>( e_3 )</td>
</tr>
<tr>
<td>( e_1 )</td>
<td>( e_1 )</td>
<td>( \mu e_0 )</td>
<td>( e_3 )</td>
<td>( \mu e_2 )</td>
</tr>
<tr>
<td>( e_2 )</td>
<td>( e_0 )</td>
<td>( e_1 )</td>
<td>( e_2 )</td>
<td>( e_3 )</td>
</tr>
<tr>
<td>( e_3 )</td>
<td>( e_1 )</td>
<td>( \mu e_0 )</td>
<td>( e_3 )</td>
<td>( \mu e_2 )</td>
</tr>
</tbody>
</table>

The right-sided units relating to some vector \( A \) can be computed from the vector equation

\[
A \circ X = A
\]

(7)

that can be reduced to the following two systems of two linear equations
with the unknowns \( x_0, x_1 \) and \( x_2, x_3 \), correspondingly:

\[
\begin{align*}
(a_0 + a_2) x_0 + \mu (a_1 + a_3) x_1 &= a_0; \\
(a_1 + a_3) x_0 + (a_0 + a_2) x_1 &= a_1;
\end{align*}
\]

\[
\begin{align*}
(a_0 + a_2) x_2 + \mu (a_1 + a_3) x_3 &= a_2; \\
(a_1 + a_3) x_2 + (a_0 + a_2) x_3 &= a_3.
\end{align*}
\]

The systems (8) and (9) have the same main determinant \( \Delta_A \):

\[
\Delta_A = (a_0 + a_2)^2 - \mu (a_1 + a_3)^2.
\]

If \( \Delta_A \neq 0 \), then the single right-sided unit \( R_A = (r_0, a_1, r_2, a_3) \) cor-
sponds to the vector \( A \), which is described by the formula

\[
R_A = \left( \frac{a_0 (a_0 + a_2) - \mu a_1 (a_1 + a_3)}{\Delta}, \frac{a_1 a_2 - a_0 a_3}{\Delta}, \frac{a_2 (a_0 + a_2) - \mu a_3 (a_1 + a_3)}{\Delta}, \frac{a_0 a_3 - a_1 a_2}{\Delta} \right).
\]

In general different right-sided units \( R_A \) correspond to different vec-
tors \( A \), i. e., the right-sided units in the considered FNAA are local unit elements. Evidently, the value \( R_A \) acts as the local right-sided unit in the frame of the sequence of the vectors \( A, A^2, \ldots, A^i, \ldots \). Be-
sides, the latter sequence is periodic and compose a finite cyclic group
with the unit element equal to \( R_A \), i. e., the element \( R_A \) is the single
local two-sided unit \( E_A \) relating to the vector \( A \) (and to cyclic group
generated by the vector \( A \)).

**Proposition 1.** The local right-sided unit \( R_A \) is contained in the
set (6) of the global right-sided units.

**Proof.** For global left-sided vector \( L = (l_0, l_1, l_2, l_3) \) with the first and
second coordinates \( l_0 = r_0 \) and \( l_1 = r_1 \) computation of the coordi-
nates \( l_2 \) and \( l_3 \) in accordance with the formula (6) gives the following:

\[
l_2 = 1 - r_0 = 1 - \frac{a_0 (a_0 + a_2) - \mu a_1 (a_1 + a_3)}{\Delta} = \frac{a_0 a_3 - a_1 a_2}{\Delta} = r_2;
\]
3.2 Global zero divisors

The vector $A$ such that $\Delta_A \neq 0$ is invertible relatively the local two-sided unit $E_A$, therefore one can call it a locally invertible vector.

**Proposition 2.** If the structural constant $\mu$ is a quadratic non-residue, then for all vectors $A$ in the considered 4-dimensional FNAA, except the vectors of the form $(h, k, -h, -k)$, where $h, k = 0, 1, \ldots, p - 1$, the non-equality $\Delta_A \neq 0$ holds true.

**Proof.** If $\mu$ is quadratic non-residue, then

$$\{\Delta_A = 0\} \Rightarrow \{(a_0 + a_2)^2 = \mu (a_1 + a_3)^2\} \Rightarrow$$

$$\Rightarrow \{a_0 + a_2 = 0; \ a_1 + a_3 = 0\} \Rightarrow \{a_2 = -a_0; \ a_3 = -a_1\}.$$  

(2)

If the structural constant $\mu$ is a quadratic non-residue, then no local right-sided unit corresponds to vectors $D$ satisfying the condition $\Delta_D = 0$. Such vectors are called non-invertible vectors. One can show that the vectors $D$ are global left-sided zero divisors, i. e., for every vector $D$ the following condition holds true for arbitrary vector $V = (v_0, v_1, v_2, v_3)$ of the considered FNAA:

$$D \circ V = O,$$  

(12)

where $O = (0, 0, 0, 0)$. Like in the case of the equation (2), the equation (12) with the unknown vector $D = (d_0, d_1, d_2, d_3)$ can be represented in the form of the following two independent systems of two equations:

$$\begin{cases}
  u_1v_0 + \mu u_2v_1 = 0; \\
  u_1v_1 + u_2v_0 = 0;
\end{cases}$$  

(13)
\[
\begin{align*}
\begin{cases}
u_1v_2 + \mu u_2v_3 &= 0; \\
u_1v_3 + u_2v_2 &= 0,
\end{cases}
\end{align*}
\] (14)

where \( u_1 = d_0 + d_2 \) and \( u_2 = d_1 + d_3 \).

For the vector \( V \) satisfying the conditions \( v_0^2 \neq v_1^2 \) and \( v_2^2 \neq v_3^2 \) each of the systems (13) and (14) has the same unique solution \( u_1 = 0 \) and \( u_2 = 0 \). The indicated solution satisfies the systems (13) and (14) for all vectors in the considered FNAA (in the cases \( v_0^2 = v_1^2 \) and \( v_2^2 = v_3^2 \) there exists some other solutions). The solution \((u_1, u_2) = (0, 0)\) defines the set of the global left-sided zero divisors \( D \) coordinates of which satisfy the conditions \( d_0 + d_2 = u_1 = 0 \) and \( d_1 + d_3 = u_2 = 0 \). These zero divisors are called global, since every of them acts as a zero divisor on every vector in the FNAA. There exist \( p^2 \) different global left-sided zero divisors that are described by the following formula:

\[
D = (d_0, d_1, d_2, d_3) = (h, k, -h, -k),
\] (15)

where \( h, k = 0, 1, 2, \ldots, p - 1 \). From Proposition 2 one can conclude that every global left-sided zero divisor is a non-invertible vector and every non-invertible vector is a global left-sided zero divisor.

**Proposition 3.** Suppose \( J = (0, 0, 1, 0) \). Then for every global left-sided zero-divisor \( D \) from the set (15) the following condition holds true \( L = D + J \), where \( L \) is a global left-sided unit. Besides, the formula \( L = D + J \), where \( D \) takes on all values from the set (15), describes the set (6).

**Proof.** From the formulas (6) and (15) we have \( L - D = J \), hence \( L = D + J \) and \( D = L - J \). The last two formulas define a bijective map of the sets (6) and (15).

The value \( J \) is contained in the set (6), i.e., \( J \) is one of global left-sided units.

### 3.3 Homomorphism of the \( \varphi_L \) type

To define different versions of the HDLP the automorphism map and homomorphism map operations are used as masking operations [5, 7, 8].
In the case of the considered 4-dimensional FNAA two different types of the homomorphism maps represent interest to define different forms of the HDLP.

Some fixed global left-sided unit $L$ sets a homomorphism map of the FNAA, called the homomorphism of the $\varphi_L$-type.

**Proposition 4.** Suppose the vector $L$ is a global left-sided unit. Then the map of the FNAA defined by the formula $\varphi_L(X) = X \circ L$, where the vector $X$ takes on all values in the algebra, is a homomorphism.

*Proof.* For two arbitrary vectors $X_1$ and $X_2$ one can get the following:

$\varphi_L(X_1 \circ X_2) = (X_1 \circ X_2) \circ L = (X_1 \circ L) \circ (X_2 \circ L) = \varphi_L(X_1) \circ \varphi_L(X_2)$;

$\varphi_L(X_1 + X_2) = (X_1 + X_2) \circ L = X_1 \circ L + X_2 \circ L = \varphi_L(X_1) + \varphi_L(X_2)$.

*Proposition 5.* The homomorphism-map operation $\varphi_L(X) = X \circ L$, where $L$ is a global left-sided unit, and the exponentiation operation $X^i$ are mutually commutative, i.e., the equality $X^i \circ L = (X \circ L)^i$ holds true.

*Proof.* Due to Proposition 4 we have $\varphi_L(X^i) = (\varphi_L(X))^i$, i.e., $X^i \circ L = (X \circ L)^i$.

*Proposition 6.* Suppose for some fixed global left-sided unit $L$ we have $\varphi_L(X) = X \circ L = U$. Then $R_U = L$, i.e. the function $\varphi_L(X)$ takes on the values the local right-sided unit of which is equal to $L$.

*Proof.* $U \circ L = X \circ L \circ L = X \circ L = U$.

*Proposition 7.* Suppose for some fixed vector $X$ and the global left-sided unit $L$ we have $\varphi_L(X) = X \circ L = U$. Then for all $p^2$ vectors of the form $V_U = U \circ L'$, where $L'$ takes on all values from the set $[6]$, the equality $\varphi_L(V_U) = V_U \circ L = U$ holds true.

*Proof.* $\varphi_L(V_U) = U \circ L' \circ L = U \circ L = U$. 

66
Proposition 8. Suppose for some fixed vector $X$ global left-sided unit $L$ we have $\varphi_L(X) = X \circ L = U$, where $U$ is such that $\Delta_U \neq 0$. Then the vector $X$ can be represented in the form $X = U \circ L'$, where $L'$ is a global left-sided unit, i.e., exactly $p^2$ different vectors of the considered $4$-dimensional algebra are mapped into the fixed value $U$.

Proof. An arbitrary global left-sided zero divisor $D$ can be represented in the form $D = U \circ D'$, where $D'$ is also a global left-sided zero divisor. Indeed, the equation $U \circ X = D$ has unique solution $X = D'$, since $\Delta_U \neq 0$. Evidently, for an arbitrary vector $V$ such that $\Delta_V \neq 0$ we have the following: $(U \circ D') \circ V = U \circ (D' \circ V) = O$. Since $\Delta_U \neq 0$, from the last equality we have $D' \circ V = O$, hence $D'$ is a global left-sided zero divisor.

One can write: \{ $X \circ L = U; \ U \circ L = U$ \} $\Rightarrow$ $(X - U) \circ L = O$ $\Rightarrow$ $(X - U) = D \Rightarrow X = U + D$, where $D$ is a global left-sided zero divisor. The last equality can be represented in the form $X = U \circ L + U \circ D' = U \circ (L + D') = U \circ L'$, where $L'$ is a global left-sided unit.

Proposition 9. If the vector $G$ satisfying the condition $\Delta_G \neq 0$ is not a global left-sided unit, then for an arbitrary natural number $k$ such that $G^k \neq G$ the non-equality $\varphi_L(G^k) \neq \varphi_L(G)$ holds true.

Proof. Suppose $\varphi_L(G^k) = \varphi_L(G)$. Then due to the Proposition 4 we have $\varphi_L(G^k - G) = \varphi_L(G^k) - \varphi_L(G) = O$. Due to the Proposition 8 we have $G^k - G = O \circ L' = O \Rightarrow G^k = G$. The obtained contradiction proves the Proposition 9.

Proposition 10. If the vector equation $X \circ V = Z$ has solution $X = S$, then $p^2$ different values $X_i = S \circ L_i$, where $L_i$ takes on all values from the set $\{L\}$, also are solutions of the given equation.

Proof. $(S \circ L_i) \circ V = S \circ (L_i \circ V) = S \circ V = Z$. Evidently, for every global left-sided unit $L'$ we have $\varphi_L(L') = L$ and for every global left-sided zero divisor $D$ we have $\varphi_L(D) = O$. 

67
From the Proposition 9 it is easy to see that some fixed cyclic group contained in the algebra is mapped with the function $\varphi_L(X)$ into another cyclic group of the same order. Moreover, if the vector $G$ is generator of the cyclic group having order $\omega$, then the algebra contains $p^2$ different cyclic groups of the order $\omega$ generated by the generators $G \circ L'$, where $L'$ take on all values from the set of global left-sided units (6). Every one of these cyclic groups is mapped by the function $\varphi_L(X)$ into the cyclic group generated by the vector $G \circ L$.

Suppose $\omega$ is the maximum value in the set of different local orders of the locally invertible elements of the considered 4-dimensional FNAA. An algebraic element having local order equal to $\omega$ generates a cyclic group of the same order. The algebra contains $p^2$ of cyclic groups having order $\omega$ and every locally invertible vector is contained in one of the mentioned cyclic groups. Taking into account that number of locally invertible vectors is equal to $p^4 - p^2$ (order of the algebra minus number of the global left-sided zero divisors), one can write

$$p^4 - p^2 = p^2 \omega.$$ 

From the last equality we have the following formula for the order of every cyclic group contained in the considered FNAA:

$$\omega = p^2 - 1. \quad (16)$$

### 3.4 Homomorphism of the $\psi_L$ type

Suppose the vector $A$ is such that $\Delta_A \neq 0$. Then one can select a random global left-sided unit $L$ and compute the single vector $B$ that satisfies the condition

$$A \circ B = L. \quad (17)$$

**Proposition 11.** Suppose $A \circ B = L$. Then for arbitrary natural number $t$ the equality $A^t \circ B^t = L$ holds true.

**Proof.**

$$A^t \circ B^t = A^{t-1} \circ (A \circ B) \circ B^{t-1} = A^{t-1} \circ B^{t-1} = \ldots A^{t-i} \circ B^{t-i} \ldots = A \circ B = L.$$
Proposition 12. Suppose $A \circ B = L$. Then the formula

$$\psi_L = B \circ X \circ A,$$

where the vector $X$ takes on all values in the considered 4-dimensional FNAA, sets the homomorphism map, called the $\psi_L$-type homomorphism.

Proof. For two arbitrary 4-dimensional vectors $X_1$ and $X_2$ one can get the following:

$$\psi_L (X_1 \circ X_2) = B \circ (X_1 \circ X_2) \circ A = B \circ (X_1 \circ L \circ X_2) \circ A =$$

$$= (B \circ X_1 \circ A) \circ (B \circ X_2 \circ A^t) = \psi_L (X_1) \circ \psi_L (X_2);$$

$$\psi_L (X_1 + X_2) = B \circ (X_1 + X_2) \circ A = (B \circ X_1 \circ A) + (B \circ X_2 \circ A) =$$

$$= \psi_L (X_1) + \psi_L (X_2).$$

Proposition 13. The $\psi_L$-type homomorphism-map operation $\psi_L(X) = B \circ X \circ A$ and the exponentiation operation $X^k$ are mutually commutative, i.e., the equality $B \circ X^k \circ A = (B \circ X \circ A)^k$ holds true.

Proof. Due to Proposition 12 we have $\psi_L(X^k) = (\psi_L(X))^k$, i.e., $B \circ X^k \circ A = (B \circ X \circ A)^k$.

Proposition 14. Suppose $V$ is an arbitrary fixed value. Then every one of the elements $V + D$, where $D$ takes on all values from the set (15), is mapped with the function $\psi_L$ into the value $\psi_L(V)$.

Proof. We have $\psi_L(V + D) = \psi_L(V) + \psi_L(D) = \psi_L(V) + O = \psi_L(V)$. □
Proposition 15. Suppose $V$ is an arbitrary fixed locally invertible element order of which is equal to $\omega$. Then the local right-sided unit relating to the value $V + D$, where $D$ is an arbitrary left-sided zero divisor from the set (15), is equal to the value $R_{V+D} = R_V + V^{\omega-1} \circ D$, where $R_V$ is the local right-sided unit related to the vector $V$.

Proof. Taking into account that the local right-sided unit $R_V$ is simultaneously on of the global left-sided units, we have $(V + D) \circ (R_V + V^{\omega-1} \circ D) = V \circ R_V + V^\omega \circ D = V + R_V \circ D = V + D$. \qed

Proposition 16. Suppose $V$ is an arbitrary fixed locally invertible element order of which is equal to $\omega$. Then the order of every of the values $V + D$, where $D$ takes on all values from the set (15), is equal to $\omega$.

Proof. We have $(V + D)^\omega = (V + D) \circ (V + D)^{\omega-1} = V \circ (V + D)^{\omega-1} = V^2 \circ (V + D)^{\omega-2} = V^{\omega-1} \circ (V + D) = V^\omega + V^{\omega-1} \circ D = R_V + V^{\omega-1} \circ D = R_{V+D}$. \qed

From the Propositions 14, 15 and 16 it is easy to see that every of $p^2$ cyclic groups contained in the in the considered 4-dimensional FNAA is mapped with the function $\varphi_L(X)$ into the single cyclic group generated by the vector $\varphi_L(G)$, where $G$ is an arbitrary vector having order equal to $p^2 - 1$.

4 Cryptoschemes Based on the HDLP

4.1 Public key-agreement scheme

Using the considered FNAA as algebraic carrier one can implement the following protocol for public key-agreement. Suppose three locally invertible 4-dimensional vectors $G$, $A$ and $B$ satisfy the condition $A \circ B = L$, where $L$ is a global left-sided unit. Then the first and second users can select the following two pairs of random integers $(x_1, t_1)$ and $(x_2, t_2)$ and compute their public keys $Y_1 = B^{t_1} \circ G^{x_1} \circ A^{t_1}$ and $Y_2 = B^{t_2} \circ G^{x_2} \circ A^{t_2}$. 

70
Procedure of generating a common secret key with using a public channel is as follows:
1. The first user sends his public key $Y_1$ to the second user.
2. The second user sends his public key $Y_2$ to the first user.
3. The first user computes the value
   \[ Z_1 = B^{t_1} \circ Y_2^{x_1} \circ A^{t_1}. \]
4. The second user computes the value
   \[ Z_2 = B^{t_2} \circ Y_1^{x_2} \circ A^{t_2}. \]

It is easy to show that the equality $Z_1 = Z_2$ holds true.

### 4.2 Digital signature scheme

To generate a public key one is to select a random integer $x < \omega$ (the private key) and the set of the vectors $G, U, N, T$, and $W$ satisfy the following conditions $T \circ U = L_1$ and $W \circ N = L_2$, where $L_1$ and $L_2 \neq L_1$ are some global left-side units.

The signer’s public key represents the triple of the vectors $Y, H,$ and $Z$ that are computed so that the following equations hold true:

\[
Y = U \circ G^x \circ T = (U \circ G \circ T)^x; \quad Z = N \circ G \circ W; \quad T \circ H \circ N = L_3,
\]

where $L_3$ is a global left-side unit such that $L_3 \neq L_1$ and $L_3 \neq L_2$.

The digital signature $(e, s)$ to some document $M$ is computed as follows:
1. Select at random an integer $k < \omega$ and compute the vector
   \[ K = U \circ G^k \circ W. \]
2. Compute the first element of the signature $e = F_h(M, K)$.
3. Compute the second element of the signature $s = k - xe \mod \omega$.

The signature verification is performed as follows:
1. Compute the vector
   \[ K' = Y^e \circ H \circ Z^s. \]
2. Compute the value \( e' = F_h(M, K') \).

3. Compare the values \( e' \) and \( e \), If \( e' = e \), then the signature is accepted as genuine. Otherwise the signature is rejected as false.

This signature scheme is based on the HDLP consisting in finding the value \( x \) when the vectors \( Y, Z, \) and \( H \) are known. The last three vectors are contained in different cyclic groups every one of which is different from the base cyclic group generated by the vector \( G \).

5 Conclusion

Properties and structure of a particular 4-dimensional FNAA have been described. The considered algebra is characterized in existing \( p^2 \) different global left-sided units. If the structural constant \( \mu \) is a quadratic non-residue, then the algebra includes \( p^4 - p^2 \) locally invertible vectors every one of which is included in one of the \( p^2 \) different cyclic groups contained in the algebra. The algebra contains \( p^2 \) global left-sided zero divisors. Using the \( \psi_L \)-type homomorphism map operation the public key agreement and signature schemes based on the HDLP have been proposed as candidates for post-quantum public-key cryptoschemes.

References


Four-dimensional Finite Non-commutative Associative Algebra...


Alexandr Moldovyan$^{1}$, Dmitrii Moldovyan$^{2}$, Victor Shcherbacov$^{3}$

$^{1}$Professor / St. Petersburg Institute for Informatics and Automation of Russian Academy of Sciences
14-th line 39, 199178, St. Petersburg
Russia
Email: maa1305@yandex.ru

$^{2}$Assistant Professor / St. Petersburg Institute for Informatics and Automation of Russian Academy of Sciences
14-th line 39, 199178, St. Petersburg
Russia
Email: mdn.spectr@mail.ru

$^{3}$Principal Researcher / Institute of Mathematics and Computer Science, Academiei, 5, Chisinau, MD-2028, Moldova
Email: victor.scherbacov@math.md
On Anonymization of Cocks’ Identity-based Encryption Scheme

Anca-Maria Nica, Ferucio Laurenţiu Tiplea

Abstract

Cocks’ identity-based encryption (IBE) scheme is the first IBE scheme that avoids the use of bilinear maps. Based on quadratic residues and due to its simplicity, the scheme gained much attention from researchers. Unfortunately, the scheme is not anonymous in the sense that the cryptotexts may reveal the identities for which they have been computed. Several anonymous variants of it have then been proposed.

In this paper we revise Joye’s approach to the anonymization of the Cocks’ IBE scheme. Due to some recent results on the distribution of quadratic residues, we present a very simple and direct approach that leads to Joye’s scheme.

Keywords: identity-based encryption, quadratic residue, indistinguishability

1 Introduction

In 1984 Adi Shamir proposed the idea of identity-based encryption [12], which is a special case of public-key encryption. This model avoids the public-key infrastructure and the trust chain for public keys. It uses instead a string which uniquely identifies the receiver and computes his public key based on it, using a publicly known hash function. An IBE cryptosystem has four probabilistic polynomial time (PPT) algorithms. The \textsc{Setup}(\lambda) algorithm outputs the public parameters together with the master secret, having as input the security parameter. The \textsc{Key-Gen}(PP,msk,ID) algorithm outputs the secret key for the identity
The third algorithm, $Enc(PP, m)$, computes the ciphertext of the message $m$ for a given identity, while the last algorithm, $Dec(sk_{ID}, c)$, outputs the initial message from the cryptotext $c$ using the secret key of the identity for which the message $m$ was encrypted.

In his paper [12], Shamir showed how one can sign using identity-based (IB) paradigm, but, regarding IB cryptosystems, only seventeen years later a concrete solution was crated. Thus, in 2001, two IBE schemes were presented. The first one was due to Boneh and Franklin [4], relying on bilinear maps, while the second one was due to Clifford Cocks [7], who proposed a pairing-free cryptosystem, using quadratic residues. Despite its cryptotext expansion, Cocks’ scheme was a starting point for many quadratic residues (QR)-based IBE schemes by virtue of its simplicity and elegance [14].

It was shown in [3] that Cocks’ scheme does not provide the anonymity of the receiver’s public key in the sense of Bellare et al. [2]. From this point on, several ideas and schemes have been proposed in order to obtain anonymous variants. The first one was due to Di Crescenzo and Saraswat [8] in 2007, but it is quite impractical because it uses four keys per bit of plaintext. Another one was due to Boneh, Gentry, and Hamburg [5], in the same year, whose security is also discussed in [13]. They improved the ciphertext length and provided anonymity but the time complexity became quartic in the security parameter. Two years later, Ateniese and Gasti created a very interesting solution by the fact that their IBE scheme is universally anonymous [9]. This means that the anonymization process is independent from encryption and can be done using only the public key of the receiver. In 2014, Clear, Tewari, and McGoldrick [6] designed a variant of Cocks’ IBE scheme which, beside the fact that it is anonymous, it keeps the time complexity close to the initial scheme and, further more, outputs shorter cryptotexts. In the same year, another scheme providing universal anonymity was proposed by Schipor [11]. Compared to the universal anonymous variant of Ateniese et al. [1], the ciphertext expansion of Schipor’s solution is considerably smaller. Joye [10] also described an anonymous version of Cocks’ scheme in 2016, starting deeper, from finding the algebraic
structure of Cocks’ IBE cryptotexts and showing also its homomorphic character.

In this paper we show how to obtain the anonymization in Joye’s work [10] easily, using the new results on the distribution of quadratic residues in [15], where a deep study on the set of Cocks’ IBE cryptotexts was undertaken.

2 Cocks’ IBE Scheme and Anonymization

The first pairing-free IBE scheme was proposed by Cocks and it is based on quadratic residues [7]. The scheme is IND-ID-CPA secure in the random oracle model (ROM) under the Quadratic Residuosity Assumption (QRA) modulo a large RSA integer. Cocks’ IBE scheme is defined by four probabilistic algorithms, as it is described in Algorithm 1 on page 68.

It was mentioned in [3, Section 4] that the scheme is not anonymous, in the sense that the outputted cryptotexts contain information about the receiver, so one can check if the ciphertext was encrypted for a specific identity. The tool which helps to decide this is Galbraith’s test, which was briefly described in [1,3].

In order to understand Galbraith’s test, we will turn our attention to Cocks ciphertexts, and mainly follow the approach in [15].

Working in an RSA group $\mathbb{Z}_n^*$, we will consider an integer $a$, which will have the Jacobi symbol modulo $n$ equal to 1 ($a$ corresponds to some identity $ID$, but, for simplicity, we will call $a$ the identity). Then, a cryptotext computed for the identity $a$ will have the following form $c = t + at^{-1} \mod n$, for some $t \in \mathbb{Z}_n^*$. That is, $t$ is a solution to the degree two congruence $t^2 - ct + a \equiv 0 \mod n$. It is useful then to denote [15]:

\[
\begin{align*}
C_n(a) &= \{(t + at^{-1})_n \mid t \in \mathbb{Z}_n^*\} \\
C_n^*(a) &= C_n(a) \cap \mathbb{Z}_n^* \\
C_n^{e_1,e_2}(a) &= \left\{c \in \mathbb{Z}_p^* \mid \left(\frac{c^2 - 4a}{p}\right) = e_1, \left(\frac{c^2 - 4a}{q}\right) = e_2\right\}
\end{align*}
\]
Algorithm 1 Cocks’ IBE scheme

procedure Setup(\( \lambda \)):
\[
\begin{align*}
\text{generates at random two large primes } (p, q) & \leftarrow \text{RSA}_\text{gen}(\lambda); \\
n & = pq; \\
e & \in R J_n^+ \setminus QR_n; \quad \triangleright \frac{1}{2} \text{ elements in } J_n^+ \text{ are q.non-residues} \\
h : \{0, 1\}^* & \rightarrow J_n^+; \quad \triangleright \text{ a hash function} \\
PP: n, e, h; & \quad \triangleright \text{ the public parameters} \\
msk: p, q; & \quad \triangleright \text{ the master secret key} \\
\text{return } (PP, msk).
\end{align*}
\]
end procedure

procedure KeyGen(msk, ID):
\[
\begin{align*}
a & = h(ID); \quad \triangleright \text{ the identity} \\
\text{if } a \in QR_n \text{ then} & \\
r & = a^{1/2}; \\
\text{else } r & = (ea)^{1/2}; \\
\text{end if} \\
\text{return } r. \quad \triangleright \text{ the secret key of the identity ID}
\end{align*}
\]
end procedure

procedure Enc(PP, ID, m):
\[
\begin{align*}
a & = h(ID); \quad \triangleright \text{ where } m \in \{\pm 1\} \\
t_1, t_2 & \leftarrow \mathbb{Z}_n^* \text{ such that } (\frac{t_1}{n}) = (\frac{t_2}{n}) = (\frac{m}{n}); \\
c_1 & = t_1 + at_1^{-1} \mod n \quad \text{and } c_2 = t_2 + eat_2^{-1} \mod n; \\
\text{return } (c_1, c_2).
\end{align*}
\]
end procedure

procedure Dec(PP, r, (c_1, c_2)):
\[
\begin{align*}
\text{if } r^2 & \equiv h(ID) \mod n \text{ then } c = c_1; \\
\text{else } c & = c_2; \\
\text{end if} \\
m & = \left(\frac{c + 2r}{n}\right); \\
\text{return } m.
\end{align*}
\]
end procedure
where \( \left( \frac{x}{m} \right) \) stands for the Jacobi symbol of \( x \) modulo \( m \), \( n \) is the product of two primes \( p, q \) and \( e_1, e_2 \in \{-1, 0, 1\} \).

The Galbraith’s test of \( c \in \mathbb{Z}_n^* \) w.r.t. \( a \), denoted \( GT_{n,a}(c) \), is

\[
GT_{n,a}(c) = \left( \frac{c^2 - 4a}{n} \right)
\]

Let \( G_n(a) = \{ c \in \mathbb{Z}_n^* \mid GT_{n,a}(c) = 1 \} \). Clearly,

\[ G_n(a) = C_{n,1}^{1,1}(a) \cup C_{n,-1}^{-1,-1}(a) \]

An integer \( c \in \mathbb{Z}_n^* \) passes Galbraith’s test w.r.t. \( n \) and \( a \) if \( GT_{n,a}(c) = 1 \) (or, \( c \in G_n(a) \)). Not all Cocks ciphertexts pass Galbraith’s test, but most of them do. The diagram in Figure 1 shows clearly which Cocks ciphertexts pass Galbraith’s test.

Galbraith’s test for anonymity is then described in Algorithm 2.

**Algorithm 2 Galbraith’s Test**

<table>
<thead>
<tr>
<th>Input</th>
<th>RSA modulus ( n ), ( a \in J_n^+ ), and ( c \in \mathbb{Z}_n^* )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Output</td>
<td>1 / 0</td>
</tr>
<tr>
<td>if</td>
<td>( \left( \frac{c^2 - 4a}{n} \right) = 1 ) then</td>
</tr>
<tr>
<td></td>
<td>1 \hspace{2cm} \text{( c \in C_{n}^*(a) ) with prob. ( \frac{1}{2} - O\left( \frac{1}{\sqrt{n}} \right) )}</td>
</tr>
<tr>
<td>else</td>
<td>0 \hspace{2cm} \text{( c \in C_{n}^*(a) ) with negl. prob.}</td>
</tr>
<tr>
<td>end if</td>
<td></td>
</tr>
</tbody>
</table>
We note that if \( \left( \frac{c^2 - 4a}{n} \right) = -1 \), then \( c \not\in C_n^*(a) \) (with probability 1). However, if \( \left( \frac{c^2 - 4a}{n} \right) = 1 \), then \( c \) is a Cocks ciphertext with probability overwhelmingly close to 1/2.

The key-privacy test consists then in repeatedly applying Galbraith’s test with values sampled from either \( G_n(a) \) (and \( G_n(ea) \)) or \( G_n(b) \) (and \( G_n(eb) \)), where \( e \) is as in Cocks’ IBE scheme.

3 Another View on Joye’s Scheme

If we look again to the set of Cocks ciphertexts, whose main part consists of \( C_n^{1,1}(a) \), we may imagine the following very simple method to anonymize them: given \( c \in C_n^{1,1}(a) \), modify it into \( c' \) on a random basis such that \( GT_{n,a}(c') = \pm 1 \). As we have to decrypt \( c' \), the ciphertext \( c \) must be altered in such a way so that the receiver be able to reverse it. From this point of view, the simplest method seems to choose and fix \( d \) such that \( GT_{n,a}(d) = -1 \), and then to look for a binary operation \( \circ \) on \( \mathbb{Z}_n^* \) such that

\[
GT_{n,a}(c \circ d) = GT_{n,a}(c) \cdot GT_{n,a}(d).
\]

If such a binary operation is found, then we may take \( c' = c \circ d \) (but, once again, we flip a coin to decide whether we keep \( c \) or compute \( c' \)).

Under these circumstances we define the operation

\[
u \circ v = \frac{uv + 4a}{u + v} \mod n,
\]

for all \( u, v \in \mathbb{Z}_n^* \) with \( (u + v, n) = 1 \).

Although this operation depends on \( n \) and \( a \), for the sake of simplicity, we will simply denote it by \( \circ \). Its basic properties are listed below.

**Proposition 1.** Let \( u, v, w \in \mathbb{Z}_n^* \) and \( a \in J_n^+ \). Then:

1. When defined, \( \circ \) is associative

\[
u \circ (v \circ w) = (u \circ v) \circ w.
\]
2. If \((u + v, n) = 1\) and \((v^2 - 4a, n) = 1\), then

\[(u \circ v) \circ (-v) = u\]

(remark that \(v \circ (-v)\) is not defined).

3. \(GT_{n,a}(u \circ v) = GT_{n,a}(u) \cdot GT_{n,a}(v)\), provided that \(u \circ v\) is defined.

4. \(u \circ u \in G_n(a)\).

Proof. (1) and (2) follow directly from the definition of \(\circ\). For (3) we have:

\[
GT_{n,a}(u \circ v) = \left(\frac{(u^2-4a)(v^2-4a)(u+v)^2}{n}\right)
\]

\[
= \left(\frac{u^2-4a}{n}\right) \left(\frac{v^2-4a}{n}\right)
\]

\[
= GT_{n,a}(u) \cdot GT_{n,a}(v).
\]

(4) follows from (3).

Proposition 1(3) says that \(u \circ v\) passes Galbraith’s test if and only if both \(u\) and \(v\) pass Galbraith’s test or both do not pass Galbraith’s test (provided that \(u \circ v\) is defined), and Proposition 1(4) says that \(u \circ u\) always passes Galbraith’s test.

Fortunately, this is all we need to obtain Joye’s approach to the anonymization of Cocks’ IBE scheme, as we can see in Algorithm 3 on page 72.

The correctness of the scheme in Algorithm 3 simply follows from Proposition 1. As with respect to security, we have the following result.

**Theorem 1.** Cocks’ AnonIBE scheme is ANON-IND-ID-CPA secure in the random oracle model under the QRA.

**Proof.** Any adversary \(A\) against Cocks’ AnonIBE scheme can be transformed into an adversary \(A'\) against Cocks’ IBE scheme, with an advantage greater than or equal to the advantage of \(A\). Therefore, as Cocks’ IBE scheme is IND-ID-CPA, Cocks’ AnonIBE scheme must be.
Algorithm 3 Cocks’ AnonIBE scheme

```plaintext
procedure Setup(λ):
P = (n, e, d, h)

> where n and e are as in Cocks’ IBE scheme

\(d \leftarrow \mathbb{Z}_n^*\) and \(h : \{0,1\}^* \rightarrow J_n^+\) are chosen so that
\(GT_{n,a}(d) = -1 = GT_{n,ea}(d)\), for any output \(a\) of \(h\)

msk = (p, q)
return (PP, msk).
end procedure

procedure Ext(msk, ID):
a = h(ID);
> private key: random square root \(r\) of \(a\) or \(ea\)
return r.
end procedure

procedure Enc(PP, ID, m):
a = h(ID);
t_0, t_1 \leftarrow \mathbb{Z}_n^* \text{ with } J_n(t_0) = m = J_n(t_1);
c_0 \leftarrow \{u, u \circ d\} \text{ where } u = t_0 + at_0^{-1} \mod n;
c_1 \leftarrow \{v, v \circ d\} \text{ where } v = t_1 + eat_1^{-1} \mod n;
return (c_0, c_1).
end procedure

procedure Dec((c_0, c_1), r):
set \(b \in \{0,1\}\) such that \(e^b a \equiv_n r^2 \mod n\);
return \(m = \begin{cases} J_n(c_b + 2r), & \text{if } GT_{n,e^b a}(c_b) = 1 \\ J_n(c_b \circ (-d) + 2r), & \text{otherwise} \end{cases}\)
end procedure
```

82
To prove that Cocks’ AnonIBE scheme is anonymous, consider the sets

\[ \mathcal{D}_0 = G_n(a_0) \cup (G_n(a_0) \circ d) \]

and

\[ \mathcal{D}_1 = G_n(a_1) \cup (G_n(a_1) \circ d), \]

where \( a_0 \) and \( a_1 \) are two public keys and \( d \) is as in Cocks’ AnonIBE scheme. We prove that \( c \in \mathcal{D}_0 \) iff \( c \in \mathcal{D}_1 \).

Let \( c \in \mathcal{D}_0 \). Assume first that, if \( GT_{n,a_1}(c) = 1 \), then \( c \in \mathcal{D}_1 \). Otherwise, \( GT_{n,a_1}(c \circ (-d)) = 1 \), which shows that \( c \circ (-d) \in G_n(a_1) \). But then, \( (c \circ (-d)) \circ d \in G_n(a_1) \circ d \), which leads to \( c \in G_n(a_1) \circ d \subseteq \mathcal{D}_1 \).

Assume now that \( GT_{n,a_0}(c) = -1 \). Then, \( GT_{n,a_0}(c \circ (-d)) = 1 \), which proves that \( c \circ (-d) \in G_n(a_0) \subseteq \mathcal{D}_0 \). The above argument proves then \( c \circ (-d) \in \mathcal{D}_1 \), from which follows that \( c \in \mathcal{D}_1 \).

As a conclusion, if \( c \in \mathcal{D}_0 \) then \( c \in \mathcal{D}_1 \). Due to the symmetry of this relation, we get the theorem.

4 Conclusion

In this paper we achieved the anonymization in Joye’s work [10] in a significantly easier way. The main resource used in order to attain this was the new results on quadratic residues in [15]. This extensive study facilitates a better understanding of the cryptotexts outputted by Cocks’ IBE scheme and their structure. Further more, [15] facilitates a detailed analyze of Galbraith’s test, capturing the essence of the anonymization process regarding Cocks’ IBE cryptotexts. This allowed us to bring clarity and capture the essence in Joye’s variant. Thus, we believe that our work will help the researcher community and we also hope that it will have an incentive effect for further research on this topic.
References


On Anonymization of Cocks’ IBE Scheme


Anca-Maria Nica, Ferucio Laurențiu Țiplea

Department of Computer Science, “Al.I.Cuza” University of Iași

Email: {contact@ancamarianica.ro, ferucio.tiplea@uaic.ro}
Attacks on generalized Markovski crypto-algorithm

Nadezhda Malyutina

Abstract

We consider modifications of Markovski crypto-algorithm [1, 2] based on n-ary groupoids [3], which are invertible on at least one place. We describe the attacks on this algorithm using the selected ciphertext and the plaintext.

Keywords: cipher, ciphertext, plaintext, Markovski algorithm, n-ary groupoid, quasigroup.

1 Introduction

The usage of quasigroups in cryptography is not very common. In spite of that various cryptosystems based on quasigroups appeared in past few years, e.g. see [4] where it is shown that the usage of quasigroups opens new ways in the design of block ciphers.

A new stream cipher for the encryption of a file system was proposed in [5]. The cipher has a very large keyspace and was claimed to be resistant against any attack.

S. Markovski [2] (see also E. Ochodkova and V. Snashel [5]) proposed a new stream cipher to encrypt the file system. M. Vojvoda has given a cryptanalysis of the file encoding system based on quasigroups [6] and showed how to break this cipher.

We consider modification of Markovski crypto-algorithm [1, 2] based on the use of n-ary groupoids [3], which are invertible on at least one place.

It is clear that Markovski crypto-algorithm which is based on binary or n-ary quasigroups has better «mixing properties» than proposed algorithm [7].

But from the other side it is well known that binary (n-ary) quasigroup that is used in Markovski algorithm is its key [8]. It is clear that number of i-invertible n-groupoids (number n is fixed) is much more then number of n-ary quasigroups (number n is fixed). This was the impetus for the construction of a generalization of the Markovski algorithm.
We describe attacks on the cipher using the selected ciphertext and the plaintext.

2 Generalized algorithm

**Definition 2.1** $n$-ary groupoid $(Q, f)$ is called invertible on the $i$-th place, $i = 1, n$, if the equation: $f(a_1, ..., a_i, x_i, a_{i+1}, ..., a_n) = a_{n+1}$ has unique solution for any elements $a_1, ..., a_i, a_{i+1}, ..., a_n, a_{n+1} \in Q$.

In this case operation $(i, n+1)f(a_1, ..., a_i-1, a_{i+1}, ..., a_n) = x_i$ is defined in unique way and we have:

$$f(a_1, ..., a_i-1, (i, n+1)f(a_1, ..., a_i-1, a_{i+1}, ..., a_n), a_{i+1}, ..., a_n) = a_{n+1}$$

$$(i, n+1)f(a_1, ..., a_i-1, f(a_1, ..., a_i-1, x_i, a_{i+1}, ..., a_n), a_{i+1}, ..., a_n) = x_i$$

**Algorithm 2.2 (Generalized algorithm 1)** Let $Q$ be a non-empty finite alphabet and $k$ be a natural number, $u_i, v_i \in Q$, $i \in \{1, ..., k\}$. Define an $n$-ary groupoid $(Q, f)$ which is invertible on $n$-th place. It is clear that groupoid $(Q, (n, n+1)f)$ is defined in a unique way.

Take the fixed elements $l_1^{(n-1)^2} (l_i \in Q)$, which are called leaders.

Let $u_1, u_2, ..., u_k$ be a $k$-tuple of letters from $Q$.

The following encryption procedure is proposed for this sequence:

$$v_1 = f(l_1^{(n-1)}, u_1),$$
$$v_2 = f(l_n^{(2n-2)}, u_2),$$
$$...,$$
$$v_{n-1} = f(l_n^{(n-1)^2}, u_{n-1}),$$
$$v_n = f(v_1^{(n-1)}, u_n),$$
$$v_{n+1} = f(v_2^{n}, u_{n+1}),$$
$$v_{n+2} = f(v_3^{n+1}, u_{n+2}),$$
$$...$$

Therefore we obtain the following ciphertext:

$$v_1, v_2, ..., v_{n-1}, v_n, v_{n+1}...$$

The deciphering algorithm is constructed similarly with binary case:

$$u_1 = (n, n+1)f(l_1^{(n-1)}, v_1),$$
$$u_2 = (n, n+1)f(l_n^{(2n-2)}, v_2),$$
$$...$$
\[ u_{n-1} = (n,n+1)f\left(l_{n^2-3n+3}^{(n-1)2}, v_{n-1}\right), \]
\[ u_n = (n,n+1)f(v_1^{(n-1)}, v_n), \]
\[ u_{n+1} = (n,n+1)f(v_2^n, v_{n+1}), \]
\[ u_{n+2} = (n,n+1)f(v_3^{n+1}, v_{n+2}), \]
\[ \ldots \]

Indeed, for example
\[ (n,n+1)f(v_1^{(n-1)}, v_n) = (n,n+1)f(v_1^{(n-1)}, f(v_1^{n-1}, u_n)) = u_n. \]

3 Attacks with selected ciphertext and plaintext

Consider an attack with text constructed using an n-ary groupoid, which is invertible in the last place (Generalized algorithm 1).

Assume the cryptanalyst has access to the decryption device loaded with the key. He can then construct the following ciphertext:

\[ q_1q_1 \ldots q_1 q_1 \ldots q_1 q_2 \ldots q_1 q_1 \ldots q_1 q_m \ldots q_1 q_m \ldots q_m \]
\[ q_2q_1 \ldots q_1 q_1 \ldots q_2 q_1 \ldots q_2 q_1 \ldots q_2 q_m \ldots q_m \]
\[ q_3q_1 \ldots q_1 q_1 \ldots q_3 q_1 \ldots q_3 q_1 \ldots q_3 q_m \ldots q_m \]

\[ \ldots \]

and enter it into the decryption device.

Thus, for a complete reconstruction of the table of values of the operation \((i,n+1)f\) and hence, the table of values of the operation \(f\), it is sufficient to submit at the input: \((n \cdot m^{n-1} + 1)(m - 1)\) characters to get all the values or \(n \cdot m^{n-1} \cdot (m - 1) + (m - 2)\) characters, when the last value is found by the exception method.

**Example 3.1** Let \((R_3, f)\) be a ternary groupoid, which is defined above the ring \((R_3, +, \cdot)\) of residues modulo 3 and which is invertible in the third place. We define ternary operation \(f\) on set \(R_3\) in the following way:

\[ f(x_1, x_2, x_3) = \alpha x_1 + \beta x_2 + \gamma x_3 = x_4, \text{ where} \]
\[ \alpha 0 = 1, \ \alpha 1 = 1, \ \alpha 2 = 0, \]
\[ \beta 0 = 1, \ \beta 1 = 1, \ \beta 2 = 2, \]
\[ \gamma 0 = 0, \ \gamma 1 = 2, \ \gamma 2 = 0. \]

Inverse operation for \(f(x_1, x_2, x_3) = \alpha x_1 + \beta x_2 + x_3 = x_4\) is:

\[ (3,4)f(x_1, x_2, x_4) = \gamma x_3 = 2 \cdot \alpha x_1 + 2 \cdot \beta x_2 + x_4. \]

We propose the following elements: \(l_1 = 2, l_2 = 1, l_3 = 2, l_4 = 0\) as leader elements.

Enter the following text into the decryption device:
\[
q_1 q_1 q_1 q_1 q_2 q_1 q_3 q_1 q_2 q_1 q_2 q_2 q_1 q_2 q_3 q_1 q_3 q_2 q_1 q_3 q_3 q_3
q_2 q_1 q_1 q_2 q_2 q_1 q_3 q_2 q_2 q_1 q_2 q_2 q_2 q_3 q_3 q_1 q_2 q_3 q_2 q_2 q_3
q_3 q_1
\]
or
000 001 002 010 011 012 020 021 022
100 101 102 110 111 112 120 121 122
20

Table 1. Decryption process

| \(u_1\) | \((3,4)f(l_1, l_2, q_1) = (3,4)f(2,1,0) = 2\) |
| \(u_2\) | \((3,4)f(l_3, l_4, q_1) = (3,4)f(2,0,0) = 2\) |
| \(u_3\) | \((3,4)f(q_1, q_1, q_1) = (3,4)f(0,0,0) = 1 - (1)\) |
| \(u_4\) | \((3,4)f(q_1, q_1, q_1) = (3,4)f(0,0,0) = 1\) |
| \(u_5\) | \((3,4)f(q_1, q_1, q_1) = (3,4)f(0,0,0) = 1\) |
| \(u_6\) | \((3,4)f(q_1, q_1, q_2) = (3,4)f(0,0,1) = 2 - (2)\) |
| \(u_7\) | \((3,4)f(q_1, q_2, q_1) = (3,4)f(0,1,0) = 1 - (3)\) |
| \(u_8\) | \((3,4)f(q_2, q_1, q_1) = (3,4)f(1,0,0) = 1 - (10)\) |
| \(u_9\) | \((3,4)f(q_1, q_1, q_3) = (3,4)f(0,0,2) = 0 - (3)\) |
| \(u_{10}\) | \((3,4)f(q_1, q_3, q_1) = (3,4)f(0,2,0) = 0 - (7)\) |
| \(u_{11}\) | \((3,4)f(q_3, q_1, q_2) = (3,4)f(2,0,1) = 0 - (20)\) |
| \(u_{12}\) | \((3,4)f(q_1, q_2, q_1) = (3,4)f(0,1,0) = 1\) |
| \(u_{13}\) | \((3,4)f(q_2, q_1, q_1) = (3,4)f(1,0,0) = 1\) |
| \(u_{14}\) | \((3,4)f(q_1, q_1, q_2) = (3,4)f(0,0,1) = 2\) |
| \(u_{15}\) | \((3,4)f(q_1, q_2, q_2) = (3,4)f(0,1,1) = 2 - (5)\) |
| \(u_{16}\) | \((3,4)f(q_2, q_2, q_1) = (3,4)f(1,1,0) = 1 - (13)\) |
| \(u_{17}\) | \((3,4)f(q_2, q_1, q_2) = (3,4)f(1,0,1) = 2 - (11)\) |
| \(u_{18}\) | \((3,4)f(q_1, q_2, q_3) = (3,4)f(0,1,2) = 0 - (6)\) |
| \(u_{19}\) | \((3,4)f(q_2, q_3, q_1) = (3,4)f(1,2,0) = 0 - (16)\) |
| \(u_{20}\) | \((3,4)f(q_3, q_1, q_3) = (3,4)f(2,0,2) = 1 - (21)\) |
| \(u_{21}\) | \((3,4)f(q_1, q_3, q_1) = (3,4)f(0,2,0) = 0\) |
| \(u_{22}\) | \((3,4)f(q_3, q_1, q_1) = (3,4)f(2,0,0) = 2 - (19)\) |
| \(u_{23}\) | \((3,4)f(q_1, q_1, q_3) = (3,4)f(0,0,2) = 0\) |
| \(u_{24}\) | \((3,4)f(q_1, q_3, q_2) = (3,4)f(0,2,1) = 1 - (8)\) |
| \(u_{25}\) | \((3,4)f(q_3, q_2, q_1) = (3,4)f(2,1,0) = 2 - (22)\) |
At the output of the descrambler we get the following 56 characters:
Thus, for a complete reconstruction of the table of values of the operation \((3,4)f\), and hence the table of values of the operation \(f\) it is enough to supply 55 characters (without the last one) for the ternary groupoid at the input, or 56 to restore all values.

Knowing the inverse operation value table for \((3,4)f\), operation value table is easily restored \(f\).

To understand the situation with burglary of the decrypted text and the leaders, consider the plaintext of the form: 101202. Using the table of values of function \(f\).

\[
\begin{align*}
  v_1 &= f(l_1, l_2, 1) = ? \\
  v_2 &= f(l_3, l_4, 0) = ? \\
  v_3 &= f(v_1, v_2, u_3) = f(v_1, v_2, 1) = ? \\
  v_4 &= f(v_2, v_3, u_4) = f(v_2, v_3, 2) = ? \\
  v_5 &= f(v_3, v_4, u_5) = f(v_3, v_4, 0) = ? \\
  v_6 &= f(v_4, v_5, u_6) = f(v_4, v_5, 2) = ?
\end{align*}
\]

Analyzing the results obtained using the table of values of the function \(f\), we obtain the following: \(f(*,*,1), f(*,*,0)\) and \(f(*,*,2)\) take any values.

Table 2. Ciphertext values

<table>
<thead>
<tr>
<th>(v_1)</th>
<th>(v_2)</th>
<th>(v_3)</th>
<th>(v_4)</th>
<th>(v_5)</th>
<th>(v_6)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>(f(0,0,1) = 1)</td>
<td>(f(0,1,2) = 2)</td>
<td>(f(1,2,0) = 1)</td>
<td>(f(2,1,2) = 1)</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>(f(0,1,1) = 1)</td>
<td>(f(1,1,2) = 2)</td>
<td>(f(1,2,0) = 1)</td>
<td>(f(2,1,2) = 1)</td>
</tr>
<tr>
<td>0</td>
<td>2</td>
<td>(f(0,2,1) = 2)</td>
<td>(f(2,2,2) = 2)</td>
<td>(f(2,2,0) = 0)</td>
<td>(f(2,0,2) = 1)</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>(f(1,0,1) = 1)</td>
<td>(f(0,1,2) = 2)</td>
<td>(f(1,2,0) = 1)</td>
<td>(f(2,1,2) = 1)</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>(f(1,1,1) = 1)</td>
<td>(f(1,1,2) = 2)</td>
<td>(f(1,2,0) = 1)</td>
<td>(f(2,1,2) = 1)</td>
</tr>
<tr>
<td>1</td>
<td>2</td>
<td>(f(1,2,1) = 2)</td>
<td>(f(2,2,2) = 2)</td>
<td>(f(2,2,0) = 0)</td>
<td>(f(2,0,2) = 1)</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>(f(2,0,1) = 0)</td>
<td>(f(0,0,2) = 2)</td>
<td>(f(0,2,0) = 1)</td>
<td>(f(2,1,2) = 1)</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>(f(2,1,1) = 0)</td>
<td>(f(1,0,2) = 2)</td>
<td>(f(0,2,0) = 1)</td>
<td>(f(2,1,2) = 1)</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>(f(2,2,1) = 1)</td>
<td>(f(2,1,2) = 1)</td>
<td>(f(1,1,0) = 0)</td>
<td>(f(1,0,2) = 2)</td>
</tr>
</tbody>
</table>

We get 9 options for possible decrypted text. Among which the third option is true. The possible values of the ciphertext will be only 9 options, i.e. to determine the true value is not particularly difficult.
The question of identifying leaders in this case loses its relevance. Thus different sets of leaders for a ternary groupoid will be $9^2 = 81$. Essentially, we do not need to determine the exact values of the leaders.

If we enter the following text into the decryption device:

$q_1q_1q_1q_2q_2q_3q_3q_3$
$q_2q_1q_3q_2q_2q_1q_3q_3$
$q_1q_2q_1q_2q_3q_2q_3q_1q_3$
$q_1q_1q_1$

or

000 111 222
100 211 022
010 121 202
00

| $u_1$ | $(3,4)f(1,1, q_1) = (3,4)f(2,1,0) = 2$ |
| $u_2$ | $(3,4)f(1,1, q_1) = (3,4)f(2,0,0) = 2$ |
| $u_3$ | $(3,4)f(1,1, q_1) = (3,4)f(0,0,0) = 1 - (1)$ |
| $u_4$ | $(3,4)f(1,1, q_1) = (3,4)f(0,0,0) = 2 - (2)$ |
| $u_5$ | $(3,4)f(1,1, q_1) = (3,4)f(0,1,1) = 2 - (5)$ |
| $u_6$ | $(3,4)f(1,1, q_1) = (3,4)f(1,1,1) = 2 - (14)$ |
| $u_7$ | $(3,4)f(1,1, q_1) = (3,4)f(1,1,0) = 2 - (15)$ |
| $u_8$ | $(3,4)f(1,1, q_1) = (3,4)f(1,2,2) = 2 - (16)$ |
| $u_9$ | $(3,4)f(1,1, q_1) = (3,4)f(2,2,2) = 0 - (27)$ |
| $u_{10}$ | $(3,4)f(1,1, q_1) = (3,4)f(2,2,2) = 0 - (26)$ |
| $u_{11}$ | $(3,4)f(1,1, q_1) = (3,4)f(2,1,0) = 2 - (22)$ |
| $u_{12}$ | $(3,4)f(1,1, q_1) = (3,4)f(1,0,0) = 1 - (10)$ |
| $u_{13}$ | $(3,4)f(1,1, q_1) = (3,4)f(0,0,2) = 0 - (3)$ |
| $u_{14}$ | $(3,4)f(1,1, q_1) = (3,4)f(0,2,1) = 1 - (8)$ |
| $u_{15}$ | $(3,4)f(1,1, q_1) = (3,4)f(2,1,1) = 0 - (23)$ |
| $u_{16}$ | $(3,4)f(1,1, q_1) = (3,4)f(1,1,0) = 1 - (13)$ |
| $u_{17}$ | $(3,4)f(1,1, q_1) = (3,4)f(1,0,2) = 0 - (12)$ |
| $u_{18}$ | $(3,4)f(1,1, q_1) = (3,4)f(0,2,2) = 2 - (9)$ |
| $u_{19}$ | $(3,4)f(1,1, q_1) = (3,4)f(2,2,0) = 1 - (25)$ |
| $u_{20}$ | $(3,4)f(1,1, q_1) = (3,4)f(2,0,1) = 0 - (20)$ |
\[ u_{21} = (3,4) f(q_1, q_2, q_3) = (3,4) f(0,1,0) = 1 - (4) \]
\[ u_{22} = (3,4) f(q_2, q_1, q_2) = (3,4) f(1,0,1) = 2 - (11) \]
\[ u_{23} = (3,4) f(q_1, q_2, q_3) = (3,4) f(0,1,2) = 0 - (6) \]
\[ u_{24} = (3,4) f(q_2, q_3, q_2) = (3,4) f(1,2,1) = 1 - (17) \]
\[ u_{25} = (3,4) f(q_3, q_2, q_3) = (3,4) f(2,1,2) = 1 - (24) \]
\[ u_{26} = (3,4) f(q_2, q_3, q_1) = (3,4) f(1,2,0) = 0 - (16) \]
\[ u_{27} = (3,4) f(q_3, q_1, q_3) = (3,4) f(2,0,2) = 1 - (21) \]
\[ u_{28} = (3,4) f(q_1, q_3, q_1) = (3,4) f(0,2,0) = 0 - (7) \]
\[ u_{29} = (3,4) f(q_3, q_1, q_1) = (3,4) f(2,0,0) = 2 - (19) - is not obligatory \]

At the output of the descrambler, we get the following 29 characters:
221 222 020
221 010 102
101 201 101
02.

Thus, for a complete reconstruction of the table of values of the operation \( (3,4) f \) it is enough to supply 28 characters (without the last one) for the ternary groupoid at the input, or 29 to restore all values.

So the minimum number of characters in a modified attack will be: \( m^n + (n - 1) \). The main question is the selection of such text for various groupoids.

If we consider the clear text attack for the previous example:
q1q1q1 q1q1q2q3q1q3 q1q2q1 q1q2q2q3q2q3 q1q3q1q3q2q1q3q3
q2q1q1 q2q1q2q2q3q2q2q2q2q3 q2q3q1q2q3q2q3q2q3q3
q3q1q1 q3q1q2q3q1q3q3q2q1 q3q2q2
or
000 001 002 010 011 012 020 021 022
100 101 102 110 111 112 120 121 122
200 201 202 210 211

Table 4. Encryption process

| \( v_1 = f(l_1, l_2, u_1) = f(2,1,0) = 2 \) | \( v_{36} = f(1,0,2) = 2 - (12) \) |
| \( v_2 = f(l_3, l_4, u_2) = f(2,0,0) = 2 \) | \( v_{37} = f(0,2,1) = 2 \) |
| \( v_3 = f(v_1, v_2, u_3) = f(2,2,0) = 0 - (25) \) | \( v_{38} = f(2,2,1) = 1 - (26) \) |
| \( v_4 = f(v_2, v_3, u_4) = f(2,0,0) = 2 - (19) \) | \( v_{39} = f(2,1,0) = 2 - (22) \) |
| \( v_5 = f(v_3, v_4, u_5) = f(0,2,0) = 1 - (7) \) | \( v_{40} = f(1,2,1) = 2 \) |
| $v_6 = f(v_4, v_5, u_6) = f(2,1,1) = 0 - (23)$ | $v_{41} = f(2,2,1) = 1$ |
| $v_7 = f(v_5, v_6, u_7) = f(1,0,0) = 0 - (10)$ | $v_{42} = f(2,1,1) = 0$ |
| $v_8 = f(v_7, v_8, u_8) = f(0,0,0) = 0 - (1)$ | $v_{43} = f(1,0,1) = 1$ |
| $v_9 = f(v_8, v_9, u_9) = f(0,0,2) = 2 - (3)$ | $v_{44} = f(0,1,1) = 1$ |
| $v_{10} = f(v_9, v_{10}, u_{10}) = f(0,2,0) = 1$ | $v_{45} = f(1,1,2) = 2 - (15)$ |
| $v_{11} = f(2,1,1) = 0$ | $v_{46} = f(1,2,1) = 2$ |
| $v_{12} = f(1,0,0) = 0$ | $v_{47} = f(2,2,2) = 2 - (27)$ |
| $v_{13} = f(0,0,0) = 0$ | $v_{48} = f(2,2,0) = 0$ |
| $v_{14} = f(0,0,1) = 1 - (2)$ | $v_{49} = f(2,0,1) = 0$ |
| $v_{15} = f(0,1,1) = 1 - (5)$ | $v_{50} = f(0,0,2) = 2$ |
| $v_{16} = f(1,1,0) = 0 - (13)$ | $v_{51} = f(0,2,1) = 2$ |
| $v_{17} = f(1,0,1) = 1 - (11)$ | $v_{52} = f(2,2,1) = 1$ |
| $v_{18} = f(0,1,2) = 2 - (6)$ | $v_{53} = f(2,1,2) = 1$ |
| $v_{19} = f(1,2,0) = 1 - (16)$ | $v_{54} = f(1,1,2) = 2$ |
| $v_{20} = f(2,1,2) = 1 - (24)$ | $v_{55} = f(1,2,2) = 0 - (18)$ |
| $v_{21} = f(1,1,0) = 0$ | $v_{56} = f(2,0,0) = 2$ |
| $v_{22} = f(1,0,0) = 0$ | $v_{57} = f(0,2,0) = 1$ |
| $v_{23} = f(0,0,2) = 2$ | $v_{58} = f(2,1,2) = 1$ |
| $v_{24} = f(0,2,1) = 2 - (8)$ | $v_{59} = f(1,1,0) = 0$ |
| $v_{25} = f(2,2,0) = 0$ | $v_{60} = f(1,0,1) = 1$ |
| $v_{26} = f(2,0,2) = 1 - (21)$ | $v_{61} = f(0,1,2) = 2$ |
| $v_{27} = f(0,1,2) = 2$ | $v_{62} = f(1,2,0) = 1$ |
| $v_{28} = f(1,2,1) = 2 - (17)$ | $v_{63} = f(2,1,2) = 1$ |
| $v_{29} = f(2,2,0) = 0$ | $v_{64} = f(1,1,2) = 2$ |
| $v_{30} = f(2,0,0) = 2$ | $v_{65} = f(1,2,1) = 2$ |
| $v_{31} = f(0,2,1) = 2$ | $v_{66} = f(2,2,0) = 0$ |
| $v_{32} = f(2,2,0) = 0$ | $v_{67} = f(2,0,2) = 1$ |
| $v_{33} = f(2,0,1) = 0 - (20)$ | $v_{68} = f(0,1,1) = 1$ |
| $v_{34} = f(0,0,1) = 1$ | $v_{69} = f(1,1,1) = 1 - (14)$ is the penultimate value. |

At the output of the encryption device, we get the following 69 characters:
To restore the table of values of the operation $f$, the 69 symbols will suffice, and the latter will be found by the method of elimination. And the last value does not come out even after a full run.

Knowing the table of operation values for $f$ easily restores the table of values of the inverse operation $(3,4)f$.

Take some encrypted text and try to crack it. For example, we have the following ciphertext: 201120. Then we have:

$u_1 = (3,4)f(l_1, l_2, v_1) = (3,4)f(l_1, l_2, 2) = ?$
$u_2 = (3,4)f(l_3, l_4, v_2) = (3,4)f(l_3, l_4, 0) = ?$
$u_3 = (3,4)f(v_1, v_2, v_3) = (3,4)f(2,0,1) = 0$
$u_4 = (3,4)f(v_2, v_3, v_4) = (3,4)f(0,1,1) = 2$
$u_5 = (3,4)f(v_3, v_4, v_5) = (3,4)f(1,1,2) = 0$
$u_6 = (3,4)f(v_4, v_5, v_6) = (3,4)f(1,2,0) = 0$

Analyzing the results obtained using the table of values of the function $f$, we obtain the following: $f(*,*,2), f(*,*,0)$ and $f(*,*,1)$ take any values.

Table 5. Ciphertext values (results)

<table>
<thead>
<tr>
<th>$u_1$</th>
<th>$u_2$</th>
<th>$u_3$</th>
<th>$u_4$</th>
<th>$u_5$</th>
<th>$u_6$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>2</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>2</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>0</td>
<td>2</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>2</td>
<td>0</td>
<td>2</td>
<td></td>
</tr>
</tbody>
</table>

Among which the sixth option is correct. The possible values of the ciphertext will be only 9 options, i.e. to determine the true value is not particularly difficult.
For an \( n \)-ary groupoid in plaintext of length \( k \), the first \((n - 1)\) characters are not cracked. The rest are unequivocally.

The possible number of variants of open and encrypted texts in both cases is the same. Will it always be so, it is necessary to check.

**Example 3.2.** Consider the \( 4 \)-ary groupoid \((R_3, f)\), \( R_3 = \{0,1,2\} \), which is defined above the ring \((R_3, +, \cdot)\) residue classes modulo 3 and which is invertible in fourth place. \( 4 \)-ary operation \( f \) on set \( R_3 \) is defined as:

\[
f(x_1, x_2, x_3, x_4) = \alpha x_1 + \beta x_2 + \gamma x_3 + x_4 = x_5, \text{ where} \\
\alpha 0 = 2, \alpha 1 = 2, \alpha 2 = 0, \\
\beta 0 = 1, \beta 1 = 2, \beta 2 = 0, \\
\gamma 0 = 1, \gamma 1 = 1, \gamma 2 = 2.
\]

Inverse operation for \( f(x_1, x_2, x_3, x_4) = \alpha x_1 + \beta x_2 + \gamma x_3 + x_4 = x_5 \) has the appearance:

\[
(4,5)f(x_1, x_2, x_3, x_5) = x_4 = 2 \cdot \alpha x_1 + 2 \cdot \beta x_2 + 2 \cdot \gamma x_3 + x_5.
\]

We used the following elements as leader elements:

\[
l_1 = 0, l_2 = 1, l_3 = 2, l_4 = 1, l_5 = 0, l_6 = 2, l_7 = 1, l_8 = 1, l_9 = 0.
\]

Enter the following text into the decryption device:

\[
0000 0001 0002 0010 0011 0012 0020 0021 0022 \\
0100 0101 0102 0110 0111 0112 0120 0121 0122 \\
0200 0201 0202 0210 0211 0212 0220 0221 0222 \\
1000 1001 1002 1010 1011 1012 1020 1021 1022 \\
1100 1101 1102 1110 1111 1112 1120 1121 1122 \\
1200 1201 1202 1210 1211 1212 1220 1221 1222 \\
20
\]

Table 6. Decryption process

<table>
<thead>
<tr>
<th>( u_1 = (4,5)f(l_1, l_2, l_3, 0) = 2 )</th>
<th>( u_{110} = (4,5)f(2,2,1,0) = 2 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( u_2 = (4,5)f(l_4, l_5, l_6, 0) = 1 )</td>
<td>( u_{111} = (4,5)f(2,1,0,0) = 0 )</td>
</tr>
<tr>
<td>( u_3 = (4,5)f(l_7, l_8, l_9, 0) = 1 )</td>
<td>( u_{112} = (4,5)f(1,0,0,0) = 2 )</td>
</tr>
<tr>
<td>( u_4 = (4,5)f(0,0,0,0) = 1 - (1) )</td>
<td>( u_{113} = (4,5)f(0,0,0,1) = 2 )</td>
</tr>
<tr>
<td>( u_5 = (4,5)f(0,0,0,0) = 1 )</td>
<td>( u_{114} = (4,5)f(0,0,1,0) = 1 )</td>
</tr>
<tr>
<td>( u_6 = (4,5)f(0,0,0,0) = 1 )</td>
<td>( u_{115} = (4,5)f(0,1,0,0) = 0 )</td>
</tr>
<tr>
<td>( u_7 = (4,5)f(0,0,0,0) = 1 )</td>
<td>( u_{116} = (4,5)f(1,0,0,1) = 0 )</td>
</tr>
<tr>
<td>( u_8 = (4,5)f(0,0,0,1) = 2 - (2) )</td>
<td>( u_{117} = (4,5)f(0,0,1,1) = 2 )</td>
</tr>
<tr>
<td>$u_9 = (4,5)f(0,0,1,0) = 1 - (4)$</td>
<td>$u_{118} = (4,5)f(0,1,1,0) = 0$</td>
</tr>
<tr>
<td>---</td>
<td>---</td>
</tr>
<tr>
<td>$u_{10} = (4,5)f(0,1,0,0) = 0 - (10)$</td>
<td>$u_{119} = (4,5)f(1,1,0,0) = 1$</td>
</tr>
<tr>
<td>$u_{11} = (4,5)f(1,0,0,0) = 2 - (28)$</td>
<td>$u_{120} = (4,5)f(1,0,0,2) = 1$</td>
</tr>
<tr>
<td>$u_{12} = (4,5)f(0,0,0,2) = 0 - (3)$</td>
<td>$u_{121} = (4,5)f(0,0,2,1) = 1$</td>
</tr>
<tr>
<td>$u_{13} = (4,5)f(0,0,2,0) = 0 - (7)$</td>
<td>$u_{122} = (4,5)f(0,2,1,0) = 2$</td>
</tr>
<tr>
<td>$u_{14} = (4,5)f(0,2,0,0) = 2 - (19)$</td>
<td>$u_{123} = (4,5)f(2,1,0,1) = 1$</td>
</tr>
<tr>
<td>$u_{15} = (4,5)f(2,0,0,1) = 2 - (56)$</td>
<td>$u_{124} = (4,5)f(1,0,1,0) = 2$</td>
</tr>
<tr>
<td>$u_{16} = (4,5)f(0,0,1,0) = 1$</td>
<td>$u_{125} = (4,5)f(0,1,0,1) = 1$</td>
</tr>
<tr>
<td>$u_{17} = (4,5)f(0,1,0,0) = 0$</td>
<td>$u_{126} = (4,5)f(1,0,1,0) = 2$</td>
</tr>
<tr>
<td>$u_{18} = (4,5)f(1,0,0,0) = 2$</td>
<td>$u_{127} = (4,5)f(0,1,0,1) = 1$</td>
</tr>
<tr>
<td>$u_{19} = (4,5)f(0,0,0,1) = 2$</td>
<td>$u_{128} = (4,5)f(1,0,1,1) = 0$</td>
</tr>
<tr>
<td>$u_{20} = (4,5)f(0,0,1,1) = 2 - (5)$</td>
<td>$u_{129} = (4,5)f(0,1,1,1) = 1$</td>
</tr>
<tr>
<td>$u_{21} = (4,5)f(0,1,1,0) = 0 - (13)$</td>
<td>$u_{130} = (4,5)f(1,1,1,0) = 1$</td>
</tr>
<tr>
<td>$u_{22} = (4,5)f(1,1,0,0) = 1 - (37)$</td>
<td>$u_{131} = (4,5)f(1,1,0,1) = 2$</td>
</tr>
<tr>
<td>$u_{23} = (4,5)f(1,0,0,1) = 0 - (29)$</td>
<td>$u_{132} = (4,5)f(1,0,1,2) = 1$</td>
</tr>
<tr>
<td>$u_{24} = (4,5)f(0,0,1,2) = 0 - (6)$</td>
<td>$u_{133} = (4,5)f(0,1,2,1) = 0$</td>
</tr>
<tr>
<td>$u_{25} = (4,5)f(0,1,2,0) = 2 - (16)$</td>
<td>$u_{134} = (4,5)f(1,2,1,0) = 0$</td>
</tr>
<tr>
<td>$u_{26} = (4,5)f(1,2,0,0) = 0 - (46)$</td>
<td>$u_{135} = (4,5)f(2,1,0,2) = 2$</td>
</tr>
<tr>
<td>$u_{27} = (4,5)f(2,0,0,2) = 0 - (57)$</td>
<td>$u_{136} = (4,5)f(1,0,2,0) = 1$</td>
</tr>
<tr>
<td>$u_{28} = (4,5)f(0,0,2,0) = 0$</td>
<td>$u_{137} = (4,5)f(0,2,0,1) = 0$</td>
</tr>
<tr>
<td>$u_{29} = (4,5)f(0,2,0,0) = 2$</td>
<td>$u_{138} = (4,5)f(2,0,1,0) = 1$</td>
</tr>
<tr>
<td>$u_{30} = (4,5)f(2,0,0,0) = 1 - (55)$</td>
<td>$u_{139} = (4,5)f(0,1,0,2) = 2$</td>
</tr>
<tr>
<td>$u_{31} = (4,5)f(0,0,0,2) = 0$</td>
<td>$u_{140} = (4,5)f(1,0,2,1) = 2$</td>
</tr>
<tr>
<td>$u_{32} = (4,5)f(0,0,2,1) = 1 - (8)$</td>
<td>$u_{141} = (4,5)f(0,2,1,1) = 0$</td>
</tr>
<tr>
<td>$u_{33} = (4,5)f(0,2,1,0) = 2 - (22)$</td>
<td>$u_{142} = (4,5)f(2,1,1,0) = 0$</td>
</tr>
<tr>
<td>$u_{34} = (4,5)f(2,1,0,0) = 0 - (64)$</td>
<td>$u_{143} = (4,5)f(1,1,0,2) = 0$</td>
</tr>
<tr>
<td>$u_{35} = (4,5)f(1,0,0,2) = 1 - (30)$</td>
<td>$u_{144} = (4,5)f(1,0,2,2) = 0$</td>
</tr>
<tr>
<td>$u_{36} = (4,5)f(0,0,2,2) = 2 - (9)$</td>
<td>$u_{145} = (4,5)f(0,2,2,1) = 2$</td>
</tr>
<tr>
<td>$u_{37} = (4,5)f(0,2,2,0) = 1 - (25)$</td>
<td>$u_{146} = (4,5)f(2,2,1,1) = 0 - (77)$</td>
</tr>
<tr>
<td>$u_{38} = (4,5)f(2,2,0,1) = 0 - (74)$</td>
<td>$u_{147} = (4,5)f(2,1,1,0) = 0$</td>
</tr>
<tr>
<td>$u_{39} = (4,5)f(2,0,1,0) = 1 - (58)$</td>
<td>$u_{148} = (4,5)f(1,1,0,0) = 1$</td>
</tr>
<tr>
<td>$u_{40} = (4,5)f(0,1,0,0) = 0$</td>
<td>$u_{149} = (4,5)f(1,0,0,1) = 0$</td>
</tr>
<tr>
<td>$u_{41} = (4,5)f(1,0,0,0) = 2$</td>
<td>$u_{150} = (4,5)f(0,0,1,1) = 2$</td>
</tr>
<tr>
<td>( u_{42} = (4,5) f(0,0,0,1) = 2 )</td>
<td>( u_{151} = (4,5) f(0,1,1,0) = 0 )</td>
</tr>
<tr>
<td>---</td>
<td>---</td>
</tr>
<tr>
<td>( u_{43} = (4,5) f(0,0,1,0) = 1 )</td>
<td>( u_{152} = (4,5) f(1,1,0,1) = 2 )</td>
</tr>
<tr>
<td>( u_{44} = (4,5) f(0,1,0,1) = 1 - (11) )</td>
<td>( u_{153} = (4,5) f(1,0,1,1) = 0 )</td>
</tr>
<tr>
<td>( u_{45} = (4,5) f(1,0,1,0) = 2 - (31) )</td>
<td>( u_{154} = (4,5) f(0,1,1,1) = 1 )</td>
</tr>
<tr>
<td>( u_{46} = (4,5) f(0,1,0,1) = 1 )</td>
<td>( u_{155} = (4,5) f(1,1,1,0) = 1 )</td>
</tr>
<tr>
<td>( u_{47} = (4,5) f(1,0,1,0) = 2 )</td>
<td>( u_{156} = (4,5) f(1,1,0,2) = 0 )</td>
</tr>
<tr>
<td>( u_{48} = (4,5) f(0,1,0,2) = 2 - (12) )</td>
<td>( u_{157} = (4,5) f(1,0,2,1) = 2 )</td>
</tr>
<tr>
<td>( u_{49} = (4,5) f(1,0,2,0) = 1 - (34) )</td>
<td>( u_{158} = (4,5) f(0,2,1,1) = 0 )</td>
</tr>
<tr>
<td>( u_{50} = (4,5) f(0,2,0,1) = 0 - (20) )</td>
<td>( u_{159} = (4,5) f(2,1,1,1) = 1 - (68) )</td>
</tr>
<tr>
<td>( u_{51} = (4,5) f(2,0,1,1) = 2 - (59) )</td>
<td>( u_{160} = (4,5) f(1,1,1,0) = 1 )</td>
</tr>
<tr>
<td>( u_{52} = (4,5) f(0,1,1,0) = 0 )</td>
<td>( u_{161} = (4,5) f(1,1,0,1) = 2 )</td>
</tr>
<tr>
<td>( u_{53} = (4,5) f(1,1,0,0) = 1 )</td>
<td>( u_{162} = (4,5) f(1,0,1,1) = 0 )</td>
</tr>
<tr>
<td>( u_{54} = (4,5) f(1,0,0,1) = 0 )</td>
<td>( u_{163} = (4,5) f(0,1,1,1) = 1 )</td>
</tr>
<tr>
<td>( u_{55} = (4,5) f(0,0,1,1) = 2 )</td>
<td>( u_{164} = (4,5) f(1,1,1,1) = 2 - (41) )</td>
</tr>
<tr>
<td>( u_{56} = (4,5) f(0,1,1,1) = 1 - (14) )</td>
<td>( u_{165} = (4,5) f(1,1,1,1) = 2 )</td>
</tr>
<tr>
<td>( u_{57} = (4,5) f(1,1,1,0) = 1 - (40) )</td>
<td>( u_{166} = (4,5) f(1,1,1,1) = 2 )</td>
</tr>
<tr>
<td>( u_{58} = (4,5) f(1,1,0,1) = 2 - (38) )</td>
<td>( u_{167} = (4,5) f(1,1,1,1) = 2 )</td>
</tr>
<tr>
<td>( u_{59} = (4,5) f(1,0,1,1) = 0 - (32) )</td>
<td>( u_{168} = (4,5) f(1,1,1,2) = 0 - (42) )</td>
</tr>
<tr>
<td>( u_{60} = (4,5) f(0,1,1,2) = 2 - (15) )</td>
<td>( u_{169} = (4,5) f(1,1,2,1) = 1 - (44) )</td>
</tr>
<tr>
<td>( u_{61} = (4,5) f(1,1,2,0) = 0 - (43) )</td>
<td>( u_{170} = (4,5) f(1,2,1,1) = 1 - (50) )</td>
</tr>
<tr>
<td>( u_{62} = (4,5) f(1,2,0,1) = 1 - (47) )</td>
<td>( u_{171} = (4,5) f(2,1,1,2) = 2 - (69) )</td>
</tr>
<tr>
<td>( u_{63} = (4,5) f(2,0,1,2) = 0 - (60) )</td>
<td>( u_{172} = (4,5) f(1,1,2,0) = 0 )</td>
</tr>
<tr>
<td>( u_{64} = (4,5) f(0,1,2,0) = 2 )</td>
<td>( u_{173} = (4,5) f(1,2,0,1) = 1 )</td>
</tr>
<tr>
<td>( u_{65} = (4,5) f(1,2,0,0) = 0 )</td>
<td>( u_{174} = (4,5) f(2,0,1,1) = 2 )</td>
</tr>
<tr>
<td>( u_{66} = (4,5) f(2,0,0,1) = 2 )</td>
<td>( u_{175} = (4,5) f(0,1,1,2) = 2 )</td>
</tr>
<tr>
<td>( u_{67} = (4,5) f(0,0,1,2) = 0 )</td>
<td>( u_{176} = (4,5) f(1,1,2,1) = 1 )</td>
</tr>
<tr>
<td>( u_{68} = (4,5) f(0,1,2,1) = 0 - (17) )</td>
<td>( u_{177} = (4,5) f(1,2,1,1) = 1 )</td>
</tr>
<tr>
<td>( u_{69} = (4,5) f(1,2,1,0) = 0 - (49) )</td>
<td>( u_{178} = (4,5) f(2,1,1,1) = 1 )</td>
</tr>
<tr>
<td>( u_{70} = (4,5) f(2,1,0,1) = 1 - (65) )</td>
<td>( u_{179} = (4,5) f(1,1,1,2) = 0 )</td>
</tr>
<tr>
<td>( u_{71} = (4,5) f(1,0,1,2) = 1 - (33) )</td>
<td>( u_{180} = (4,5) f(1,1,2,2) = 2 - (45) )</td>
</tr>
<tr>
<td>( u_{72} = (4,5) f(0,1,2,2) = 1 - (18) )</td>
<td>( u_{181} = (4,5) f(1,2,2,1) = 0 - (53) )</td>
</tr>
<tr>
<td>( u_{73} = (4,5) f(1,2,2,0) = 2 - (52) )</td>
<td>( u_{182} = (4,5) f(2,2,1,2) = 1 - (78) )</td>
</tr>
<tr>
<td>( u_{74} = (4,5) f(2,2,0,2) = 1 - (75) )</td>
<td>( u_{183} = (4,5) f(2,1,2,0) = 2 )</td>
</tr>
<tr>
<td>( u_{75} )</td>
<td>( u_{76} )</td>
</tr>
<tr>
<td>( (4,5) f(2,0,2,0) = 0 - (61) )</td>
<td>( (4,5) f(0,2,0,0) = 2 )</td>
</tr>
</tbody>
</table>
\[ u_{108} = (4,5) f(0,2,2,2) = 0 - (27) \]
\[ u_{109} = (4,5) f(2,2,2,1) = 2 - (80) \]
\[ u_{217} = (4,5) f(2,2,2,2) = 0 - (81) \]
\[ u_{218} = (4,5) f(2,2,2,0) = 1 - (79) \text{ is not obligatory} \]

At the output of the descrambler we get the following 218 characters:
2111 1112 1020 0221 0222 0100 2000 2101 2012  
1010 2211 2122 1020 1021 1202 0102 0200 0111  
2102 1000 1211 0112 0110 0021 2221 2022 2200  
2202 2100 2011 1212 1210 1121 0021 0122 0000  
2001 0202 0110 2011 2012 2220 1120 1221 1102  
0120 2021 2202 1100 1101 1012 0212 0010 0221  
01.

Thus, for a complete reconstruction of the operation value table \((4,5)f\) hence, the tables of the values of the operation \(f\) are sufficient to supply 218 (or 217) characters for the 4-ary groupoid at the input.

For an \(n\)-ary groupoid, the required number of characters is:
\((n \cdot n^{n-1} + 1)(m - 1)\) characters to get all the values or 
\(n \cdot n^{n-1} \cdot (m - 1) + (m - 2)\), when the last value is found by the exception method.

As for an attack using plaintext, even a full run of all 324 characters is not enough to restore all the values of the function. Only 78 of 81 values are restored.

4 Conclusion

In this article, we looked at some types of attacks on the Markovski cipher with the help of open and encrypted texts.

Thus, for a complete reconstruction of the table of values of the operation \((i,n+1)f\) and hence, the table of values of the operation \(f\), it is sufficient to submit at the input: \((n \cdot n^{n-1} + 1)(m - 1)\) characters to get all the values.

As for the plaintext attack, it was possible to establish the lower limit value of the necessary characters to restore the table of the values of function \(f\). But the question remains what kind of text to give at the input of the encrypting device so as not to exceed the received limit of characters, and is it always will it be possible?
We conduct a comparative analysis, identified positive and negative points in these attacks. In the future, we plan to continue attacks on the cipher built with the help of generalized Markovski algorithms.

References

Nadezhda Malyutina
Department of Mathematics,
State University Dimitrie Cantemir,
Academiei str. 3/2, MD-2028 Kishinev, Moldova.
Email: 231003.Bab.Nadezhda@mail.ru
On Weak Unpredictability of Legendre Sequences

Victor Pescaru

Abstract

Cryptographic number generators are primitives of crucial importance to cryptography and information security as many cryptographic applications require random integers (key generation, nonces, salts etc.). Two such generators output binary sequences by computing the Legendre and Jacobi symbols of strictly increasing sequences of integers. It has been shown that weakly unpredictable Legendre sequences lead to strong unpredictable Jacobi sequences. However, whether Legendre sequences are weakly unpredictable or not has remained an open question. In this paper we give a positive answer to this question and show that the Legendre sequences are weakly unpredictable.

Keywords: Legendre symbol, quadratic residue, unpredictability, pseudo-random generator

1 Introduction

A pseudo-random number generator (PRNG), also known as a deterministic random bit generator (DRBG), is an algorithm which takes as input a security parameter $\lambda$ and a seed and produces an output a sequence of bits or integers whose distribution is indistinguishable from the random uniform distribution. In cryptography, indistinguishability is defined by means of probabilistic polynomial algorithms (tests), where the unpredictability tests play a crucial role [1].

In cryptography, PRNG candidates typically base their security on hard problems such as factorization or discreet logarithm [1]. Damgård
has proposed another method to obtain pseudo-random binary sequences, namely by using Legendre and Jacobi sequences. These are simply obtained by computing the Legendre or Jacobi symbol of a sequence of integers, usually taken in increasing order [2]. Clearly, the security of such sequences depend on the distribution of Legendre and Jacobi symbols, a problem that still needs deep research. It was shown in [2] that weakly unpredictable Legendre sequences may be used to define strongly unpredictable Jacobi sequences. However, the question of whether Legendre sequences are weakly unpredictable has left open.

In this paper we prove that Legendre sequences are weakly unpredictable and, thus we give a positive answer to the open problem in [2]. The main idea is to use an estimate on the distribution of Jacobi patterns of polynomial length that has been developed in [4].

The paper is structured into four sections, the first one being the introductory section. We then recall basic definitions, notations, and results in the second section. Our main result is presented in the third section. We conclude the paper in the last section.

2 Jacobi Sequences

We recall in this section a few concepts on number theory [3], and then we introduce Jacobi sequences as they were defined in [2].

Let \( p \) be an odd prime number. An integer \( a \) is a quadratic residue modulo \( p \) if it is congruent to a perfect square modulo \( p \), and is a quadratic non-residue modulo \( p \) otherwise. The Legendre symbol is a function of \( a \) and \( p \) defined as:

\[
\left( \frac{a}{p} \right) = \begin{cases} 
1, & \text{if } a \text{ is a quadratic residue modulo } p \text{ and } a \not\equiv 0 \mod p \\
-1, & \text{if } a \text{ is a quadratic non-residue modulo } p \\
0, & \text{if } a \equiv 0 \mod p
\end{cases}
\]

For an integer \( a \) and a positive odd integer \( n \), the Jacobi symbol the product of the Legendre symbols corresponding to the prime factors of \( n \):

\[
\left( \frac{a}{n} \right) = \left( \frac{a}{p_1} \right)^{\alpha_1} \left( \frac{a}{p_2} \right)^{\alpha_2} \cdots \left( \frac{a}{p_k} \right)^{\alpha_k}
\]
On Weak Unpredictability of Legendre Sequences

where \( n = p_1^{\alpha_1} p_2^{\alpha_2} \cdots p_k^{\alpha_k} \).

Given a security parameter \( \lambda \), a \( \lambda \)-bit positive integer \( n \), an integer \( a \in \mathbb{Z}_n \), and a positive integer \( \ell \) of polynomial size in \( \lambda \), define an \((n, a, \ell)\)-Jacobi sequence \([2]\) as being the sequence of Jacobi symbols

\[
\left( \frac{a}{n} \right), \left( \frac{a+1}{n} \right), \ldots, \left( \frac{a+\ell}{n} \right)
\]

When \( n \) is a prime, this is also called a Legendre sequence. However, for the sake of simplicity, we will only use the terminology of Jacobi sequence. The integer \( a \) is the root or starting point, \( n \) is the modulus, and \( \ell \) is the length of the sequence.

An Jacobi generator is a deterministic polynomial-time algorithm \( G \) that, when seeded by \( \lambda, n, a, \) and \( \ell \) as above, generates a Jacobi sequence of length \( \ell \) with the root \( a \).

One of the main questions on such generators is about their pseudo-randomness. This clearly can be reduced to the prediction of the next element of a sequence of polynomial length. In other words, given a sequence as above, where \( \ell \) is polynomial in the security parameter \( \lambda \), the question is to predict the next symbol \( \left( \frac{a+\ell+1}{n} \right) \).

Damgård introduced two notions of unpredictability:

1. A generator \( G \) is strongly unpredictable if for any polynomial \( P \), any probabilistic circuit \( C \), there exists \( \lambda_0 \) such that

\[
P(C(G(a)_1, \ldots, G(a)_{i-1}) = G(a)_i) < \frac{1}{2} + \frac{1}{P(\lambda)},
\]

for all \( \lambda \geq \lambda_0 \) and all \( i > 1 \);

2. A generator \( G \) is weakly unpredictable if for any polynomial \( P \), any probabilistic circuit \( C \), there exists \( \lambda_0 \) such that

\[
P(C(G(a)_1, \ldots, G(a)_{i-1}) = G(a)_i) < 1 - \frac{1}{P(\lambda)},
\]

for all \( \lambda \geq \lambda_0 \) and all \( i > 1 \).
Given a polynomial $Q(\lambda)$ and $a = (a_1, \ldots, a_{Q(\lambda)})$, define $G^Q$ as being

$$G^Q(a) = G(a_1) \oplus \cdots \oplus G(a_{Q(\lambda)})$$

(the XOR is component-wise computed). Now, one can prove that $G^{\lambda^2}$ is a strongly unpredictable generator if $G$ is weakly unpredictable. Moreover, if the Legendre generator is weakly unpredictable then the Jacobi generator is strongly unpredictable [2].

The following problem was left open in [2].

Open problem 1. Is the Legendre generator weakly unpredictable?

3 Legendre Generators are Weakly Unpredictable

In this section we give a positive answer to the open problem mentioned in the previous section. To this, we need to recall an important result obtained in [4]. Let $p$ be a prime, $a_1, \ldots, a_t \in \mathbb{Z}_p$ be pairwise distinct integers, and $L(i) \in \{-1, 1\}$ for all $1 \leq i \leq t$. It was shown in [4] that

$$\left| P \left( \left( \frac{x+a_1}{p} \right), \ldots, \left( \frac{x+a_t}{p} \right) \right) = L \left| x \leftarrow \mathbb{Z}_p \right. \right. \right. \left. \left. - \frac{1}{2^t} \right| < \frac{t(3 + \sqrt{p})}{p} \right.$$ 

This says that the probability that the Legendre sequence

$$\left( \left( \frac{x+a_1}{p} \right), \ldots, \left( \frac{x+a_t}{p} \right) \right)$$

match a given sequence $L$ differs from that of a fair coin by no more than

$$\frac{t(3 + \sqrt{p})}{p}$$

Now we are ready to prove our main result.

Theorem 3.1. Legendre sequences are weakly unpredictable.
On Weak Unpredictability of Legendre Sequences

Proof. Let $\lambda$ be a security parameter, $p$ be a prime of size $\lambda$, and $\ell$ be an integer of polynomial size in $\lambda$. Denote by $X^i$ the random variable that outputs
\[
\left(\left(\frac{a}{p}\right), \left(\frac{a+1}{p}\right), \ldots, \left(\frac{a+i}{p}\right)\right)
\]
on input $a \leftarrow \mathbb{Z}_p$. Then,
\[
P(X^{\ell+1}(a)) = P(X^{\ell+1}(a)|X^\ell(a)) \cdot P(X^\ell(a))
\]
It is straightforward to see that the unpredictability problem is reduced to the computation of $P(X^{\ell+1}(a)|X^\ell(a))$. Therefore, using the result mentioned above we have
\[
P(X^{\ell+1}(a)|X^\ell(a)) = \frac{P(X^{\ell+1}(a))}{P(X^\ell(a))}
\]
\[
< \frac{1}{2^{\ell+1}} + \frac{(\ell+1)(3+\sqrt{p})}{p}
\]
\[
< \frac{1}{2^{\ell}} - \frac{\ell(3+\sqrt{p})}{p}
\]
\[
< \frac{1}{2^{\ell+1}} + \frac{(\ell+1)(3+\sqrt{p})}{p}
\]
\[
= \frac{p + 2^{\ell+1}(\ell + 1)(3 + \sqrt{p})}{2p - 2^{\ell+1}(\ell + 1)(3 + \sqrt{p})}
\]
\[
= 1 - \frac{p - 2^{\ell+2}(\ell + 1)(3 + \sqrt{p})}{2p - 2^{\ell+1}(\ell + 1)(3 + \sqrt{p})}
\]
\[
= 1 - \frac{1}{\frac{3p}{2} + \frac{1}{2p - 2^{\ell+2}(\ell + 1)(3 + \sqrt{p})}}
\]
So, to prove that
\[
P(X^{\ell+1}(a)|X^\ell(a)) < \frac{1}{2} - \frac{1}{P(\ell)}
\]
107
can be reduced to showing
\[
\frac{1}{2} + \frac{3p}{2(p - 2^\ell + 2(\ell + 1)(3 + \sqrt{p}))} < P(\ell).
\]

However,
\[
\lim_{\ell \to \infty} \left( \frac{1}{2} + \frac{3p}{2(p - 2^\ell + 2(\ell + 1)(3 + \sqrt{p}))} \right) = \frac{1}{2},
\]
while \( \lim_{\ell \to \infty} P(\ell) = \infty \). So, it follows that it there exists \( \ell_0 \) such that
\[
\frac{1}{2} + \frac{3p}{2(p - 2^\ell + 2(\ell + 1)(3 + \sqrt{p}))} < P(\ell),
\]
for all \( \ell > \ell_0 \).

Directly from Theorem 3.1 we get the following results.

**Corollary 3.1.** Legendre generators are weakly unpredictable.

**Corollary 3.2.** Jacobi generators are strongly unpredictable.

*Proof.* From Corollary 3.1 above and Corollary 3.1 in [2].

## 4 Conclusions

We have shown in this paper that the Legendre generator as defined in [2] is weakly unpredictable. This proves that one can define strong unpredictable Jacobi generators.

**Acknowledgement**

We would like to thank Prof.dr. Ferucio Laurentţiu Ţiplea for suggesting this problem and for useful discussions on the topic.
References


Victor Pescaru
Department of Computer Science, “Al.I.Cuza” University of Iași
Email: {victor.pescaruinfo@gmail.com}
On Formalization Styles and SAD System

Alexander Lyaletski, Alexandre Lyaletsy

Abstract

A short description of styles of mathematical knowledge formalization is given. The most famous automated reasoning systems including the English SAD system are classified w.r.t. them. Specific features of the English SAD system are described. Some of possible ways of its further development are pointed out.

Keywords: style of mathematical knowledge formalization, automated reasoning system, English SAD system.

1 Introduction

Investigations on artificial intelligence concern not only the creation of devices that fully or partially simulate the human physical activity, they also touch upon questions and problems relating to the ability of a human to make reasoning in the environment of formalized languages suitable both for a user and computer.

In the early 1970’s, Academician V.M. Glushkov proposed a program for automated processing of mathematical texts called Evidence Algorithm (EA) [1]. This program was focused on computer support for human mathematical activities both in academic studies and teaching of mathematical disciplines.

According to EA, the kernel of a logical search engine should be a procedure for establishing the obviousness of (provable or verifiable) statement in terms of a certain deductive mechanism.

Such a procedure should be strengthened by additional means: to have the opportunity to search for auxiliary statements or any other relevant information, to apply algebraic transformations (or, in modern
terminology, systems of computer algebra), to apply methods used by a man, etc. All these means should be able to exchange by data with the help of a certain formalized language.

The following requirements were formulated to the language. Its syntax and semantics should be formalized. It must admit formulation of axioms of a theory under consideration, theorems and their proofs to ensure the integrity of texts and definitions. In order for the language thesaurus to be replenished, it must be separated from the language grammar. In addition, such a language should be close to the language of usual mathematical publications for providing a user with the convenience of creating and processing a text in a interactive mode. It should also allow formulating tasks that are not directly connected with inference search.

Deductive methods should be developed simultaneously with the language according to the following requirements: the syntactic form of an initial task should be preserved; deductive constructions should be made in the signature of an initial theory; inference search should be goal-oriented; processing equations (solving equations) must be separated from deductive processes.

Thus, the EA program is oriented on the construction of various components necessary for the automated processing of mathematical texts and based on making the simultaneous research in the development of formalized languages for the writing of mathematical texts in the form most favorable for a user, in the direction of evolutionary development of the notion of machine (deductive and inductive) step of proving, in the direction of creating the informational environment for EA, which affects the obviousness of the machine-made step of a proof, and in the direction of constructing interactive means for processing mathematical texts.

The study on EA can be divided into two main historical stages: the first one — from 1970 to 1990, when the Russian SAD system with the formal Russian-style language was implemented on the ES-line computers, and the second one — from 1998 up to the present time, when the English SAD system [2, 3] with the formal English-style
language ForTheL [4] was constructed for modern computers. (Some historical details can be found in [5].)

Below, we stop on some specific features of the English SAD system analyzing some modern approaches to automated theorem proving and the place of the English SAD system among them.

2 Formalization Styles in Automated Reasoning

To characterize the place English SAD in the automated theorem-proving “world”, it is convenient to consider the variety of existing approaches to automated theorem-proving with respect to the three characteristics: the style of formalization that they support, the style of proof that they require, and the granularity of proof that they accept.

Formalization style. This is characterized by the choice of preliminaries, the form of definitions (whether they are computable), the way of reasoning (whether it is constructive, or strictly typed, or calculation-based), and so on. Obviously, the style is influenced, above all, by the choice of base logic and of fundamental theories used for formalization.

At present, there exist two main trends in formalization: applying higher-order logic (type theory) and first-order logic (set theory). Types are used in the majority of the well-known assistant-systems such as Isabelle/HOL [6], Coq [7], Omega [8], PVS [9], HOL [10], Automath [11], Theorema [12], Lambda-Clam [13], and others. The type-theoretic approach favors inductively defined domains and recursive definitions and is well-suited to the formalization of concepts of programming or engineering. It may not, however, be the ideal framework for the formalization of traditional mathematics [14], although most of the systems just mentioned include a collection of purely mathematical theories.

On the opposite side, the system Mizar [15] uses classical first-order logic and Tarski-Grothendieck set theory. This approach corresponds well to the traditional style of mathematical presentation and the col-
lection of JFM articles verified in Mizar base constitutes the largest library of mathematical computer knowledge at present.

The English SAD system does not adopt any kind of set theory (or any other foundational theory) as the common base for all formalizations. It prefers to define a particular set of preliminaries for a problem under consideration, choosing base concepts on the appropriate level of abstraction in first-order classical logic.

Proof style. Another important property of a proof assistant is what kind of input it takes. Interactive systems are most often tactic-driven, meaning that a given statement is being proved by a sequence of instructions given to a system. These instructions, tactics, can be primitive, like applying an inference rule, or rather complex, like generating a proof plan for the current subgoal or running an external prover. Systems of that type are Isabelle, PVS, Omega, Coq, HOL and others. Working with such a system is easy if it provides a terse set of powerful tactics which are generally sufficient to capture the desired inferences.

The second type of system accepts propositions and proofs written in a formal logical language. Of course, this language must be extended with facilities that structure the logical formulas into a proof. The verification system must be capable of checking each successive proof step. Typical representatives of systems of that kind are Mizar and English SAD; Isabelle, with introduction of Isar [16] (containing a structured proof language imitating a language of mathematical proofs) can be considered as a proof-driven system, too.

The distinction between the two forms of input is not sharp. If one can prove theorems in a tactic-driven proof assistant using mostly the tactics of intermediate goal introduction and of automated subgoal closure, such a system can viewed as proof-driven. Conversely, if the steps in a proof that is submitted to a proof-driven system need to be supplied with detailed hints to the verifier, such a system can be viewed as tactic-driven, with the hints playing the role of the tactics.

Proof granularity. The reasoning power of mathematical assistance can be weaker or stronger, varying the requirements that are placed on the user. Systems can therefore range between proof checkers or
“proof finders”. The former accept only proof steps having the form of inference rule applications, and, hence, proofs supplied by them have to be fully detailed. Mizar is a system of that kind, though the set of inference rules of Mizar is quite large. Systems which we call “proof finders” use proof search methods or proof planning technique and try to close the “gaps” in proofs. The systems English SAD, Theorema, Nqthm [17], and ACL2 [18] are proof-driven systems of that kind.

Tactic-driven systems usually have enhanceable sets of tactics, so that the “reasoning power” is not peculiar to a system as such. Almost any tactic-driven system can be classified as a proof finder. However, some experiments with Isabelle and Coq show that following a suggested formalization style may be crucial here: whenever one tries to prove a complex theorem “at once”, without splitting it to a number of ad-hoc lemmas, without using special tactics and existing libraries, the “proof construction dialog” quickly becomes complex, wide-branched, and hardly traceable.

3 English SAD System: Specific Features

Having located the English SAD system in the ”three-dimensional space”, we provide a short description of its current state. English SAD is designed to perform three main tasks [2, 3]:

– an inference search in classical first-order logic [19];

– theorem proving in an environment of self-contained mathematical texts [20] written in the ForTheL language. (This language is a formal proof-writing language, but it is very close to the natural mathematical one);

– verifying self-contained mathematical ForTheL-texts containing propositions and their proofs [21].

When working English SAD, the user begins by writing an input text in ForTheL. This text contains the problem description, its premises, claims, and proofs for the claims, together with relevant preliminary information.

ForTheL is designed to be close to the natural language of real-
life mathematical publications issued by human mathematicians. In particular, the grammar of a ForTheL sentence follows the rules of common English grammar, though, of course, only a small fragment of English is formalized in the syntax of ForTheL.

The advantages of such an approach were a common point of many previous discussions (see, for example, the archives of the QED Project [22]). There are at least two reasons to pursue a verbose “natural” style. First, this provides the framework with a user-friendly interface. Second, English SAD developers share the viewpoint that a natural human text usually contains a useful information that lies beyond the scope of classical logic.

For example, in natural language, definitions distinguish from ordinary axioms, theorems from intermediate statements inside a proof, and so on. In a natural language sentence, there are nouns, which denote classes of entities; adjectives and verbs, which act as attributes and restrict classes; adjectives and verbs, which act as predicates and may relate different entities. ForTheL’s intention is to preserve these distinctions in formalization.

According to the Evidence Algorithm programme, the English SAD system can perform the following text’s transformations.

At the first level, the parser module of English SAD analyzes an input text, its structure, defined with ForTheL mark-up, and its logical content, encoded in ForTheL statements, and translates the text into its internal representation. The result of translation is a simply organized tree of statements, and being linearized, it gives a series of goal statements to deduce from predecessors.

At the second level, goal statements are processed one-by-one by the foreground reasoner of English SAD. This module is intended to reduce a given proof task to a number of inference-search subtasks for an automated prover. The reasoner works in a dialog with the prover: it may split the main goal to several simpler subgoals or propose an alternative subgoal in case if the prover fails to prove the current one. At present, the toolkit of the module is poor enough and contains some simplification methods on propositional level. Also note that this
module becomes redundant, when English SAD solves the problem of theorem proving.

Inference search tasks are resolved by a background prover at the third level. The English SAD prover is based on a special goal-driven sequent calculus for classical first-order logic with equality. The original notion of admissible substitution used in the calculus permits to preserve the initial signature of the task so that accumulated substitutional equations can be sent to a specialized solver, e.g. an external computer algebra system. Note that the English SAD system was implemented in such a way that English SAD can be connected with any of its external first-order provers, for example, with Otter [23], SPASS [24], or Vampire [25].

As the final step, the English SAD system outputs the result of its session.

There exist several interesting examples of proof verification made by English SAD in different branches of Mathematics: Ramseys finite and infinite theorems (Ramsey Theory), Cauchy-Bouniakowsky-Schwarz inequality (Mathematical Analysis), Bezouts identity in terms of abstract rings (Ring Theory), some properties of finite groups (Group Theory), Tarski’s fixed point theorem (Partially Ordered Set Theory), Furstenbergs proof of the infinitude of primes (Number Theory).

4 Possible Ways of Further Development of English SAD System

The given description of English SAD demonstrates that it satisfies the existing requirements to the complex processing of formalized mathematical texts. But its trial operation as well as a number of investigations and results obtained in automated reasoning last years have shown the desirability and possibilities of improving the capabilities of English SAD in the following directions.

Languages. The nearest objective is to provide the existing English ForTheL language with a LaTeX-environment in order to reach the reading of ForTheL-texts in the form closest to usual mathemat-
ical texts. Besides, there are drafts of the Russian and Ukrainian versions of the (English) ForTheL language. As a result, there exists the possibility to construct the next bidirectional translators: English ForTheL-texts ↔ Russian ForTheL-texts, English ForTheL-texts ↔ Ukrainian ForTheL-texts, and Russian ForTheL-texts ↔ Ukrainian ForTheL-texts, which will give the opportunity for using such a multilanguage extension of English SAD by a human, who knows only one of these languages, as well as for making automatic translation of a ForTheL-text written in one of these languages into a ForTheL-text written in another.

Reasoning. It is going to increase the heuristic possibilities of the system by using human-like proof methods depending on a subject domain under consideration. Besides, tools and methods for interfacing with some of the famous computer algebra systems are planned to develop and implement.

Deduction. On the basis of the research made on computer-oriented inference search in classical and non-classical sequent logics (see, for example, [26]), one can try to create a toolkit giving the possibility to construct one or another proof search method depending on a desire of an English SAD user or a subject domain under consideration. (This feature can play an important role in the case, when the application of non-classical reasoning becomes a necessary element for the successful decision of a task under consideration.)

5 Conclusion

The approach used in constructing the English English SAD system can be useful in any domain, where precise deductive style formalism is appreciated as the means of problem description. (Note that the problem formalization is always the hardest part of the whole work. Right formalization is a 80% guarantee of a successful verification.) In particular, these domains are: automated theorem proving, verification of mathematical papers, online training in mathematics and logic, construction of knowledge bases for formal theories, integration
of symbolic calculation with deduction. Also, it can be adapted for solving logical problems of decision-making theory and for verifying formal specifications of both software and hardware.

In the long run, the ideas used during creating the English SAD system and its toolkit can lead to constructing and supporting a powerful computer- and knowledge-based (Internet) infrastructure for mathematical research and education requiring deductive processing of formalized (not obligatory mathematical) texts.

References


Alexandre Lyaletsky, Alexander Lyaletski

National University of Life and Environmental Sciences of Ukraine
Email: foraal@mail.ru, a.lyaleski@nubip.edu.ua
Approximations of Finitely Supported Sets

Andrei Alexandru, Gabriel Ciobanu

Abstract

The theory of finitely supported algebraic structures represents a reformulation of Zermelo-Fraenkel set theory obtained by requiring that every set structure to be finitely supported according to a certain action of a group of permutations of some basic elements named atoms. Approximation of sets is a fundamental notion of rough set theory proposed by Pawlak. In this paper we study the approximation of finitely supported sets by using the approximation techniques from the theory of rough sets.

Keywords: finitely supported structures, Galois connections, fixed points, Pawlak approximation.

1 Introduction

The theory of finitely supported algebraic structures is strongly related to the theory of nominal sets [2] used in computer science. It is called ‘Finitely Supported Mathematics’ in pure set theoretical papers related to the foundations of mathematics [1]. This theory represents an alternative framework for working with infinite structures hierarchically constructed by involving some basic elements (called atoms) by dealing only with a finite number of entities that form their supports.

A nominal set is defined as a usual Zermelo-Fraenkel (ZF) set endowed with a group action of the group of (finitary) permutations over a certain fixed countable ZF set \( A \) (its elements are also called atoms by analogy with the Fraenkel and Mostowski models of set theory with atoms), formed by elements whose internal structure is not taken into consideration (i.e. by elements that can be checked only for equality)
and satisfying a finite support requirement. This requirement states that for any element in a nominal set there should exist a finite set of atoms such that any permutation fixing pointwise this set of atoms also leaves the element invariant under the related group action. Nominal sets represent a categorical mathematical theory of names studying scope, binding, freshness and renaming in formal languages based upon symmetry. Inductively defined finitely supported sets (that are finitely supported elements in the power set of a nominal set) involving the name-abstraction together with Cartesian product and disjoint union can encode syntax modulo renaming of bound variables. In this way, the standard theory of algebraic data types can be extended to include signatures involving binding operators. In particular, there is an associated notion of structural recursion for defining syntax-manipulating functions and a notion of proof by structural induction. Generalizations of nominal sets are involved in the study of automata or programming languages over infinite alphabets.

Finitely Supported Mathematics (FSM) is an alternative name for nominal algebraic structures, a name used in papers focusing on the foundations of set theory (rather than on applications in computer science). In order to describe FSM as a theory of finitely supported algebraic structures (that is finitely supported sets together with finitely supported internal algebraic operations), we use invariant sets which are essentially nominal sets extended over infinite (not necessarily countable) set of atoms The name ’invariant’ is motivated by the Tarski’s approach regarding logicality (i.e. a logical notion is defined by Tarski as one that is invariant under the permutations of the universe of discourse). FSM contains the family of ‘non-atomic’ (ordinary) ZF sets (which are proved to be trivial FSM sets) and the family of ‘atomic’ sets with finite supports (hierarchically constructed from the empty set and the fixed ZF set $A$). The main question now is whether a classical ZF result (obtained in the framework of non-atomic sets) can be adequately reformulated by replacing ‘non-atomic element/set’ with ‘atomic finitely supported element/set’ (according to the canonical actions of the group of permutations of $A$) in order to be valid also for
Approximations of Finitely Supported Sets

atomic sets with finite supports. The (non-atomic) ZF results cannot be directly translated into the framework of atomic finitely supported sets, unless we are able to reprove their new formulations internally in FSM, i.e. by involving only finitely supported structures even in the intermediate steps of the proof. This is because the family of finitely supported sets is not closed under subset constructions, and we cannot use something outside FSM in order to prove something in FSM.

The meta-theoretical techniques for the translation of a result from non-atomic structures to atomic structures are based on a refinement of the finite support principle form [2] called “S-finite supports principle” claiming that for any finite set $S$ of atoms, anything that is definable in higher order logic from $S$-supported structures using $S$-supported constructions is also $S$-supported. The formal involvement of the $S$-finite support principles implies a constructive method for defining the support of a structure by employing the supports of the sub-structures of a related structure.

Formally, in order to describe FSM, we fix an infinite ZF set $A$ (however, despite classical set theory with atoms, we do not need to modify the ZF axioms of extensionality and foundation). A finite set (without any other considerations) is referred to a set of the form $\{x_1, \ldots, x_n\}$. An infinite set (without any other considerations) is referred to a set which is not finite. The elements of $A$ (atoms) are entities whose internal structure is considered to be irrelevant, and which are considered as basic for a higher-order construction. An invariant set $(X, \cdot)$ is defined as a ZF set $X$ equipped with an action $\cdot$ on $X$ of the group of all permutations of $A$, having the additional property that any element $x \in X$ is finitely supported. In a pair $(X, \cdot)$ formed by a ZF set $X$ and a group action $\cdot$ on $X$ of the group of all permutations of $A$ (denoted by $S_A$), an arbitrary element $x \in X$ is finitely supported if and only if there exists a finite family $S \subseteq A$ such that any permutation of $A$ that fixes $S$ pointwise (i.e. any $\pi \in \text{Fix}(S)$, where $\text{Fix}(S) = \{\pi \in S_A \mid \pi(a) = a, \forall a \in S\}$) also leaves $x$ invariant under the group action $\cdot$. The least finite set (w.r.t. the inclusion relation) supporting a finitely supported element $x$ (which is defined
as the intersection of all finite sets supporting \( x \) is called the **support** of \( x \) and is denoted by \( \text{supp}(x) \). An empty supported element \( x \in X \) is called equivariant. If there exists an action \( \cdot \) of the group of permutations of \( A \) on a set \( X \), then there is a canonical action \( \ast \) of the group of permutations of \( A \) on \( \varphi(X) = \{ Y \mid Y \subseteq X \} \), defined by \((\pi, Y) \mapsto \pi \ast Y := \{ \pi \cdot y \mid y \in Y \}\) for all permutations \( \pi \) of \( A \) and all \( Y \subseteq X \). The set \( \varphi_{fs}(X) \) represents the family of those finitely supported subsets of \( X \) as elements in \( \varphi(X) \) with respect to the action \( \ast \); \( \varphi_{fs}(X) \) is an invariant set whenever \( X \) is an invariant set. A subset of an invariant set having the property that all its elements are supported by the same set of atoms is called uniformly supported. The Cartesian product of two invariant sets \( (X, \cdot) \) and \( (Y, \diamond) \) is an invariant set with the canonical action \((\pi, (x, y)) \mapsto (\pi \cdot x, \pi \diamond y)\). Generally, an FSM set is a finitely supported subset of an invariant set. The class of FSM sets is closed under power set of Cartesian constructions. The sets \( \varphi_{fin}(X) \) of all finite subsets of \( X \) and \( \varphi_{cofin}(X) \) of all subsets of \( X \) having finite complements are FSM sets whenever \( X \) is an FSM set. A relation (or, particularly, a function) between two invariant sets is finitely supported/equivariant if it is finitely supported/equivariant as a subset of the Cartesian product of those two invariant sets. Particularly, a function between two FSM sets \( (X, \cdot) \) and \( (Y, \diamond) \) is supported by a finite set \( S \) if and only if \( f(\pi \cdot x) = \pi \diamond f(x) \), \( \pi \cdot x \in X \) and \( \pi \diamond f(x) \in Y \) for all \( x \in X \) and all permutations \( \pi \) that fix \( S \) pointwise. The set of all functions from the FSM set \( (X, \cdot) \) to the FSM set \( (Y, \diamond) \) is an FSM set denoted by \( Y_{fs}^{X} \). Whenever an element \( a \in A \) appears in (the construction of) an FSM set \( (X, \cdot) \) (particularly if \( X = A \)), the effect of a permutation of atoms \( \pi \) on \( a \), under \( \cdot \), is \( \pi(a) \). Therefore, canonical actions in FSM are defined from the action \( a \mapsto \pi(a) \) of the group of permutations of atoms on \( A \), using the rules of constructing actions for power sets and Cartesian products.
2 Approximations of Finitely Supported Sets

Definition 2.1. An invariant partially ordered set is an invariant set equipped with an equivariant partially ordered relation.

Definition 2.2. Let \((P, \sqsubseteq_P, \cdot_P)\) and \((Q, \sqsubseteq_Q, \cdot_Q)\) be two invariant posets, and \(f : P \to Q, g : Q \to P\) two functions. The pair \((f, g)\) is a finitely supported Galois connection between \(P\) and \(Q\) if the following conditions are satisfied:

- \(f\) and \(g\) are finitely supported order preserving functions;
- for all \(p \in P\) and \(q \in Q\), \(f(p) \sqsubseteq_Q q\) if and only if \(p \sqsubseteq_P g(q)\).

The map \(g\) is called the finitely supported adjoint (of \(f\)) and \(f\) is called the finitely supported co-adjoint (of \(g\)).

The following FSM characterization of Galois connections can be directly proved as in the ZF framework.

Proposition 2.3. Let \((P, \sqsubseteq_P, \cdot_P)\) and \((Q, \sqsubseteq_Q, \cdot_Q)\) be two invariant posets, and \(f : P \to Q, g : Q \to P\) two functions. The pair \((f, g)\) is a finitely supported Galois connection between \(P\) and \(Q\) if and only if the following conditions are satisfied:

- \(f\) and \(g\) are finitely supported order preserving functions;
- for all \(p \in P\) and \(q \in Q\), \(f(g(q)) \sqsubseteq_Q q\) and \(p \sqsubseteq_P g(f(p))\).

Let \((U, \cdot)\) be an invariant set and \(\varepsilon\) a finitely supported equivalence relation on \(U\). We denote by \([x]_\varepsilon\) the equivalence class of an element \(x \in U\), i.e. \([x]_\varepsilon = \{y \mid y \varepsilon x\}\). We define the following functions (named FSM Pawlak’s approximations of \(\varepsilon\)):

- \(\varepsilon : \wp_{fs}(U) \to \wp_{fs}(U), \varepsilon(X) = \{x \in U \mid [x]_\varepsilon \subseteq X\}\);
- \(\overline{\varepsilon} : \wp_{fs}(U) \to \wp_{fs}(U), \overline{\varepsilon}(X) = \{x \in U \mid [x]_\varepsilon \cap X \neq \emptyset\}\).

Theorem 2.4. The pair \((\overline{\varepsilon}, \varepsilon)\) is well-defined, and form a finitely supported Galois connection on \(\wp_{fs}(U)\).
We can prove easily that \((x \cdot y)\cdot x = x\) is an invariant set. Let 
\(x \in \text{Fix}(\text{supp}(x))\) and \(x \in U\). We have \(x \cdot [x]_x = \{x \cdot y \mid y \in x\}\) and 
\([x \cdot x]_x = \{z \mid z \in (x \cdot x)\}\). Let \(t \in x \cdot [x]_x\). Then \(t = x \cdot y\), for some 
\(y \in x\). Since \(x\) is finitely supported and \(x\) fixes its support pointwise, we have 
\((x \cdot y)\cdot x = x\), and so \(t \in [x \cdot x]_x\). Conversely, if \(t \in [x \cdot x]_x\), then 
\(t = x \cdot y\), for some \(y \in x\). Since \(x\) is finitely supported and \(x\) fixes its support pointwise, we also have 
\((x^{-1} \cdot t)\cdot x\), that is, \(t = x \cdot (x^{-1} \cdot t) \in x \cdot [x]_x\). Therefore, 
x \cdot [x]_x = [x \cdot x]_x\) for all \(x \in \text{Fix}(\text{supp}(x))\) and all \(x \in U\).

In order to prove that \(x \in \mathcal{E}(x)\) and \(x \in \mathcal{E}(x)\) are supported by 
\(x \cdot \mathcal{E}(x) \cup \mathcal{E}(x)\). Let \(x \in \text{Fix}(\mathcal{E}(x) \cup \mathcal{E}(x))\). Thus, \(x \cdot x = X\). 
Let \(c \in X\), that is, \([c]_x \subseteq X\). Since \(x \in \text{Fix}(\mathcal{E}(x))\), we have 
\([x \cdot c]_x = x \cdot [c]_x \subseteq \mathcal{E}(x) \cdot x = x\). Therefore, \(x \cdot c \in X\), and \(c \cdot \mathcal{E}(x)\) is finitely supported. Let \(d \in \mathcal{E}(x)\), that is, \([d]_x \cap X = \emptyset\). There exists 
d \in [d]_x and \(d \in X\). Since \(x\) fixes both \(\mathcal{E}(x)\) and \(\mathcal{E}(x)\), we have 
\(x \cdot d \in x \cdot [d]_x = [x \cdot d]_x \) and \(x \cdot d \in X\). Therefore, \(x \cdot d \in [x \cdot d]_x \cap X\), and so \(x \cdot d \in \mathcal{E}(x) \cap X\) is finitely supported.

It remains to prove that \(x \in \mathcal{E}(x)\) and \(x \in \mathcal{E}(x)\) are finitely supported (by \(\mathcal{E}(x)\)). 
This follows directly from the \(S\)-finite support principle, but we also provide an alternative direct proof. Let \(x \in \text{Fix}(\mathcal{E}(x))\) and \(x \in \mathcal{E}(x)\). We have to prove that \(x \in x \cdot x = x \cdot x\) and \(x \cdot x = x \cdot x\) is 
\(x \cdot x\). Let \(y \in x \cdot x\), that is \([y]_x \subseteq x \cdot x\). We have \(y = x \cdot (x \cdot y)\), where 
\([x \cdot y]_x = x \cdot x \cdot [y]_x \subseteq x \cdot x \cdot (x \cdot x) = x\). Therefore, \(y \in x \cdot x\). 
Conversely, let \(y \in x \cdot x\), that is, \(y = x \cdot x\) where \([x]_x \subseteq x\). We have 
\([y]_x = [x \cdot x]_x = x \cdot x \cdot [x]_x \subseteq x \cdot x\), and so \(y \in x \cdot x\).

Now, let \(y \in x \cdot x\), that is, \([y]_x \cap (x \cdot x) = \emptyset\). Let \(z \in [y]_x \cap (x \cdot x)\). 
We have \(x \cdot z = x \cdot [y]_x = x \cdot [x \cdot y]_x = x \cdot [x \cdot z]_x = x\). Therefore, \([y]_x \cap X = \emptyset\). Since \(y = x \cdot (x \cdot y)\), we have \(y \in x \cdot x\). 
Conversely, let \(y \in x \cdot x\). We have \(y = x \cdot x\) where \([x]_x \cap X = \emptyset\). If 
z \in [x]_x \cap X, then \(x \cdot z \in [y]_x \cap (x \cdot x)\). Therefore, \([y]_x \cap (x \cdot x) = \emptyset\), 
and so \(y \in x \cdot x\).
Now, fix some \( X \in \wp_{fs}(U) \). Let \( x \) be an arbitrary element of \( X \). Let \( y \in [x]_\varepsilon \). Since \( \varepsilon \) is an equivalence relation, we also have \( x \in [y]_\varepsilon \). However, \( x \in X \), and so \( [y]_\varepsilon \cap X \neq \emptyset \). Therefore, \([x]_\varepsilon \subseteq \varepsilon(X)\). Since \( x \) is an arbitrary element from \( X \), we obtain \( X \subseteq \varepsilon(\varepsilon(X)) \). Now, let \( x \) be an arbitrary element from \( \varepsilon(\varepsilon(X)) \), that is, \([x]_\varepsilon \cap \varepsilon(X) \neq \emptyset \). Let \( y \in [x]_\varepsilon \cap \varepsilon(X) \). This means \( x \in [y]_\varepsilon \), and \([y]_\varepsilon \subseteq X \). Therefore, \( x \in X \), and so \( \varepsilon(\varepsilon(X)) \subseteq X \). According to Proposition 2.3, we have that \((\varepsilon, \varepsilon)\) is a finitely supported Galois connection on \( \wp_{fs}(U) \).

**Definition 2.5.**
- An invariant lattice is an invariant set \((L, \cdot)\) together with an equivariant (empty supported) lattice order \( \sqsubseteq \) on \( L \). If \( \sqsubseteq \) is finitely (possibly non-empty) supported, then \( L \) is called finitely supported lattice.

- An invariant complete lattice is an invariant partially ordered set \((L, \sqsubseteq, \cdot)\) such that every finitely supported subset \( X \subseteq L \) has a least upper bound with respect to the order relation \( \sqsubseteq \). If \( \sqsubseteq \) is finitely supported, then \( L \) is called finitely supported complete lattice.

- An invariant lattice \((L, \sqsubseteq, \cdot)\) is called an invariant Boolean lattice if it is distributive, it is bounded by a unique least element and a unique greatest element, and it is uniquely complemented. If \( \sqsubseteq \) is finitely supported, then \( L \) is called finitely supported Boolean lattice.

The following result is a generalization of Theorem 3.34 in [1] obtained by generalizing from ‘equivariant (empty supported)’ to ‘finitely (possibly non-empty) supported’.

**Lemma 2.6.** Let \((L, \sqsubseteq, \cdot)\) be an invariant complete lattice and \( f : L \to L \) a finitely supported order preserving function over \( L \). Let \( P \) be the set of fixed points of \( f \). Then \((P, \sqsubseteq, \cdot)\) is a non-empty finitely supported complete sublattice of \( L \).

**Proof.** We prove that \( P \) is finitely supported.
Since $f$ is finitely supported by $\text{supp}(f)$ it follows that for all $\pi \in \text{Fix}(\text{supp}(f))$ and all $x \in L$ we have $f(\pi \cdot x) = \pi \cdot f(x)$. Thus, for $\pi \in \text{Fix}(\text{supp}(f))$, whenever $x$ is a fixed point of $f$, we have $f(\pi \cdot x) = \pi \cdot f(x) = \pi \cdot x$, and so $\pi \cdot x$ is also a fixed point of $f$. Thus $\pi \ast P = P$ for all $\pi \in \text{Fix}(\text{supp}(f))$, which means that $(P, \cdot|_P)$ is a finitely supported subset (supported by $\text{supp}(f)$) of the invariant set $(L, \cdot)$, and also $\sqsubseteq|_P$ is a finitely supported subset (supported by $\text{supp}(f)$) of $\sqsubseteq$.

We prove now that $P$ is non-empty.

Let $D = \{d \in L \mid d \sqsubseteq f(d)\}$. First we remark that $D$ is non-empty. This is obvious because at least 0 (the least element of $L$) belongs to $D$. We prove that $D$ is finitely supported. We claim that $\text{supp}(f)$ supports $D$. Let $\pi \in \text{Fix}(\text{supp}(f))$ and $d \in D$ be arbitrarily chosen. Then $d \sqsubseteq f(d)$ (or, equivalently $(d, f(d)) \in \sqsubseteq$), and because $\sqsubseteq$ is equivariant we also have $\pi \otimes (d, f(d)) \in \sqsubseteq$, that is $\pi \cdot d \sqsubseteq \pi \cdot f(d)$, where $\otimes$ represents the canonical action on $D \times D$. Since $\pi \in \text{Fix}(\text{supp}(f))$ and $\text{supp}(f)$ supports $f$, we have $\pi \cdot d \sqsubseteq \pi \cdot f(d) = f(\pi \cdot d)$, and so $\pi \cdot d \in D$.

Since $d$ was chosen arbitrarily from $D$, we have $\pi \ast D \subseteq D$. We prove by contradiction that $\pi \ast D = D$. Let us suppose that $\pi \ast D \subsetneq D$. By induction, we get $\pi^n \ast D \subsetneq D$ for all $n \geq 1$. However, $\pi$ is a finitary permutation, and so there exists $k \in \mathbb{N}$ such that $\pi^k = \text{Id}$. We obtain $D \subseteq D$, a contradiction. It follows that $\pi \ast D = D$ whenever $\pi \in \text{Fix}(\text{supp}(f))$, and so $\text{supp}(f)$ supports $D$. Thus, $D$ is finitely supported, and there exists the least upper bound of $D$, $d_0 = \sqcup D$. For each $d \in D$ we have $d \subseteq d_0$. Since $f$ preserves the order relation, we have $f(d) \subseteq f(d_0)$. Since $d \in D$, it follows that $d \subseteq f(d) \subseteq f(d_0)$. Therefore, $d \subseteq f(d_0)$ for each $d \in D$. According to the definition of a least upper bound, we have that $d_0 \sqsubseteq f(d_0)$, which means that $d_0 \in D$. However, because $f$ is order preserving, we have $f(x) \in D$ for each $x \in D$. Since $d_0 \in D$, it follows that $f(d_0) \in D$. Thus, $f(d_0) \subseteq d_0$ because $d_0 = \sqcup D$. Therefore, we get $f(d_0) = d_0$, and so $P$ is non-empty.

We prove also that $P$ is a complete sublattice of $L$.

We have to prove that any subset of $P$ which is finitely supported as a subset of the invariant set $L$ has a least upper bound in $P$. Let $X$ be an arbitrary finitely supported subset of $L$ that is contained in the
finely supported set $P$. We have to prove that $X$ has a least upper bound in $P$. We already know that $X$ has a least upper bound (denoted by $\sqcup X$) in $L$ because $(L, \sqsubseteq, \cdot)$ is an invariant complete lattice.

Let $x \in X$ be an arbitrary element. We have that $x \subseteq \sqcup X$, and so $f(x) \subseteq f(\sqcup X)$. However, $X$ contains only fixed points of $f$, and so $f(x) = x$ and $x \subseteq f(\sqcup X)$. According to the definition of a least upper bound, it follows that $\sqcup X \subseteq f(\sqcup X)$. Now, let $y \supseteq \sqcup X$. Since $f$ is a order preserving function, we also have $f(y) \supseteq f(\sqcup X)$. We have already proved that $\sqcup X \subseteq f(\sqcup X)$, and hence $f(y) \supseteq \sqcup X$. We get that $f(y) \supseteq \sqcup X$ whenever $y \supseteq \sqcup X$.

Let $D' = \{d \in L \mid f(d) \subseteq d \text{ and } \sqcup X \subseteq d\}$. We prove that $\text{supp}(f) \cup \text{supp}(\sqcup X)$ supports $D'$, and so $D'$ is a finely supported set. We know that $\text{supp}(\sqcup X)$ exists because $L$ is an invariant set and $\sqcup X \in L$. Let us consider $\pi \in \text{Fix}(\text{supp}(f) \cup \text{supp}(\sqcup X))$, and $d \in D'$ be arbitrarily chosen. Then $f(d) \subseteq d$. Since $\sqsubseteq$ is equivariant, we also have $\pi \cdot f(d) \subseteq \pi \cdot d$. Moreover, because $f$ is supported by $\text{supp}(f)$ and $\pi$ fixes $\text{supp}(f)$ pointwise, we have $\pi \cdot f(d) = f(\pi \cdot d)$. Thus, $f(\pi \cdot d) \subseteq \pi \cdot d$. Since $d \in D'$, we also have $\sqcup X \subseteq d$. Therefore, $\pi \cdot \sqcup X \subseteq \pi \cdot d$. However, $\pi \cdot \sqcup X = \sqcup X$ because $\pi \in \text{Fix}(\text{supp}(\sqcup X))$. Finally, we obtain $\sqcup X \subseteq \pi \cdot d$, and so $\pi \cdot d \in D'$, which means $\pi \cdot D' \subseteq D'$. Since each permutation of $A$ is of finite order, as above we obtain $\pi \cdot D' = D'$. Thus, $\text{supp}(f) \cup \text{supp}(\sqcup X)$ supports $D'$, and so there exists the greatest lower bound of $D'$ denoted by $e = \sqcap D'$. Then for each $d \in D'$, we have $e \subseteq d$. Since $f$ preserves the order relation, we have also $f(e) \subseteq f(d)$. Since $d \in D'$, it follows that $f(e) \subseteq f(d) \subseteq d$. Therefore, $f(e) \subseteq d$ for each $d \in D'$. According to the definition of a greatest lower bound, we have that $f(e) \subseteq e$. Furthermore, $d \sqsupseteq \sqcup X$ for each $d \in D'$ implies $\sqcap D' \sqsubseteq \sqcup X$, which means $e \in D'$. However, because $f$ is order preserving and because $f(y) \supseteq \sqcup X$ whenever $y \supseteq \sqcup X$, we have that $f(x) \in D'$ for each $x \in D'$. Since $e \in D'$, it follows that $f(e) \in D'$, and so $e \sqsubseteq f(e)$ because $e = \sqcap D'$.

We proved that $e$ is a fixed point of $f$ such that $\sqcup X \subseteq e$. Therefore, $e \in P$ is an upper bound for $X$. What remains to be proved is that $e$ is the least upper bound for $X$ in the system $(P, \sqsubseteq)$. Let $e' \in P$ be
another upper bound for \( X \). Then \( \sqcup X \subseteq e' \) (since \( \sqcup X \) is the least upper bound for \( X \) in \( L \), and clearly \( e' \) is an upper bound for \( X \) in \( L \); it follows that \( e' \in D' \). Since \( e = \sqcap D' \), we get \( e \subseteq e' \). This means \( e = \sqcup X \) in \((P, \sqsubseteq)\).

\[ \text{Lemma 2.7. Let } (X, \cdot) \text{ be an invariant set.} \]

Then \((\wp_{fs}(X), \subseteq, \star)\) is an invariant complete Boolean lattice.

\[ \text{Proof. According to the definition of the canonical action } \star \text{ on } \wp_{fs}(X), \]

for any finitely supported subsets \( Y \) and \( Z \) of \( X \) with \( Y \subseteq Z \), we have

\[ \pi \star Y = \{ \pi \cdot y \mid y \in Y \} \subseteq \{ \pi \cdot z \mid z \in Z \} = \pi \star Z. \]

Thus, \( \subseteq \) is an equivariant order relation on \( \wp_{fs}(X) \).

Let \( F = (X_i)_{i \in I} \) be a finitely supported family of finitely supported subsets of \( X \) (we used the notation \((X_i)_{i \in I}\) only for an easy writing, without assuming that the mapping \( i \mapsto X_i \) is finitely supported). We know that \( \sqcup F = \bigcup_{i \in I} X_i \) exists in \( X \). We have to prove that \( \sqcup X_i \in \wp_{fs}(X) \). We claim that \( \text{supp}(F) \) supports \( \bigcup_{i \in I} X_i \). Let us consider \( \pi \in \text{Fix}(\text{supp}(F)) \). Let \( x \in \bigcup_{i \in I} X_i \). There exists \( j \in I \) such that \( x \in X_j \).

Since \( \pi \in \text{Fix}(\text{supp}(F)) \), we have \( \pi \star X_j \in F \), that is, there exists \( k \in I \) such that \( \pi \star X_j = X_k \). Therefore, \( \pi \cdot x \in \pi \star X_j = X_k \), and so \( \pi \cdot x \in \bigcup_{i \in I} X_i \). We obtain \( \pi \star \bigcup_{i \in I} X_i = \bigcup_{i \in I} X_i \), and so \( \bigcup_{i \in I} X_i \) is finitely supported. Therefore, \((\wp_{fs}(X), \subseteq, \star)\) is an invariant complete lattice.

The distributivity property over \((\wp_{fs}(X), \subseteq, \star)\) follows from the distributivity property on \( \wp(X) \) because the intersection and the union of every two finitely supported subsets of \( X \) is finitely supported by the union of the related supports. The greatest lower bound and the least upper bound in \( \wp_{fs}(X) \) are \( X \) and \( \emptyset \), respectively. Let \( U \in \wp_{fs}(X) \).

The complement of \( U \) is \( X \setminus U \). If \( S \) is a finite set of atoms supporting \( U \), we claim that \( S \) supports \( X \setminus U \). Indeed, let \( \pi \in \text{Fix}(S) \), which means \( \pi \cdot x \in U \), \( \forall x \in U \). We also have \( \pi^{-1} \in \text{Fix}(S) \). Let \( y \in X \setminus U \). If \( \pi \cdot y \in U \), then \( y \in \pi^{-1} \star U = U \). Therefore, \( \pi \cdot y \in X \setminus U \), which means \( S \) supports \( X \setminus U \). Therefore, \( X \setminus U \in \wp_{fs}(X) \), and \((\wp_{fs}(X), \subseteq, \star)\) is an invariant Boolean lattice. \[ \square \]
Approximations of Finitely Supported Sets

**Theorem 2.8.** Let \((U, \cdot)\) be an invariant set and \(\varepsilon\) a finitely supported equivalence relation on \(U\). Then the function \(\overline{\varepsilon}\) preserves the union of finitely supported families of finitely supported subsets of \(U\), while \(\varepsilon\) preserves the intersection of finitely supported families of finitely supported subsets of \(U\).

**Proof.** Let \(\mathcal{F}\) be a finitely supported family of finitely supported subsets of \(U\). According to Lemma 2.7, there exists \(\bigcup_{X \in \mathcal{F}} X\), and similarly, there also exists \(\bigcap_{X \in \mathcal{F}} X\). We claim that the family \(\mathcal{F}' = (\overline{\varepsilon}(X))_{X \in \mathcal{F}}\) is finitely supported by \(\text{supp}(\mathcal{F}) \cup \text{supp}(\varepsilon)\). Indeed, let \(Y \in \mathcal{F}'\), i.e. \(Y = \overline{\varepsilon}(X)\) for some \(X \in \mathcal{F}\), and \(\pi \in \text{Fix}(\text{supp}(\mathcal{F}) \cup \text{supp}(\varepsilon))\). Since \(\text{supp}(\varepsilon)\) supports \(\varepsilon\) we have \(\pi \star Y = \pi \star \overline{\varepsilon}(X) = \overline{\varepsilon}(\pi \star X) \in \mathcal{F}'\); the last membership relation follows because \(\pi\) fixes \(\text{supp}(\mathcal{F})\) pointwise, and so \(\pi \star X \in \mathcal{F}\). According to Lemma 2.7, there should exist \(\bigcup_{X \in \mathcal{F}} \varepsilon(X)\). Analogously, for any finitely supported family \(\mathcal{G}\) of finitely supported subsets of \(U\), there should exist \(\bigcap_{X \in \mathcal{G}} X\) and \(\bigcap_{X \in \mathcal{G}} \varepsilon(X)\).

We can directly verify that \(\overline{\varepsilon}(\bigcup_{X \in \mathcal{F}} X) = \bigcup_{X \in \mathcal{F}} \overline{\varepsilon}(X)\) and \(\varepsilon\left(\bigcap_{X \in \mathcal{G}} X\right) = \bigcap_{X \in \mathcal{G}} \varepsilon(X)\). As in the proof of Theorem 2.4, since \(\cap\) is an equivariant mapping, for any fixed \(x \in U\) we have that the family \(([x]_\varepsilon \cap X)_{X \in \mathcal{F}}\) is finitely supported by \(\text{supp}(x) \cup \text{supp}(\varepsilon) \cup \text{supp}(\mathcal{F})\), and so this family has a union. Indeed, for \(\pi \in \text{Fix}(\text{supp}(x) \cup \text{supp}(\varepsilon) \cup \text{supp}(\mathcal{F}))\) and \(X \in \mathcal{F}\) we have \(\pi \star ([x]_\varepsilon \cap X) = (\pi \star [x]_\varepsilon) \cap (\pi \star X) = [\pi \cdot x]_\varepsilon \cap (\pi \star X) = [x]_\varepsilon \cap (\pi \star X)\), where \(\pi \star X \in \mathcal{F}\). By direct calculation we have \(x \in \overline{\varepsilon}(\bigcup_{X \in \mathcal{F}} X) \leftrightarrow [x]_\varepsilon \cap (\bigcup_{X \in \mathcal{F}} X) \neq \emptyset \leftrightarrow \bigcup_{X \in \mathcal{F}} ([x]_\varepsilon \cap X) \neq \emptyset \leftrightarrow \exists Y \in \mathcal{F}, [x]_\varepsilon \cap Y \neq \emptyset \leftrightarrow \exists Y \in \mathcal{F}, x \in \overline{\varepsilon}(Y) \leftrightarrow x \in \bigcup_{X \in \mathcal{F}} \overline{\varepsilon}(X)\). Similarly, \(x \in \varepsilon\left(\bigcap_{X \in \mathcal{G}} X\right) \leftrightarrow [x]_\varepsilon \subseteq \bigcap_{X \in \mathcal{G}} X \leftrightarrow [x]_\varepsilon \subseteq X, \forall X \in \mathcal{G} \leftrightarrow x \in \bigcap_{X \in \mathcal{G}} \varepsilon(X)\).

**Theorem 2.9.** Let \((U, \cdot)\) be an invariant set and \(\varepsilon\) a finitely supported equivalence relation on \(U\). The set of all fixed points of \(\overline{\varepsilon}\) forms a finitely supported and complete Boolean sublattice of \(\wp(\wp(U)\subseteq)\).

**Proof.** We proved that \((\wp(U)\subseteq), \star)\) is an invariant complete Boolean lattice. We can easy remark that \(Y \subseteq \overline{\varepsilon}(Y), \forall Y \in \wp(U)\) (claim 1).
Also, we have that $U \setminus (\mathcal{E}(U \setminus Y)) = \mathcal{E}(Y)$, $\forall Y \in \varphi_{fs}(X)$ (claim 2). If $X$ is a fixed point of $\mathcal{E}$, i.e. $\mathcal{E}(X) = X$, and we denote by $X' = U \setminus X$, by (claim 2) we have $\mathcal{E}(X') = U \setminus (\mathcal{E}(U \setminus X')) = U \setminus (\mathcal{E}(X)) = U \setminus X = X'$. However from Proposition 2.3, since $(\mathcal{E}, \mathcal{E})$ is a finitely supported Galois connection on $\varphi_{fs}(U)$, we have $\mathcal{E}(\mathcal{E}(X')) \subseteq X'$, and so $\mathcal{E}(X') \subseteq X'$. From (claim 1) we also have $X' \subseteq \mathcal{E}(X')$, and so $X'$ is a fixed point of $\mathcal{E}$. This means that the fixed points of $\mathcal{E}$ form a Boolean sublattice of $\varphi_{fs}(U)$.

Furthermore, since $\mathcal{E}$ is finitely supported and order preserving (it is immediate that $X \subseteq Y \Rightarrow \mathcal{E}(X) \subseteq \mathcal{E}(Y)$), according to Lemma 2.6 and Lemma 2.7, we have that the set of all fixed points of $\mathcal{E}$ form a finitely supported complete sublattice of $\varphi_{fs}(U)$. Thus, the set of all fixed points of $\mathcal{E}$ forms a finitely supported, complete Boolean sublattice of $\varphi_{fs}(U)$. □

3 Conclusion

Using Pawlak approximation functions from the theory of rough sets, we were able to define and study upper and lower approximations of finitely supported subsets of (infinite) invariant sets. For this purpose, for each invariant set $(U, \cdot)$ and a finitely supported equivalence relation $\varepsilon$ on $U$, we defined the functions $\varepsilon : \varphi_{fs}(U) \rightarrow \varphi_{fs}(U)$ by $\varepsilon(X) = \{x \in U \mid [x]_\varepsilon \subseteq X\}$, and $\overline{\varepsilon} : \varphi_{fs}(U) \rightarrow \varphi_{fs}(U)$ by $\overline{\varepsilon}(X) = \{x \in U \mid [x]_\varepsilon \cap X \neq \emptyset\}$, where $[x]_\varepsilon$ is the equivalence class of $x$ modulo $\varepsilon$. These functions are well-defined in FSM, finitely supported by $\text{supp}(\varepsilon)$. They preserve the supremum and the infimum of finitely supported families in $\varphi_{fs}(U)$, and the pair $(\varepsilon, \overline{\varepsilon})$ is a finitely supported Galois connection on $\varphi_{fs}(U)$. Furthermore, the set of all fixed points of $\varepsilon$ forms a finitely supported complete Boolean sublattice of the invariant Boolean complete lattice $\varphi_{fs}(U)$. By using the $S$-finite support principle of FSM, we were able to extend related results presented in [1] from the framework of ‘equivariant (empty supported)’ sets to the general framework of ‘finitely (possibly non-empty) supported’ sets. We have also involved a different and simpler proving
method for the properties of approximations of finitely supported sets.

References


Compositions of Automata Defined on Finite Quasigroups

Volodymyr G. Skobelev, Volodymyr V. Skobelev

Abstract

Basic compositions, namely the parallel and sequential composition, and the feedback, for reversible automata defined on finite quasigroups are introduced and investigated. Some conditions under which the result of composition is a reversible automaton are established.

Keywords: automata, finite quasigroups, compositions.

1 Introduction

Within the last two decades, researches in the area of non-associative algebraic systems have been intensified significantly. This situation has been formed by the natural development of modern algebra, as well as by numerous successful applications of these algebraic systems for resolving of applied problems [1, 2].

Among the researches stated above, considerable attention is paid for studying quasigroups [3, 4]. It is well-known that finite quasigroups have been applied successively for resolving various problems of modern Cryptology [4-7]. Proceeding from these applications, researches of automata models defined on finite quasigroups seems to be an actual problem. Indeed, semi-automata are mathematical models for iterated hash-functions, while reversible automata are mathematical models for stream ciphers. Automata models defined on abstract finite quasigroups have been investigated in [8]. Besides, automata models defined on finite $T$-quasigroups [9] have been investigated in [10].
The main aim of the given paper is to investigate basic compositions of automata models defined on finite quasigroups. Analyzing these compositions, we will restrict ourselves by presentation of transition and output mappings for automata models in the form that is usually accepted in Abstract Automata Theory. Re-writing of these formulae in terms of quasigroups doesn’t cause difficulties, but significantly complicate the statement of the results.

2 Basic Notions and Definitions

2.1 Preliminaries

We recall that a quasigroup is an algebraic system \( Q = (Q, \circ) \), where \( \circ : Q \times Q \rightarrow Q \) is a binary operation such that for any \( a, b \in Q \) each of the equations \( a \circ x = b \) and \( x \circ a = b \) has a single solution.

Automata models on finite quasigroups can be defined as follows. Let \( \mathcal{Q}_Q \) be the set of all quasigroups defined on the set \( Q \) \((1 < |Q| < \infty)\).

With any quasigroup \( Q = (Q, \circ) \in \mathcal{Q}_Q \) the family of permutation semi-automata

\[ \mathcal{M}_Q = \{ M_\alpha = (Q, Q, \delta_\alpha) \}_{\alpha \in \{r,l\}} \]

can be associated, where the transition mapping \( \delta_\alpha \) \((\alpha \in \{r,l\})\) is defined as follows: \( \delta_r(q, x) = q \circ x \) and \( \delta_l(q, x) = x \circ q \). The transition mapping \( \delta_\alpha \) \((\alpha \in \{r,l\})\) can be extended on the set \( Q \times Q^+ \) in the usual way. It has been established in [8] that for any semi-automaton \( M_\alpha \in \mathcal{M}_Q \) \((Q = (Q, \circ) \in \mathcal{Q}_Q)\) the following two equalities hold:

\[ P^{(1)}_{M_\alpha, q, n}(q') = |Q|^{-1} \quad (\alpha \in \{r,l\}; q, q' \in Q; n \in \mathbb{N}), \]

\[ P^{(2)}_{M_\alpha, q, n} = |Q|^{-1} \left( 1 - \frac{|Q| - 1}{|Q|^n - 1} \right) \quad (\alpha \in \{r,l\}; q \in Q; n \in \mathbb{N}), \]

where \( P^{(1)}_{M_\alpha, q, n}(q') \) \((\alpha \in \{r,l\}; q, q' \in Q; n \in \mathbb{N})\) is the probability that randomly chosen input string \( p \in Q^n \) is a solution of the equation...
\[ \delta_\alpha(q, p) = q', \text{ and } P^{(2)}_{M_\alpha,q,n} (\alpha \in \{r, l\}; q \in Q; n \in \mathbb{N}) \text{ is the probability} \]

that two different randomly chosen input strings \( p, p' \in Q^n \) form some solution of the equation \( \delta_\alpha(q, p) = \delta_\alpha(q, p') \).

Therefore, any semi-automaton \( M_\alpha \in M_Q \) can be used as a mathematical model for some family of computationally secured iterative hash-functions.

With any ordered pair of quasigroups \((Q_1, Q_2) \in \Omega_Q \times \Omega_Q\) (where \( Q_i = (Q, \circ_i) \) \( i = 1, 2 \)) the family of reduced reversible Mealy automata

\[ A_{Q_1, Q_2} = \{A_{\alpha, \beta} = (Q, Q, Q, \delta_\alpha, \lambda_\beta)\}_{\alpha, \beta \in \{r, l\}} \]

can be associated, where the transition mapping \( \delta_\alpha (\alpha \in \{r, l\}) \) is defined as follows: \( \delta_r(q, x) = q \circ_1 x \), and \( \delta_l(q, x) = x \circ_2 q \), while the output mapping \( \lambda_\beta (\beta \in \{r, l\}) \) is defined as follows: \( \lambda_r(q, x) = q \circ_2 x \), and \( \lambda_l(q, x) = x \circ_2 q \).

Let \( S_Q \) be the symmetric group defined on the set \( Q \). With any quasigroup \( Q = (Q, \circ) \in \Omega_Q \) the family of reduced reversible Moore automata

\[ A_{Q, S_Q} = \{A_{\alpha, f} = (Q, Q, Q, \delta_\alpha, f \delta_\alpha)\}_{\alpha \in \{r, l\}, f \in S_Q} \]

can be associated, where the transition mapping \( \delta_\alpha (\alpha \in \{r, l\}) \) is defined as follows: \( \delta_r(q, x) = q \circ x \) and \( \delta_l(q, x) = x \circ q \).

It has been established in [8] that for any \( M \in A_{Q_1, Q_2} \cup A_{Q, S_Q} \) the ordered pair of initial automata \(( (M, q), (M^{-1}, q) ) (q \in Q) \) is some mathematical model of computationally secure stream cipher for which the initial state \( q \) is the secret short time key.

A quasigroup \( Q = (Q, \circ) \) is a \( T \)-quasigroup [9] if there exist some abelian group \((Q, +)\), an ordered pair of its automorphisms \((\varphi, \psi)\), and some fixed element \( c \in Q \), such that for all \( a, b \in Q \) holds the identity

\[ a \circ b = \varphi(a) + \psi(b) + c. \]

Therefore, this \( T \)-quasigroup \( Q = (Q, \circ) \) can be presented as the system \((Q, +, \varphi, \psi, c) (\varphi, \psi \in \text{Aut}(Q, +); c \in Q)\).
It has been established in [10] that any abelian group \((Q,+)\) generates the set
\[
\mathcal{Q}_{(Q,+)} = \{(Q,+,[\varphi,\psi,c]) \mid \varphi,\psi \in \text{Aut}(Q,+) & c \in Q\}
\]
of pair-wise different \(T\)-quasigroups.

Therefore, automata models defined on finite \(T\)-quasigroups form some non-trivial sufficiently wide subset of the set of automata models defined on finite quasigroups.

Automata models defined on finite \(T\)-quasigroups have been studied in [10]. The main result is based on the fundamental theorem for finite abelian groups.

Indeed, let an abelian group
\[
\mathcal{G} = (Q,+) \ (|Q| = p_1^{r_1} \cdots p_t^{r_t})
\]
be presented as the direct sum of cyclic subgroups of prime-power order, i.e.
\[
\mathcal{G} \cong \bigoplus_{i=1}^{t} \left( \bigoplus_{j=1}^{k_i} \mathbb{Z}_{p_j^{c_{ij}}} \right),
\]
where \(1 \leq c_{i1} \leq \cdots \leq c_{ik_i}\), and \(r_i = \sum_{j=1}^{k_i} c_{ij}\) for all \(i = 1, \ldots, t\).

Any automata model defined on a finite \(T\)-quasigroup \((Q,+,[\varphi,\psi,c])\) \((\varphi,\psi \in \text{Aut}(Q,+); c \in Q)\) can be presented as the parallel composition of simpler automata models. For this purpose, it is sufficient to present an automorphism \(\chi \in \text{Aut}(Q,+)(\chi \in \{\varphi,\psi\})\) as the vector which components are automorphisms of the abelian groups \(\mathbb{Z}_{p_j^{c_{ij}}}\) \((i = 1, \ldots, t; j = 1, \ldots, k_i)\).

As a result, time spent for one clock period for the automata models defined on finite \(T\)-quasigroups significantly decreases in comparison with time spent for one clock period for the automata models defined on abstract finite quasigroups.

This result can be generalized significantly for compositions of automata models defined on finite quasigroups.
Since any semi-automaton \( M = (Q, Q, \delta) \) can be treated as the automaton \( M = (Q, Q, Q, \delta, i_Q) \), where \( i_Q : Q \to Q \) is the identity mapping, it is sufficient to consider only compositions of automata defined on finite quasigroups.

### 2.2 Composition of automata

Let

\[
\mathcal{F}_{Q \times Q, Q} = \{ f : Q \times Q \to Q | \text{Val} \mu = Q \},
\]

and

\[
M_i = (Q, Q, Q, \delta^{(i)}, \lambda^{(i)}) \quad (i = 1, 2)
\]

be any automata defined on quasigroups arbitrarily selected from the set \( \Omega_Q \).

Four essentially different types of the parallel composition for the automata \( M_1 \) and \( M_2 \) can be defined as follows:

1) the automaton

\[
M_1 \parallel M_2 = (Q \times Q, Q \times Q, Q \times Q, \delta, \lambda),
\]

where

\[
\delta((q_1, q_2), (x_1, x_2)) = (\delta^{(1)}(q_1, x_1), \delta^{(2)}(q_2, x_2)),
\]

\[
\lambda((q_1, q_2), (x_1, x_2)) = (\lambda^{(1)}(q_1, x_1), \lambda^{(2)}(q_2; x_2)) ;
\]

2) the automaton

\[
M_1 \lor M_2 = (Q \times Q, Q, Q \times Q, \delta, \lambda),
\]

where

\[
\delta((q_1, q_2), x) = (\delta^{(1)}(q_1, x), \delta^{(2)}(q_2, x)),
\]

\[
\lambda((q_1, q_2), x) = (\lambda^{(1)}(q_1, x), \lambda^{(2)}(q_2; x)) ;
\]

3) the automaton

\[
M_1 \land \mu M_2 = (Q \times Q, Q \times Q, Q, \delta, \lambda) \quad (\mu \in \mathcal{F}_{Q \times Q, Q}),
\]
where the transition mapping $\delta$ is defined by formula (2), and

$$\lambda((q_1, q_2), (x_1, x_2)) = \mu(\lambda^{(1)}(q_1, x_1), \lambda^{(2)}(q_2; x_2));$$ \hspace{1cm} (8)

4) the automaton

$$M_1 \bigvee \bigwedge_{\mu} M_2 = (Q \times Q, Q, Q, \delta, \lambda) \quad (\mu \in \mathcal{F}_{Q \times Q, Q}),$$ \hspace{1cm} (9)

where the transition mapping $\delta$ is defined by formula (5), and

$$\lambda((q_1, q_2), x) = \mu(\lambda^{(1)}(q_1, x), \lambda^{(2)}(q_2, x)).$$ \hspace{1cm} (10)

The sequential composition $M_1 \rightarrow M_2$ of the automata $M_1$ and $M_2$ can be defined as the automaton

$$M = (Q \times Q, Q, Q, \delta, \lambda),$$ \hspace{1cm} (11)

where

$$\delta((q_1, q_2), x) = (\delta^{(1)}(q_1, x), \delta^{(2)}(q_2, \lambda^{(1)}(q_1, x))),$$ \hspace{1cm} (12)

$$\lambda((q_1, q_2), x) = \lambda^{(2)}(q_2, \lambda^{(1)}(q_1, x)).$$ \hspace{1cm} (13)

Four essentially different types of the feedback for the automaton $M_1 = (Q, Q, Q, \delta^{(1)}, \lambda^{(1)})$ can be defined as follows:

1) the automaton

$$\varphi^{(1)}_{\kappa} M_1 = (Q, Q, Q, \delta, \lambda) \quad (\kappa \in \mathcal{F}_{Q \times Q, Q}),$$ \hspace{1cm} (14)

where

$$\delta(q_{t+1}, x_{t+1}) = \delta^{(1)}(q_t, \kappa(x_{t+1}, y_t)),$$ \hspace{1cm} (15)

$$\lambda(q_{t+1}, x_{t+1}) = \lambda^{(1)}(q_t, \kappa(x_{t+1}, y_t)),$$ \hspace{1cm} (16)

for all $t \in \mathbb{Z}_+$, and where $y_0 \in Q$ is some fixed element;

2) the automaton

$$\varphi^{(2)}_{\mu} M_1 = (Q, Q, Q, \delta, \lambda) \quad (\mu \in \mathcal{F}_{Q \times Q, Q}),$$ \hspace{1cm} (17)
where \( \lambda = \lambda^{(1)} \), and

\[
\delta(q_{t+1}, x_{t+1}) = \mu(\delta^{(1)}(q_t, x_{t+1}), y_t),
\]

(18)

for all \( t \in \mathbb{Z}_+ \), and where \( y_0 \in Q \) is some fixed element;

3) the automaton

\[
\varrho_{\nu}^{(3)} M_1 = (Q, Q, Q, \delta, \lambda) \ (\nu \in \mathcal{F}_{Q \times Q, Q}),
\]

(19)

where \( \delta = \delta^{(1)} \), and

\[
\lambda(q_{t+1}, x_{t+1}) = \nu(\delta^{(1)}(q_t, x_{t+1}), y_t),
\]

(20)

for all \( t \in \mathbb{Z}_+ \), and where \( y_0 \in Q \) is some fixed element;

4) the automaton

\[
\varrho_{\mu, \nu}^{(4)} M_1 = (Q, Q, Q, \delta, \lambda) \ (\mu, \nu \in \mathcal{F}_{Q \times Q, Q}),
\]

(21)

where the transition mapping \( \delta \) is defined by formula (18), and the output mapping \( \lambda \) is defined by formula (20).

3 Main Results

Compositions of automata defined in Subsection 2.2, in addition to their immediate application in Structural Automata Theory, can be considered as some subject domain where composition of quasigroups and (possibly, partial) operations defined on it can be efficiently used. The basic properties of these compositions of automata can be characterized as follows.

3.1 Parallel and sequential composition of automata

Formulae (1), (4), (7), (9), and (11) imply that for any automata defined on quasigroups arbitrarily selected from the set \( \mathcal{Q}_Q \), the definition of automata \( M_1 \parallel M_2, M_1 \lor M_2, M_1 \land_\mu M_2, M_1 \lor \land_\mu M_2, \) and \( M_1 \rightarrow M_2 \) is based essentially on some quasigroup with the basic set \( Q \times Q \).
The following theorem can be proved on the base of the definition of the set of automata $A_{\tilde{Q}_1, \tilde{Q}_2} \cup A_{\tilde{Q}, Q}$, formulae (1)-(13), and the factor that $|Q| > 1$.

**Theorem 1.** For all quasigroups $\tilde{Q}_1, \tilde{Q}_2, Q \in \mathfrak{Q}_Q$, and for all automata $M_1, M_2 \in A_{\tilde{Q}_1, \tilde{Q}_2} \cup A_{\mathfrak{Q}, S_Q}$:

(i) each of the automata $M_1 \parallel M_2$, $M_1 \lor M_2$, and $M_1 \rightarrow M_2$ is a reversible automaton;

(ii) the automaton $M_1 \land \mu M_2$ is not a reversible automaton;

(iii) the automaton $M_1 \lor \land \mu M_2$ is a reversible automaton if and only if the following statement is true:

\[(\forall q_1, q_2 \in \tilde{Q})(\forall x_1, x_2 \in Q)(x_1 \neq x_2 \Rightarrow \\
\Rightarrow \mu(\lambda(1)(q_1, x_1), \lambda(2)(q_2, x_1)) \neq \mu(\lambda(1)(q_1, x_2), \lambda(2)(q_2, x_2))). \tag{22}\]

Due to items (ii) and (iii), the parallel composition of reversible automata, chosen from the set $A_{\tilde{Q}_1, \tilde{Q}_2} \cup A_{\mathfrak{Q}, S_Q}$, is not always a reversible automaton. At the same time, formula (22) implies that the automaton $M_1 \lor \land \mu M_2$ is obviously a reversible automaton, if $\mu$ is the operation in some quasigroup with the basic set $\tilde{Q}$.

It has been noted above, that the parallel and sequential composition of automata is based essentially on some quasigroup with the basic set $\tilde{Q} = Q \times Q$. Therefore, to resolve the problem of the design the parallel or sequential decomposition of any reversible automaton defined on a quasigroup $\tilde{Q} = (\tilde{Q}, \circ)$ onto the automata chosen from the set $A_{\tilde{Q}_1, \tilde{Q}_2} \cup A_{\mathfrak{Q}, S_Q}$ there is the need to carry out the following actions.

Firstly, it is necessary to construct some decomposition of the set $\tilde{Q}$, i.e. to choose some solution (if it exists) of the equation $\tilde{Q} = X \times X$.

Secondly, it is necessary to choose suitable quasigroups from the set $\mathfrak{Q}_Q$. The word combination "suitable quasigroups" has the following meaning.

If we try to design any of the decomposition $M_1 \parallel M_2$, $M_1 \lor M_2$, or $M_1 \rightarrow M_2$, then the operation in the quasigroup $\tilde{Q}$ can be presented naturally only in terms of the operations in the quasigroups chosen from the set $\mathfrak{Q}_Q$.
Compositions of Automata...

If we try to design the decomposition $M_1 \setminus \mu M_2$ or $M_1 \setminus \mu M_2$, then the operation in the quasigroup $\tilde{Q}$ can be presented naturally in terms of the operations in the quasigroups chosen from the set $\Omega_Q$ and some mapping chosen from the set $\mathcal{F}_{Q \times Q, Q}$.

From the position of Quasigroups Theory, in both these cases we come to need to solve some specific systems of equations on quasigroups, that can be applied in the considered subject domain, i.e. in Automata Theory

### 3.2 The feedback for an automaton

The following theorem establishes some sufficient conditions under which the feedback for an automaton $M_1 = (Q, Q, Q, \delta^{(1)}, \lambda^{(1)})$ is a reversible automaton. This theorem can be proved on the base of definition of the set of automata $A_{Q_1, Q_2} \cup A_{Q, S_Q}$, formulae (14)-(21), and the factor that $|Q| > 1$.

**Theorem 2.** For all quasigroups $Q_1, Q_2, Q \in \Omega_Q$, and for any automaton $M_1 \in A_{Q_1, Q_2} \cup A_{Q, S_Q}$:

(i) the automaton $\mathcal{M}_\kappa M_1$ is a reversible automaton, if the following statement is true:

$$(\forall x \in Q)(\forall y_1, y_2 \in Q)(y_1 \neq y_2 \Rightarrow \kappa(x, y_1) \neq \kappa(x, y_2))$$

$$& (\forall y \in Q)(\forall x_1, x_2 \in Q)(x_1 \neq x_2 \Rightarrow \kappa(x_1, y) \neq \kappa(x_2, y)); \quad (23)$$

(ii) the automaton $\mathcal{M}_\mu M_1$ is a reversible automaton, if the following statement is true:

$$(\forall q \in Q)(\forall y_1, y_2 \in Q)(y_1 \neq y_2 \Rightarrow \mu(q, y_1) \neq \mu(q, y_2))$$

$$& (\forall q_1, q_2 \in Q)(\forall y \in Q)(q_1 \neq q_2 \Rightarrow \mu(q_1, y) \neq \mu(q_2, y)); \quad (24)$$

(iii) the automaton $\mathcal{M}_\nu M_1$ is a reversible automaton, if the following statement is true:

$$(\forall q \in Q)(\forall y_1, y_2 \in Q)(y_1 \neq y_2 \Rightarrow \nu(q, y_1) \neq \nu(q, y_2))$$
\((\forall q_1, q_2 \in Q)(\forall y \in Q)(q_1 \neq q_2 \Rightarrow \nu(q_1, y) \neq \nu(q_2, y)); \quad (25)\)

\((iv)\) the automaton \(\mathcal{A}^{(4)}_{\mu,\nu}M_1\) is a reversible automaton, if both the statements \((24)\) and \((25)\) are true.

Due to Theorem 2, for each automaton \(M_1 = (Q, Q, Q, \delta^{(1)}, \lambda^{(1)})\) any of its feedback is a reversible automaton, if each of the mappings \(\kappa, \mu, \nu \in \mathcal{F}_{Q \times Q, Q}\) used in the definition of this feedback is the operation in some quasigroup with the basic set \(Q\).

4 Conclusion

In the given paper, basic compositions, namely the parallel and sequential composition, and the feedback, for reversible automata defined on finite quasigroups have been introduced. We have considered four different variants for the parallel composition, and four different variants for the feedback. The compositions of automata introduced in the given paper have been investigated from the position that the result satisfies to the property ”to be a reversible automaton”. The interrelations between this property and the solving of systems of equations of a specific form in finite quasigroups have been established. Detailed investigation of these interrelations defines some future trend of research. Detailed investigation of introduced compositions for the case when initial automata defined on finite \(T\)-quasigroups is another trend for future research.

References


Compositions of Automata...


Volodymyr G. Skobelev¹, Volodymyr V. Skobelev²

¹Leading Researcher/Institute of Cybernetics of NAS of Ukraine, Leading Researcher/Institute of Applied Mathematics and Mechanics of NAS of Ukraine
Email: skobelevvg@gmail.com

²Senior Researcher/Institute of Cybernetics of NAS of Ukraine, Senior Researcher/Institute of Applied Mathematics and Mechanics of NAS of Ukraine
Email: vvskobelev@incyb.kiev.ua
On Some Groupoids with Bol-Moufang Type Identities

Grigorii Horosh, Victor Shcherbacov, Alexandru Tcachenco, Tatiana Yatsko

Abstract

We count the number of groupoids of order 2, 3 and 4 with Bol-Moufang type identities which are listed in [6, 4].

Keywords: groupoid, Bol-Moufang type identity, (12)-parastrophe, (12)-parastrophic identity.

1 Introduction

A binary groupoid $(G, \cdot)$ is a non-empty set $G$ together with a binary operation “$\cdot$” which is defined on the set $G$. This definition is very general, therefore usually groupoids with some identities are studied. For example, groupoid with the identity of associativity is a semi-group.

For groupoids the following natural problem is researched: how many groupoids with some identities of small order there exist? A list of numbers of semigroups of orders up to 8 is given in [14], of order 9 – in [7]; a list of numbers of quasigroups up to 11 is given in [12, 16].

An identity based on a single binary operation is of Bol-Moufang type if “both sides consist of the same three different letters but one of them occurs twice on each side” [8]. We use list of generalized Bol-Moufang type identities given in [6].

We continue the study of groupoids with some Bol-Moufang type identities [11, 2, 15, 5, 3, 4]. Some results presented in this paper are...
given in [3, 4]. We use almost the same algorithm and program as in [3, 4].

Various properties of Bol-Moufang type identities in quasigroups and loops are studied in [8, 13, 6, 1].

We recall, groupoid \((Q, \ast)\) is called a quasigroup, if the following conditions are true [2]:

\[(\forall u, v \in Q)(\exists! x, y \in Q)(u \ast x = v \& y \ast u = v).\]

Groupoid \((G, \cdot)\) is isomorphic to groupoid \((G, \circ)\) if there exists a permutation \(\alpha\) of symmetric group \(S_G\) such that

\[x \circ y = \alpha^{-1}(\alpha x \cdot \alpha y)\]

for all \(x, y \in G\).

Groupoid \((G, \cdot)\) is anti-isomorphic to groupoid \((G, \circ)\) if there exists a permutation \(\alpha\) of symmetric group \(S_G\) such that

\[x \cdot y = \alpha^{-1}(\alpha y \cdot \alpha x)\]

for all \(x, y \in G\).

**Remark 1.** If groupoid \((G, \cdot)\) is anti-isomorphic to groupoid \((G, \circ)\), then groupoid \((G, \circ)\) is anti-isomorphic to groupoid \((G, \cdot)\). Really \(x \cdot y = \alpha(\alpha^{-1}y \circ \alpha^{-1}x)\) for all \(x, y \in G\).

**Remark 2.** In commutative groupoid \((G, \cdot)\) any anti-isomorphism coincides with isomorphism.

## 2 Results

### 2.1 Some results on groupoids of order two

It is clear that there exist 16 groupoids of order 2 and there exist \(n^2\) of groupoids of order \(n\).

We list isomorphic pairs of groupoids of order two. If a groupoid does not have a pair, then this groupoid has automorphism group of order two.

Below quadruple 22 12 means groupoid of order 2 with the following Cayley table:

\[
\begin{array}{c|cc}
\ast & 1 & 2 \\
\hline
1 & 2 & 2 \\
2 & 1 & 2 \\
\end{array}
\]
and so on. In such record groupoid is commutative if and only two elements, the second and the third, of a quadruple are equal.

It is easy to check that the following propositions are fulfilled.

**Proposition 1.** Only the following groupoids of order two are isomorphic in pairs:
- \(11 \ 11 \ and \ 22 \ 22;\)
- \(11 \ 12 \ and \ 12 \ 22;\)
- \(11 \ 21 \ and \ 21 \ 22;\)
- \(11 \ 22;\)
- \(12 \ 11 \ and \ 22 \ 12;\)
- \(12 \ 12;\)
- \(12 \ 21 \ and \ 21 \ 12;\)
- \(21 \ 11 \ and \ 22 \ 21;\)
- \(21 \ 21;\)
- \(22 \ 11.\)

**Proposition 2.** Only the following groupoids of order two are anti-isomorphic in pairs:
- \(11 \ 21 \ and \ 22 \ 12;\)
- \(21 \ 22 \ and \ 12 \ 11;\)
- \(11 \ 22 \ and \ 12 \ 12;\)
- \(21 \ 21 \ and \ 22 \ 11.\)

**Proposition 3.** Only the following groupoids of order two are isomorphic or anti-isomorphic:
- \(11 \ 11 \ and \ 22 \ 22;\)
- \(11 \ 12 \ and \ 12 \ 22;\)
- \(11 \ 21 \ and \ 21 \ 22 \ and \ 22 \ 12 \ and \ 12 \ 11;\)
- \(11 \ 22 \ and \ 12 \ 12;\)
- \(12 \ 21 \ and \ 21 \ 12;\)
- \(21 \ 11 \ and \ 22 \ 21;\)
- \(21 \ 21 \ and \ 22 \ 11.\)

**Corollary 1.** The following groupoids of order two are non isomorphic and non anti-isomorphic in pairs: \(11 \ 11; \ 11 \ 12; \ 11 \ 21; \ 11 \ 22; \ 12 \ 21; \ 21 \ 11; \ 21 \ 21.\)
Using the list of groupoids which is presented in Proposition 3 we can compose other lists of groupoids for Corollary 1. For example, instead of groupoid 1111 we can write groupoid 2222 and so on.

In the list presented in Corollary 1 semigroups of order two are underlined [16].

2.2 (12)-parastrophes of identities

We recall, (12)-parastroph of groupoid \((G, \cdot)\) is a groupoid \((G, *)\), where operation “\(*\)” is obtained by the following rule:

\[
x \ast y = y \cdot x.
\]

(1)

It is clear that for any groupoid \((G, \cdot)\) there exists its (12)-parastroph groupoid \((G, *)\).

Cayley table of groupoid \((G, *)\) is a mirror image of the Cayley table of groupoid \((G, \cdot)\) relative to main diagonal.

Suppose that an identity \(F\) is true in groupoid \((G, \cdot)\). Then we can obtain (12)-parastrophic identity \((F^*)\) of the identity \(F\) replacing the operation “\(\cdot\)” on the operation “\(*\)” and changing the order of variables using rule (1).

It is clear that an identity \(F\) is true in groupoid \((G, \cdot)\) if and only if in groupoid \((Q, *)\) identity \(F^*\) is true. Therefore we can formulate the following proposition.

Proposition 4. The number of groupoids of a finite fixed order in which the identity \(F\) is true coincides with the number of groupoids in which the identity \(F^*\) is true.

Example 1. Way 1. We find (12)-parastroph of the Bol-Moufang type identity \(F_1\): \(xy \cdot zx = (xy \cdot z)x\). Numeration of identities is taken from [8, 9].

We have \((x \ast z) \ast (y \ast x) = x \ast (z \ast (y \ast x))\). After renaming of variables \((y \leftrightarrow z)\) and operation \((\ast \rightarrow \cdot)\) we obtain the following Bol-Moufang type identity \(F_3\): \(xy \cdot zx = x(y \cdot zx)\).

Therefore \((F_1)^* = F_3\). It is true and vice versa \((F_3)^* = F_1\).
Way 2. We recall, left translation of a groupoid \((G, \cdot)\) is defined as follows: \(L_a x = a \cdot x\) for all \(x \in G\); right translation of a groupoid \((G, \cdot)\) is defined similarly: \(R_a x = x \cdot a\) for all \(x \in G\) and a fixed element \(a \in G\).

Then we can re-write identity \(F_1\) in the following form: \(L_x y \cdot R_x z = R_x (L_x y \cdot z)\).

There exists the following connections between left and right translations of a groupoid \((G, \cdot)\) and its \((12)\)-parastrophe [12, 15]:

\[
L^*_a = R_a, R^*_a = L_a.
\] (2)

Further, using rules (1) and (2) we have \(L_x z \cdot R_x y = L_x (z \cdot R_x y)\), \(x z \cdot y x = x (z \cdot y x)\). After renaming of variables \((y \leftrightarrow z)\) we obtain the following Bol-Moufang type identity \(F_3\): \(x y \cdot z x = x (y \cdot z x)\), i.e., \((F_1)^* = F_3\).

We count number of groupoids of orders two, three and four with Bol-Moufang type identities given in [6, 4].

2.3 Number of groupoids of order two, three and four with Bol-Moufang type identities

A new algorithm was developed and the corresponding program was written for generating the groupoids of small (2, 3, and 4) orders with Bol-Moufang identities.

Notice, number of groupoids of order 3 with mentioned in table identities are also given in [4].

Identities Left Bol and Right Bol, LC- and RC-, LN- and RN-, L2 and L3, M1 and M3, M2 and M4, T1 and T3, T4 and T5, are \((12)\)-parastrophic identities. Therefore the numbers of groupoids of fixed order with these \((12)\)-parastrophic identities coincide.
Table 1: Number of groupoids of order 2, 3 and 4 with Bol-Moufang identities.

<table>
<thead>
<tr>
<th>Name</th>
<th>Abbr.</th>
<th>Ident.</th>
<th>#2</th>
<th>#3</th>
<th>#4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Extra</td>
<td>EL</td>
<td>$x(y(zx)) = ((xy)z)x$</td>
<td>10</td>
<td>239</td>
<td>18744</td>
</tr>
<tr>
<td>Moufang</td>
<td>ML</td>
<td>$(xy)(zx) = (x(yz))x$</td>
<td>9</td>
<td>196</td>
<td>25113</td>
</tr>
<tr>
<td>Left Bol</td>
<td>LB</td>
<td>$x(y(xz)) = (x(yx))z$</td>
<td>9</td>
<td>215</td>
<td>22875</td>
</tr>
<tr>
<td>Right Bol</td>
<td>RB</td>
<td>$y((xz)x) = ((yx)z)x$</td>
<td>9</td>
<td>215</td>
<td>22875</td>
</tr>
<tr>
<td>C-loops</td>
<td>CL</td>
<td>$y(x(xz)) = ((yx)x)z$</td>
<td>10</td>
<td>209</td>
<td>26583</td>
</tr>
<tr>
<td>LC-loops</td>
<td>LC</td>
<td>$(xx)(yz) = (x(yx))z$</td>
<td>9</td>
<td>220</td>
<td>26583</td>
</tr>
<tr>
<td>RC-loops</td>
<td>RC</td>
<td>$y((zx)x) = (yz)(xx)$</td>
<td>9</td>
<td>220</td>
<td>26583</td>
</tr>
<tr>
<td>Middle Nuclear Square</td>
<td>MN</td>
<td>$y((xx)x) = (y(xx))z$</td>
<td>8</td>
<td>350</td>
<td>122328</td>
</tr>
<tr>
<td>Right Nuclear Square</td>
<td>RN</td>
<td>$y((xx)x) = (y(xx))z$</td>
<td>12</td>
<td>932</td>
<td>2753064</td>
</tr>
<tr>
<td>Left Nuclear Square</td>
<td>LN</td>
<td>$((xx)y)z = (xx)(yz)$</td>
<td>12</td>
<td>932</td>
<td>2753064</td>
</tr>
<tr>
<td>Comm. Moufang</td>
<td>CM</td>
<td>$(xy)(xz) = (xx)(zy)$</td>
<td>8</td>
<td>297</td>
<td>111640</td>
</tr>
<tr>
<td>Comm. C-loop</td>
<td>CC</td>
<td>$(y(xy))z = x(y(yz))$</td>
<td>8</td>
<td>169</td>
<td>12598</td>
</tr>
<tr>
<td>Comm. Alternative</td>
<td>CA</td>
<td>$((xx)y)z = z(x(yx))$</td>
<td>6</td>
<td>110</td>
<td>10416</td>
</tr>
<tr>
<td>Comm. Nuclear square</td>
<td>CN</td>
<td>$((xx)y)z = (xx)(zy)$</td>
<td>9</td>
<td>472</td>
<td>1321661</td>
</tr>
<tr>
<td>Comm. loops</td>
<td>CP</td>
<td>$((yx)x)z = z(x(yx))$</td>
<td>8</td>
<td>744</td>
<td>1078744</td>
</tr>
<tr>
<td>Cheban, 1</td>
<td>C1</td>
<td>$x((xy)y) = (yx)(xz)$</td>
<td>8</td>
<td>219</td>
<td>19846</td>
</tr>
<tr>
<td>Cheban, 2</td>
<td>C2</td>
<td>$x((xy)y) = (yx)(xz)$</td>
<td>6</td>
<td>153</td>
<td>12382</td>
</tr>
<tr>
<td>Lonely, I</td>
<td>L1</td>
<td>$(x(xy))z = y((zx)x)$</td>
<td>6</td>
<td>117</td>
<td>6076</td>
</tr>
<tr>
<td>Cheban, I, Dual</td>
<td>CD</td>
<td>$(yx)(xz) = (y(xz)x)$</td>
<td>8</td>
<td>219</td>
<td>19846</td>
</tr>
<tr>
<td>Lonely, II</td>
<td>L2</td>
<td>$(x(xy))z = y((zx)x)$</td>
<td>7</td>
<td>157</td>
<td>11489</td>
</tr>
<tr>
<td>Lonely, III</td>
<td>L3</td>
<td>$(x(xy))z = y((zx)x)$</td>
<td>7</td>
<td>157</td>
<td>11489</td>
</tr>
<tr>
<td>Mate, I</td>
<td>M1</td>
<td>$(x(xy))z = ((yz)x)x$</td>
<td>6</td>
<td>111</td>
<td>11188</td>
</tr>
</tbody>
</table>

Continued on next page
Table 1 – Continued from previous page

<table>
<thead>
<tr>
<th>Name</th>
<th>Abbr.</th>
<th>Ident.</th>
<th>#2</th>
<th>#3</th>
<th>#4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mate, II</td>
<td>M2</td>
<td>((y(xx))z = ((yz)x)x)</td>
<td>7</td>
<td>196</td>
<td>26785</td>
</tr>
<tr>
<td>Mate, III</td>
<td>M3</td>
<td>(x(x(yz)) = y((zx)x))</td>
<td>6</td>
<td>111</td>
<td>11188</td>
</tr>
<tr>
<td>Mate, IV</td>
<td>M4</td>
<td>(x(x(yz)) = y((xx)z))</td>
<td>7</td>
<td>196</td>
<td>26785</td>
</tr>
<tr>
<td>Triad, I</td>
<td>T1</td>
<td>((xx)(yz) = y(z(xx)))</td>
<td>6</td>
<td>162</td>
<td>67152</td>
</tr>
<tr>
<td>Triad, II</td>
<td>T2</td>
<td>(((xx)y)z = y(z(xx)))</td>
<td>6</td>
<td>180</td>
<td>53832</td>
</tr>
<tr>
<td>Triad, III</td>
<td>T3</td>
<td>((xx)y)z = (yz)(xx))</td>
<td>6</td>
<td>162</td>
<td>67152</td>
</tr>
<tr>
<td>Triad, IV</td>
<td>T4</td>
<td>((xx)y)z = ((yz)x)x)</td>
<td>6</td>
<td>132</td>
<td>42456</td>
</tr>
<tr>
<td>Triad, V</td>
<td>T5</td>
<td>(x(x(yz)) = y(z(xx)))</td>
<td>6</td>
<td>132</td>
<td>42456</td>
</tr>
<tr>
<td>Triad, VI</td>
<td>T6</td>
<td>((xx)(yz) = (yz)(xx))</td>
<td>8</td>
<td>1419</td>
<td>9356968</td>
</tr>
<tr>
<td>Triad, VII</td>
<td>T7</td>
<td>((xx)y)z = ((yx)x)z)</td>
<td>12</td>
<td>428</td>
<td>2914658</td>
</tr>
<tr>
<td>Triad, VIII</td>
<td>T8</td>
<td>((xx)(yz) = y((zx)x))</td>
<td>6</td>
<td>120</td>
<td>11580</td>
</tr>
<tr>
<td>Triad, IX</td>
<td>T9</td>
<td>((xx)y)z = y(z(xx)))</td>
<td>6</td>
<td>102</td>
<td>6192</td>
</tr>
<tr>
<td>Frute</td>
<td>FR</td>
<td>((xx)y)z = (yx)zx)</td>
<td>6</td>
<td>129</td>
<td>16600</td>
</tr>
<tr>
<td>Crazy Loop</td>
<td>CR</td>
<td>((xx)y)z = (yx)(xz))</td>
<td>7</td>
<td>136</td>
<td>12545</td>
</tr>
<tr>
<td>Krypton</td>
<td>KR</td>
<td>((xx)(yz) = (x(yz))x)</td>
<td>9</td>
<td>268</td>
<td>93227</td>
</tr>
</tbody>
</table>

3 Conclusion

In this paper the number of groupoids with very small order (2, 3 and 4) was shown. The complexity of calculation is \(n^{n\times n}\), this means that a number of possible groupoids is:

<table>
<thead>
<tr>
<th>#</th>
<th>Count</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>16</td>
</tr>
<tr>
<td>3</td>
<td>19,683</td>
</tr>
<tr>
<td>4</td>
<td>4,294,967,296</td>
</tr>
<tr>
<td>5</td>
<td>298,023,223,876,953,150</td>
</tr>
</tbody>
</table>

In the case of order 2, we spend 3 hours for calculation for every identity. One of the next steps in the development of our researches is the optimization of the proposed algorithm.
Acknowledgment. The authors thank very much both Referees for valuable remarks.

References


On Some Groupoids with Bol-Moufang Type Identities


Grigorii Horosh¹, Victor Shcherbacov², Alexandr Tcachenco³, Tatiana Yatsko⁴

¹Ph.D. Student/Institute of Mathematics and Computer Science of Moldova
Email: grigorii.horos@math.md

²Principal Researcher/Institute of Mathematics and Computer Science of Moldova
Email: victor.scherbacov@math.md

³Master Student /Institute of Mathematics and Computer Science of Moldova
Email: atkacheno405@gmail.com

⁴Master Student/Shevchenko Transnistria State University
Email: yaczkot@bk.ru
Maximal Slupecki Iterative Algebras of the 4-valued Diagonalizable Algebra and Related Algorithms

Andrei Rusu, Elena Rusu

Abstract

The 4-valued diagonalizable algebra $D_4$ (which is known to be the simplest non-classical model of the propositional provability logic) is considered together with its corresponding iterative Post algebra $D_4$. The maximal Slupecki iterative algebras of $D_4$ are presented including the algorithm related to the ultra-weak completeness problem in it.

Keywords: iterative Post algebra, function algebra, clone, completeness problem.

1 Introduction

Study of compositions of functions is important for the three branches of mathematics: the theory of many-valued logics, automata theory, and universal algebra. In the articles [1, 2], E. Post, one of the founders of the mathematical theory of many-valued logics, considered two problems. The first problem reads: which functions are obtainable by finite compositions from a given collection of initial functions? The second reads: describe all composition-closed classes of functions on a fixed set $A$. These problems are still actual when $A$ comprises more than two elements and is endowed with algebraic operations.
2 Preliminary Notions and Results

Since composition is not an algebraic operation, A. I. Malcev [3, 4] suggested to consider, instead of classes of functions with composition, special algebras whose underlying sets are composition-closed sets of functions. A. I. Malcev called these algebras iterative and pre-iterative algebras and described them as follows:

Given an arbitrary set $A$, denote by $P_A^{(n)}$ the set of all $n$-ary functions that are defined on $A$ and have values in $A$. Put $P_A = \cup_n P_A^{(n)}$. The iterative Post algebra is the algebra $P_A = (P_A; \zeta, \tau, \Delta, \nabla, \ast)$ whose operations for an $n$-ary function $f$ with $n > 1$ and an $m$-ary function $g$ are defined as follows:

\[
(\zeta f)(x_1, \ldots, x_n) = f(x_2, x_3, \ldots, x_n, x_1),
\]

\[
(\tau f)(x_1, \ldots, x_n) = f(x_2, x_1, x_3, \ldots, x_n),
\]

\[
(\Delta f)(x_1, \ldots, x_{n-1}) = f(x_1, x_1, x_2, \ldots, x_{n-1}),
\]

\[
(\nabla f)(x_1, \ldots, x_{n+1}) = f(x_2, x_3, \ldots, x_{n+1}),
\]

\[
(f \ast g)(x_1, \ldots, x_{n+m-1}) = f(g(x_1, \ldots, x_m), x_m, \ldots, x_{m+n-1}).
\]

The pre-iterative Post algebra is the algebra $P_A^* = (P_A; \zeta, \tau, \Delta, \ast)$. If the set $A$ comprises $k$ elements then we say that these algebras have finite rank $k$.

The subalgebras of the algebras $P_A$ and $P_A^*$ are referred to as the iterative and pre-iterative algebras respectively. A clone is a pre-iterative algebra containing all selectors $e^n_i(x_1, \ldots, x_n) = x_i$. At present, the clone theory is an important and popular direction in universal algebra. However, in applications, for example, in automata theory it is most natural to use iterative algebras, since such an algebra, containing a function $f$, also contains all functions that differ from $f$ in fictitious variables.

The classical problems of the theory of iterative algebras are the completeness problem and the problem of describing all subalgebras of these algebras. E. Post [1, 2] found all subalgebras of the algebra...
\( \mathcal{P}_2 \) and thereby solved both problems for the iterative Post algebra of rank 2. Description for all maximal subalgebras of the algebra \( \mathcal{P}_k \) was found for \( k = 3 \) by S. V. Yablonskiĭ [5, 6] and for the other values of \( k \) by I. Rosenberg [7, 8].

In view of great complexity of the universal criterion for completeness of a system of functions in the algebra \( \mathcal{P}_k \), a series of authors studied the possibility of generating this algebra by a set of functions containing a Slupecki function and some unary functions. (A Slupecki function is a function in \( \mathcal{P}_k \) having more than one essential variable and taking \( k \) different values. Each system of functions complete in \( \mathcal{P}_k \) must contain such a function.) A set of functions which together with an arbitrary Slupecki function generates the algebra \( \mathcal{P}_k \) is called fundamental. J. Slupecki [9] demonstrated that if \( k > 2 \) then the set \( P_k^{(1)} \) of all unary functions belonging to \( \mathcal{P}_k \) is fundamental.

A diagonalizable algebra [10] \( \mathfrak{D} \) is a boolean algebra \( \mathfrak{A} = (A; \&; \lor; \supset; \neg; 0, 1) \) with an additional operation \( \Box \) satisfying the relations:

\[
\begin{align*}
\Box(x \supset y) &= (\Box x \supset \Box y), \\
\Box x &= \Box \Box x, \\
\Box(\Box x \supset x) &= \Box s, \\
\Box 1 &= 1.
\end{align*}
\]

where 1 is the unit of \( \mathfrak{A} \). The diagonalizable algebras serve as algebraic models for propositional provability logic \( GL \) [11].

We consider the 4-valued diagonalizable algebra \( \mathfrak{D}^4 = (\{0, \rho, \sigma, 1\}; \&; \lor; \supset; \neg; 0, 1, \Box) \), where boolean operations are defined as usual, and

\[
\Box 0 = \Box \rho = \sigma, \quad \Box \sigma = \Box 1 = 1.
\]

In the following we consider the iterative Post algebra \( \mathfrak{D}^4 = (\mathfrak{D}^4; \zeta, \tau, \Delta, \nabla, *) \) of the term operations of the algebra \( \mathfrak{D}^4 \). We will investigate maximal Slupecki iterative algebras of the algebra \( \mathfrak{D}^4 \) (i.e. maximal iterative algebras containing all unary term functions of the algebra \( \mathfrak{D}^4 \)).

As usual a relation \( R(x_1, \ldots, x_m) \) on the algebra \( \mathfrak{D}^4 \) is any subset of \( \{0, \rho, \sigma, 1\}^m \). They say the term function \( f(x_1, \ldots, x_n) \) conserves
the relation $R(x_1, \ldots, x_m)$ on the algebra $\mathfrak{D}4$ if for any elements $\alpha_{ij} \in \{0, \rho, \sigma, 1\}$ $(i = 1, \ldots, m, \ j = 1, \ldots, n)$ for which the following relations hold

$$R(\alpha_{11}, \ldots, \alpha_{m1}), \ldots, R(\alpha_{1n}, \ldots, \alpha_{mn})$$

we have also the relation

$$R(f(\alpha_{11}, \ldots, \alpha_{1n}), \ldots, f(\alpha_{m1}, \ldots, \alpha_{mn})).$$

We use symbols $==$ to read it as ”is defined as”. Also denote the term function $((x \supset y) \& (y \supset x))$ by $(x \sim y)$. We consider next relations on the diagonalizable algebra $\mathfrak{D}4$ [12]:

$$R_{38}(x, y, z, u) == ((\Box x = \Box y) \& (\Box z = \Box u) \& (\Box x = \Box z) \&$$

$$(x \sim y) = (z \sim u)),

R_{39}(x, y, z, u) == ((\Box x = \Box y) \& (\Box z = \Box u) \&$$

$$(x = y) \lor (z = u) \lor (\Box x = \Box z)),

R_{40}(x, y, z, u) == (\Box(x \sim y) = \Box(z \sim u)).$$

Denote classes of term functions that preserve the relations $R_{38}$, $R_{39}$, and $R_{40}$ defined above on the diagonalizable algebra $\mathfrak{D}4$ by $K_{38}$, $K_{39}$, and $K_{40}$.

### 3 Main Result

**Theorem.** Classes of term functions $K_{38}$, $K_{39}$, and $K_{40}$ of the diagonalizable algebra $\mathfrak{D}4$ which preserve the corresponding relations $R_{38}$, $R_{39}$, and $R_{40}$, are the only maximal Slupecki iterative algebras of the algebra $\mathfrak{D}4$.

**Proof.** As it was mentioned earlier, each Slupecki class of term functions contain all unary terms. According to [12] any term of the diagonalizable algebra $\mathfrak{D}4$ conserves the relation $\Box x = \Box y$ on $\mathfrak{D}4$. This
Table 1. Unary operations of $\mathcal{D}4$

<table>
<thead>
<tr>
<th>$p$</th>
<th>$I_{1j}$</th>
<th>$I_{2j}$</th>
<th>$I_{3j}$</th>
<th>$I_{4j}$</th>
<th>$I_{5j}$</th>
<th>$I_{6j}$</th>
<th>$I_{7j}$</th>
<th>$I_{8j}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>$\rho$</td>
<td>$\rho$</td>
<td>$\sigma$</td>
<td>$\sigma$</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>$\rho$</td>
<td>0</td>
<td>$\rho$</td>
<td>0</td>
<td>$\rho$</td>
<td>$\sigma$</td>
<td>1</td>
<td>$\sigma$</td>
<td>1</td>
</tr>
<tr>
<td>$p$</td>
<td>$I_{i1}$</td>
<td>$I_{i2}$</td>
<td>$I_{i3}$</td>
<td>$I_{i4}$</td>
<td>$I_{i5}$</td>
<td>$I_{i6}$</td>
<td>$I_{i7}$</td>
<td>$I_{i8}$</td>
</tr>
<tr>
<td>$\sigma$</td>
<td>0</td>
<td>0</td>
<td>$\rho$</td>
<td>$\rho$</td>
<td>$\sigma$</td>
<td>$\sigma$</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>$\rho$</td>
<td>0</td>
<td>$\rho$</td>
<td>$\sigma$</td>
<td>1</td>
<td>$\sigma$</td>
<td>1</td>
</tr>
</tbody>
</table>

fact implies there are only 64 distinct unary term functions on $\mathcal{D}4$ presented in table 1. (Unary term functions $I_{ij}$ of the algebra $\mathcal{D}4$ from table 1, where $I_{ij}(p)$ ($i = 1, \ldots, 8; j = 1, \ldots, 8$) denotes the unary term function which for $p = 0$ and $p = \rho$ takes values from the $i$-th column, and for $p = \sigma$ and $p = 1$ it takes values from the $j$-th column. For example, $I_{11} = 0$, $I_{16} = p$, $I_{73} = \neg p$, $I_{58} = \Delta p$, $I_{88} = 1$.)

Now let us convince ourselves that the classes of term functions $K_{38}$, $K_{39}$, and $K_{40}$ are incomplete (i.e. they do not contain all term functions of $\mathcal{D}4$). In can be verified that each of them do not contain term function $((p \& q) \lor (p \& r) \lor (q \& r))$. Next we have to show that these classes $K_{38}$, $K_{39}$, and $K_{40}$ are two by two distinct. So, consider first functions presented in table 2. It can be checked that all functions $g_1, g_2, g_3, g_4$ are term functions since they conserve relation $\square x = \square y$ on $\mathcal{D}4$. Subsequently, classes $K_{38}$, $K_{39}$, and $K_{40}$ are two by two distinct by next relations:

\[
\begin{align*}
\{g_1, g_4\} & \subseteq K_{38}, & \{g_2, g_3\} & \cap K_{38} = \emptyset, \\
\{g_3, g_4\} & \subseteq K_{39}, & \{g_1, g_2\} & \cap K_{39} = \emptyset, \\
\{g_1, g_2\} & \subseteq K_{40}, & \{g_4, g_4\} & \cap K_{40} = \emptyset.
\end{align*}
\]

The necessary part of the theorem follows from the fact that the classes of term functions $K_{38}$, $K_{39}$, and $K_{40}$ are iterative Post algebras of the diagonalizable algebra $\mathcal{D}4$ and since they contains all unary term functions the conclusion is they are Slupecki iterative Post algebras. Since they are distinct it follows they are at most maximal.
The proof these classes are the only maximal Slupecki iterative Post algebras of $\mathfrak{D}4$ follows from the next lemmas and following considerations. Suppose using the system of term functions $\Sigma$ together with unary term functions and Malcev operations $\{\zeta, \tau, \Delta, \nabla, \ast\}$ we managed to obtain all term functions of $\mathfrak{D}4$. Suppose $\Sigma$ contains a system of term functions $\{f_1, f_2, f_3\}$ which do not belong to the corresponding classes $K_{38}$, $K_{39}$, and $K_{40}$, and do not contain other variables excepting $p_1, \ldots, p_n$.

Now consider the sufficient part of the theorem. Suppose the system of formulas $\Sigma$ is ultra-weakly complete in the logic $L\mathfrak{B}_2$. Suppose $\Sigma$ contains a system of formulas $\{f_1, f_2, f_3\}$ which do not belong to the corresponding classes $K_{38}$, $K_{39}$, and $K_{40}$ and do not contain other variables excepting $p_1, \ldots, p_n$. It is not supposed the term functions $f_1, f_2, f_3$ are distinct. It is sufficiently now to prove that conjunction $(p \& q)$ can be obtained in $\mathfrak{D}4$ from unary term functions, term functions $\{f_1, f_2, f_3\}$ using Malcev operations. The continuation of the proof is presented in the next lemmas 1, 2, 3 and 4. □

**Lemma 1.** Term functions $C(p, q)$ and $D(p, q)$ satisfying conditions

$$C[0, 0] = C[0, \rho] = C[\rho, 0] = 0, \ C[\rho, \rho] = \rho$$
$$D[\sigma, \sigma] = C[\sigma, 1] = D[1, \sigma] = \sigma, \ D[1, 1] = 1.$$  (1)

can be obtained via Malcev operations and unary term functions in $\mathfrak{D}4$ via term function $f_1$. 

\begin{table}[h]
\centering
\begin{tabular}{|c|c|c|c|c|}
\hline
$p$ & 0 & 0 & 0 & 0 \\
\hline
$q$ & 0 & $\rho$ & $\sigma$ & 1 \\
\hline
$g_1$ & 1 & $\sigma$ & $\rho$ & 0 \\
$g_2$ & $\rho$ & $\rho$ & $\rho$ & $\rho$ \\
$g_3$ & 0 & $\rho$ & 1 & 1 \\
$g_4$ & 0 & $\rho$ & $\sigma$ & 1 \\
\hline
\end{tabular}
\caption{The tables of the functions $g_1, g_2, g_3, g_4$}
\end{table}
Proof. Consider function $f_1$, which do not conserve the relation $R_{38}$ on $\mathcal{D}4$. Then there exist four ordered sets of elements $<\alpha_1,\ldots,\alpha_n>$, $<\beta_1,\ldots,\beta_n>$, $<\gamma_1,\ldots,\gamma_n>$ and $<\delta_1,\ldots,\delta_n>$ from $\mathcal{D}4$ such that

$$R_{38}(\alpha_i,\beta_i,\gamma_i,\delta_i), \ i = 1,\ldots,n$$

(2)

$$\begin{pmatrix}
F_1[\alpha_1,\ldots,\alpha_n] \\
F_1[\beta_1,\ldots,\beta_n] \\
F_1[\gamma_1,\ldots,\gamma_n] \\
F_1[\delta_1,\ldots,\delta_n]
\end{pmatrix} \subseteq M,$$

(3)

where

$$M = \begin{pmatrix}
0 & 0 & 0 & 0 & \rho & \rho & \rho & \rho & \sigma & \sigma & \sigma & 1 & 1 & 1 & 1 \\
0 & 0 & \rho & \rho & 0 & 0 & \rho & \rho & \sigma & \sigma & 1 & 1 & \sigma & 1 & 1 \\
0 & \rho & 0 & \rho & 0 & \rho & 0 & \rho & \sigma & 1 & \sigma & 1 & \sigma & 1 & 1 \\
\rho & 0 & 0 & \rho & 0 & \rho & \rho & 0 & 1 & \sigma & \sigma & 1 & \sigma & 1 & 1 & \sigma
\end{pmatrix}.$$ 

(4)

The right-hand side of the relation (3) determines 16 possible cases for $f_1$. Consider term functions $B(p_1,\ldots,p_n)$, defined by the scheme

$$B(p_1,\ldots,p_n) = \begin{cases} I_{23}[f_1], & \text{if } f_1[\alpha_1,\ldots,\alpha_n] \in \{0,1\}, \\ I_{32}[f_1], & \text{if } F_1[\alpha_1,\ldots,\alpha_n] \in \{\rho,\sigma\} \end{cases}$$

Term $B$ is obtained from unary terms and $f_1$ using Malcev operations. It is easy to verify that

$$\begin{pmatrix}
B[\alpha_1,\ldots,\alpha_n] \\
B[\beta_1,\ldots,\beta_n] \\
B[\gamma_1,\ldots,\gamma_n] \\
B[\delta_1,\ldots,\delta_n]
\end{pmatrix} \subseteq \begin{pmatrix}
0 & 0 & 0 & 0 \\
0 & 0 & \rho & \rho \\
0 & \rho & 0 & \rho \\
\rho & 0 & 0 & \rho
\end{pmatrix}.$$ 

(4)

Denote elements $B[\alpha_1,\ldots,\alpha_n]$, $B[\beta_1,\ldots,\beta_n]$, $B[\gamma_1,\ldots,\gamma_n]$ and $B[\delta_1,\ldots,\delta_n]$ by correspondent letters $\alpha$, $\beta$, $\gamma$ and $\delta$. Build term $B'(p,q,r) = B[B'_1,\ldots,B'_n]$, where for any $i = 1,\ldots,n$

$$B'_i(p,q,r) = 0, \ \text{if } \alpha_i = 0, \ \beta_i = 0, \ \gamma_i = 0, \ \delta_i = 0,$$
$p$, if $\alpha_i = 0$, $\beta_i = 0$, $\gamma_i = \rho$, $\delta_i = \rho$,
$q$, if $\alpha_i = 0$, $\beta_i = \rho$, $\gamma_i = 0$, $\delta_i = \rho$,
$r$, if $\alpha_i = 0$, $\beta_i = \rho$, $\gamma_i = \rho$, $\delta_i = 0$,
$I_{37}[r]$, if $\alpha_i = \rho$, $\beta_i = 0$, $\gamma_i = 0$, $\delta_i = \rho$,
$I_{37}[q]$, if $\alpha_i = \rho$, $\beta_i = 0$, $\gamma_i = \rho$, $\delta_i = 0$,
$I_{37}[p]$, if $\alpha_i = \rho$, $\beta_i = \rho$, $\gamma_i = 0$, $\delta_i = 0$,
$\rho$, if $\alpha_i = \rho$, $\beta_i = \rho$, $\gamma_i = \rho$, $\delta_i = \rho$,
$\sigma$, if $\alpha_i = \sigma$, $\beta_i = \sigma$, $\gamma_i = \sigma$, $\delta_i = \sigma$,
$I_{63}(p)$, if $\alpha_i = \sigma$, $\beta_i = \sigma$, $\gamma_i = 1$, $\delta_i = 1$,
$I_{63}[q]$, if $\alpha_i = \sigma$, $\beta_i = 1$, $\gamma_i = \sigma$, $\delta_i = 1$,
$I_{63}[r]$, if $\alpha_i = \sigma$, $\beta_i = 1$, $\gamma_i = 1$, $\delta_i = \sigma$,
$I_{72}[r]$, if $\alpha_i = 1$, $\beta_i = \sigma$, $\gamma_i = \sigma$, $\delta_i = 1$,
$I_{72}[q]$, if $\alpha_i = 1$, $\beta_i = \sigma$, $\gamma_i = 1$, $\delta_i = \sigma$,
$I_{72}(p)$, if $\alpha_i = 1$, $\beta_i = 1$, $\gamma_i = \sigma$, $\delta_i = \sigma$,
$1$, if $\alpha_i = 1$, $\beta_i = 1$, $\gamma_i = 1$, $\delta_i = 1$,

(there are no other cases for $\alpha_i$, $\beta_i$, $\gamma_i$, $\delta_i$). $B'(p, q, r)$ is obtained via Malcev operations, unary terms and $B$. Obviously, by relations (2) and (3), we get $B'_i[0, 0, 0] = \alpha_i$, $B'_i[0, \rho, \rho] = \beta_i$, $B'_i[\rho, 0, \rho] = \gamma_i$ and $B'_i[\rho, \rho, 0] = \delta_i$. Taking into account (4), we obtain

$$
\begin{pmatrix}
B'_i[0, 0, 0] \\
B'_i[0, \rho, \rho] \\
B'_i[\rho, 0, \rho] \\
B'_i[\rho, \rho, 0]
\end{pmatrix}
\subseteq
\begin{pmatrix}
0 & 0 & 0 & 0 \\
0 & 0 & \rho & \rho \\
0 & \rho & 0 & \rho \\
\rho & 0 & 0 & \rho
\end{pmatrix}.
$$

(5)
Examine term \( A(p, q, r) \) defined as

\[
A(p, q, r) = \begin{cases} 
B'(p, q, r), & \text{if } \alpha = \beta = \gamma = 0, \delta = \rho, \\
B'[p, I_{37}[q], I_{37}[r]], & \text{if } \alpha = \beta = \delta = 0, \gamma = \rho, \\
B'[I_{37}(p), q, I_{37}[r]], & \text{if } \alpha = \gamma = \delta = 0, \beta = \rho, \\
I_{37}[B'[I_{37}(p), I_{37}[q], r]], & \text{if } \beta = \gamma = \delta = \rho, \alpha = 0.
\end{cases}
\]

Term \( A(p, q, r) \) is obtained via Malcev operations and terms from lemma. Then, by relations (5), we obtain

\[
A[0, 0, 0] = A[0, \rho, \rho] = A[\rho, 0, \rho] = 0, \ A[\rho, \rho, 0] = \rho. \tag{6}
\]

Since \( A \) conserves relation \( \Box x = \Box y \) on \( \mathfrak{D}4 \), it follows that

\[
A[0, \rho, 0] \in \{0, \rho\}, \ A[\rho, 0, 0] \in \{0, \rho\}.
\]

Thus, taking into account equalities (6), there are 4 possible sub-cases for term \( A \). In each of these sub-cases term \( C(p, q) \) is obtained via Malcev operations, unary terms and terms from lemma and \( A(p, q, r) \) in the following way:

1) If \( A[0, \rho, 0] = 0, A[\rho, 0, 0] = 0 \), then \( C(p, q) = A[p, q, 0] \).
2) If \( A[0, \rho, 0] = 0, A[\rho, 0, 0] = \rho \), then \( C(p, q) = A[A_5(p), p, q] \).
3) If \( A[0, \rho, 0] = \rho, A[\rho, 0, 0] = 0 \), then \( C(p, q) = A[p, A_5(p), q] \).
4) If \( A[0, \rho, 0] = \rho, A[\rho, 0, 0] = \rho \), then \( C(p, q) = A[A_5(p), q, A_5[q]] \).

To finalize the proof of the lemma let us note that \( D(p, q) = I_{62}[C[I_{62}(p), I_{62}[q]]] \). Lemma 1 is proved.

The next three lemmas are proved in a similar way, so we only present them here.

**Lemma 2.** A term functions \( S(p, q) \) satisfying conditions

\[
S[0, 0] = 0, \ S[0, \rho] = \rho, \ S[\sigma, 1] = \sigma, \ S[1, 1] = 1, \tag{7}
\]

is obtained via Malcev operations, unary term functions of \( \mathfrak{D}4 \) and any term function \( f_2 \).
Lemma 3. The term function \( I_{18}[(p \lor q)] \) is obtained via unary terms, Malcev operations and term \( f_3 \) in the algebra \( D_4 \).

Lemma 4. Conjunction term function \((p \& q)\) is obtained in algebra \( D_4 \) via terms \( C(p, q) \) and \( D(p, q) \) satisfying conditions (1), via term \( S(p, q) \) which respects restrictions (7) and term \( I_{18}[(p \lor q)] \) using Malcev operations and unary terms.

A system \( \Sigma \) of term functions of \( D_4 \) is ultra-weak complete relative to expressibility in \( D_4 \) if any term function of \( D_4 \) can be obtained from unary term functions and functions of \( \Sigma \) via Malcev operations \( \{\zeta, \tau, \Delta, \nabla, *\} \) defined earlier.

Corollary. There is a relative simple algorithm to test whether a system of term functions of \( D_4 \) is ultra-weak complete relative to expressibility of term functions in \( D_4 \).

The algorithm for testing ultra-weak completeness of a system \( \Sigma \) of term functions in \( D_4 \) is based on two main steps: consider the union of \( \Sigma \) with unary term functions and then verify that the resulting system is not contained in any of the classes \( K_{38}, K_{39}, \text{ or } K_{40} \).

4 Conclusion

Further investigations of Slupecki iterative Post algebras of diagonalizable algebras are under consideration. Since diagonalizable algebras are algebraic models for propositional provability logic of Gödel-Löb \( GL \) and its extensions, maximal Slupecki iterative Post algebras may tell interesting things about functional properties of the propositional provability logic \( GL \). A way to investigate these questions is to move from relative simple cases to more difficult ones, i.e. to extend the study to different slices [13] of extensions of \( GL \) (a slice here is something similar to corresponding Hosoi slices of propositional superintuitionistic logics [14, 15, 16, 17]).

Acknowledgments. National Agency for Research and Development has supported part of the research for this paper through the
research project 15.817.02.02A ”Models and Technologies for Intelligent Systems and High Performance Computing”.

References


Andrei Rusu$^{1,2}$, Elena Rusu$^{3}$

$^1$Vladimir Andrunachievici Institute of Mathematics and Computer Science  
andrei.rusu@math.md

$^2$Ovidius University of Constanta  
agrusu@univ-ovidius.ro

$^3$Technical University of Moldova  
elena.rusu@mate.utm.md
Some Developments of Aggregate Theory

Ioachim M. Drugus, Volodymyr G. Skobelev

Abstract

Aggregate theory, intended to serve as a framework for representation of data structures, earlier introduced by first author, has been further developed by the two authors, and the results of their research are shared in this paper.

Keywords: atom, class, set, generalized Boolean algebra, ordered pair, data representation

1 Introduction

The aggregate theory was introduced in [1] as a mathematical foundation for data structures. This is a weak algebraic set theory, which admits atoms and primitive ordered pairs (i.e. ordered pairs treated as primitive entities, and not a special kind of sets). ‘Aggregate’ is a common name for sets, atoms, and ordered pairs, and in aggregate theory these three notions are treated on equal footing. This enlargement of the inventory of types of the objects residing within the universe of discourse was a reason for the choice of a name for this theory different from set theory.

Another reason for the choice of this name is historical – ‘aggregate theory’ was the name used for the set theory in the first English translation of Cantors pioneering paper on set theory. Notice, that in Cantor set theory, alongside sets, also atoms (which Cantor called ‘points’) and ordered pairs (see ‘Cantor’s pairing function’) were used. The aggregate theory introduced in [1] differs from Cantor set theory mainly in the manner of presentation - the modern variant of aggregate
theory in [1] was presented axiomatically and in algebraic manner i.e. with algebraic identities and quasi-identities as axioms.

The fact that, unlike a pure set theory, the aggregate theory operates with all three main types of objects of set theory is one of the reasons why this theory is expected to be an appropriate framework for representing any kind of data structures, and not only of those which are called “mathematical structures”. The fact that the aggregate theory is algebraic is a reason for the expectation that a computing framework for data structures based on this theory would be more constructive than other such frameworks.

The aggregate theory, as it was initially specified, is the weakest among mainstream set theories, i.e. the theorems of this theory, rewritten in the standard language of set theory, are easily deducible in any of the mainstream set theory (simplistically, the axioms of this theory postulate the properties of finite unions, singletons and ordered pairs), and in this sense, this theory is the common part of the mainstream set theories. This is yet another reason for the belief that other theories useful for computer science can be developed from it, as this theory’s extensions or as its variants. To put it in other words, the initial axiom system can be further developed both to increase the theory and, thus, to describe a narrower class of aggregates, or even modified to describe aggregates which reflect a slightly different conception. The results of such developments form the material of the current paper.

Whereas this paper is a common work of both authors, the first author focused on the “narrative” introductory sections 1-4 of the paper, describing the conceptual aspect of aggregate theory and on terminology appropriate for this theory, and the second author focused on the development of the logic of an operation used in this theory.

2 Aggregate Algebras

Aggregate theory is the equational theory of universal algebras called aggregate algebras [1]. Since this theory can be axiomatized in different manners, and in the language of different signatures, here the aggregate
algebras will be defined informally in algebraic terms.

Before proceeding to the definition of aggregate algebras, recall that a generalized Boolean algebra, GBA (or “generalized Boolean lattice”) is a distributive lattice with 0, which is relatively complemented ([2], p. 143). Notice, that in such algebra each principal ideal is a Boolean algebra ([2], p. 11).

**Definition 1.** An aggregate algebra is a GBA, equipped with a unary operation called “successor”, with operation symbol ‘◦’, and a binary operation called “association” with no operation symbol (“empty operator”), such that the identities below hold:

\[ a \circ = b \circ \Rightarrow a = b, \]
\[ (a, b) = (a', b') \Rightarrow a = a' \& b = b'. \]

The aggregate theory is the equational theory of the class of these algebraic structures (the modifier “equational” refers to the fact that the axioms for equality should be added to the theory).

### 3 Aggregate Theory as a Framework for Data Representation

In mathematics, there are “representation theorems” (and even “representation theories”) for various classes of mathematical structures, like groups, boolean algebras, Lie algebras - theorems, which describe how such structures can be represented through less complex structures, which can be finally represented as sets.

However, “representation of a set” does not even sound as a correct term for theory of sets, which treats the notion of set as fundamental and focuses only on description of this notion through various axiomatic systems - an approach which defies any kind of representation (description vs representation). However, the notion “set representation” must, obviously, be playing in computer science preoccupied by “data representation” a role similar to that of the notion of set in mathematics.
Similarly, fundamental for computer science is the notion of representation of ordered pairs (since this notion is strongly exploited in LISP programming language); less clear sounds such a statement on fundamentality of the representation of atoms. The intuitive vision which stood in foundation of aggregate theory is strongly based on the notion of data representation. This vision is described summarily in the next paragraph.

Per our treatment, data is what is “given” (see Latin “datum” with plural “data”) to an agent for processing - it is the “matter” which emerges at agents “input”, as well as the “matter” which emerges at agents “output” (to be ”given” to other agents for further processing). The aggregates are individual entities of which consists the “matter” processed by the agents. A notion, e.g. the notion of set, resides in minds, and it must be represented in a certain form to become “data”. Thus, any theory intended to explain phenomena of data structures has to do with “representation” (whatever this might be). One can say that data structures, in contrast with mathematical structures, are necessarily results of some representation.

4 Representations of Sets

In a Venn diagram, a finite set is usually represented as a contour enclosing the representations of its elements. In text, a finite set is usually represented as a list of denotations of its elements, separated by commas, and enclosed between braces ‘{’, ‘}’ – a kind of contour in the linear space of text. According to these two representations on paper, a set can be imagined (i.e. “represented in mind”) as an “abstract container”, which “contains” the elements of the set making up its “content”. The term “abstract container” appears to be appropriate also for the applications of set theory, as this can be illustrated by the example coming next (an example of set used in a discussion between Dedekind and Frege) - the constellation Orion.

Consider the constellation Orion as a set of stars, where the stars are treated as elements of the set. The stars are “givens”, they are
“data” – but where in this picture is the set itself? One can say that the set will be also “given”, if an imaginary contour is drawn around the stars – an abstract container. The stars are material bodies, but the set is an “abstract container” added to the picture in human mind – these two kinds of entities live in different universes - the material Cosmos and the Mind. This illustration shows that abstract container is not element of the set. Here, we will be using the following terminology: the set will be said to be the “contents” of the abstract container. The set without the abstract container is an object different from the set, and per the practice of [3], this object be called “multitude” (see the Figure 1).

![Figure 1. Representation of a set.](image)

In all three kinds of representations of a set – Venn diagrams, denotation in text, image in mind – in addition to the elements of a set, there is also an additional “structural element” which deserves to have a name; we will refer to it as identity of the set (this kind of “identity” has to do with identification and has nothing to do with “algebraic identity”). Thus, the contour within a Venn diagram, as well as the two braces which enclose a list, and the abstract container in mind - all these are “identities” of sets. The extensionality axiom is not falsified by existence of an additional structural element, because the identity of a set is not an element of the set, it is a part of the representation of the set.

Various notations are used in the linear space of text, Venn diagrams are drawn on the plane, and there is a space of mental presen-
tation, which sounds appropriate to be treated as dimension-less. The terminology can be changed according to the space for representation in order to be descriptive. Really, the term “container” is a good counterpart for the term “identity”, when the presentation is done in plane or 3D space, but it is not descriptive for the dimension-less presentation in mind, where the content cannot be said to be “contained” or “enclosed” in a “container”. However, we can always say that a (piece of) “content” has an “identity”, or that it is “identifiable”. “Identifier” is not the same as “identity”. These two notions can be distinguished by imagining the identity of a thing as an attribute (a variable), and identifier as a value of the attribute).

To generalize the discourse above, one could distinguish between the notions “entity” and “object” and regard “entity” as most general notion, whereas “object” as a partial case of entity – an “identified entity”, an entity which has an identity. This entity will then be said to be the “content” of the object (even though “content” would be expected to be part of the “content-container” opposition, and not of the “content-identity” opposition). However, the linguists similarly talk about “content” of a proposition, even though generally a proposition is not treated as a container.

Attaching an identity to an entity \( e \), in result of which it becomes an object, is treated here as application of a unary operation named “individuation”, and the result of application of this operation is denoted as \( e^\circ \). The sign \( ^\circ \) used here symbolizes the “abstract container”. The successor operation is intended to explicate the mental operation of enclosing an entity (e.g. a ”multitude”) into an “abstract container” (to obtain a set).

The use of the term “individuation” can be justified like this: set theorists use to call “individual” or “object” of a set theory any entity which resides in its universe of discourse – be it atom, set, (primitive) ordered pair, or something else. Even though, usually, “object” is treated as most general term, in previous sentence “individual” and “object” are treated as synonyms, and the discussion about “objects” in the paragraph before next can be treated as a discussion about
“individuals”. The braces ‘{’, ‘}’ in denotation of a set symbolize the application of individuation operation - an operation which maps into the universe of discourse (where “live” the “individuals”).

The codomain (domain of variation) of the individuation operation is the universe of discourse, but which is its domain (of definition)? Per the vision behind the “aggregate theory”, the individuation operation is defined on classes (which are subclasses of the universe of discourse) – a topic (see [3]), which requires a special treatment and will not be further discussed here.

Even though the term “individuation” is useful for the applications, the term “successor” is more appropriate for theory, and this term will be used from this point on.

5 Successor Algebra

In aggregate theory, we investigate the algebra $\mathfrak{A} = (A; \mathcal{F})$ – an algebraic system, where $A$ is a fixed non-empty set and $\mathcal{F}$ is the set of its operators, among which there is the successor operator $\circ - a$ symbol of unary operation. The reduct of this algebra with only the successor operator in its signature is referenced here as “successor subalgebra”.

We denote $a^{n\circ} (a \in A, n \in \mathbb{Z}^+)$ the element $(\ldots(a^{\circ})^{\circ}\ldots)^{\circ}$ of the set $A$, such that the operation “$\circ$” is applied $n$ times. Thus, $a^{0\circ} = a$ for all elements $a \in A$. Similarly, we denote $a^{-m\circ} (a \in A, m \in \mathbb{N})$ the element $b \in A$, if it exists, such that $b^{m\circ} = a$.

Let $\mathfrak{A}_1 = (A; \{\circ\})$ be the subalgebra of the algebra $\mathfrak{A}$, such that the following three axioms hold:

Axiom 1. For any element $a \in A$ the infinite sequence

$$a, a^{\circ}, a^{2\circ}, \ldots$$

(1)

consists of pair-wise different elements of the set $A$.

Axiom 2. For any elements $a, b \in A$ the following formula is true:

$$a^{\circ} = b^{\circ} \Rightarrow a = b.$$ 

(2)
Axiom 3. For any element \( a \in A \) there exist some element \( b \in A \) and an integer \( n \in \mathbb{Z}_+ \), such that \( a = b^n \), and besides \( b \neq c^m \) for each element \( c \in A \) and each integer \( m \in \mathbb{N} \).

Let’s note some immediate consequences of axioms 1-3.

Axiom 1 implies that the following three propositions are true.

**Proposition 1.** The set \( A \) is, at least, a countable set.

**Proposition 2.** The non-equality \( a \neq a^o \) holds for each element \( a \in A \).

**Proposition 3.** For each element \( a \in A \) the mapping \( a^n \to n (n \in \mathbb{N}) \) is an isomorphism between the sequence (1) and the sequence \( 0, 1, 2, \ldots \).

Let \( A^{(0)} = \{ a \in A | (\forall b \in A)(\forall n \in \mathbb{N})(b^n \neq a) \} \). (3)

Axiom 3 and formula (3) imply that the following proposition is true.

**Proposition 4.** The set \( A^{(0)} \) is some non-empty subset of the set \( A \).

Let
\[
A^{(i+1)} = \{ a^o | a \in A^{(i)} \} \ (i \in \mathbb{Z}_+) . \tag{4}
\]

Axiom 3, Proposition 4 and formula (4) imply that the following proposition is true.

**Proposition 5.** The sets \( A^{(i)} \) \((i \in \mathbb{Z}_+)\) are non-empty pair-wise disjoint subsets of the set \( A \), such that the following identity is true
\[
A = \bigcup_{i=0}^{\infty} A^{(i)} . \tag{5}
\]

Let ”\( \prec \)” be the ordering relation defined on the set \( A \) as follows: for any element \( a \in A^{(0)} \) we set:
\[
(\forall a \in A^{(0)})(\forall m, n \in \mathbb{Z}_+)(a^m \prec a^n \iff m < n) . \tag{6}
\]

Due to Proposition 1, 3, and 5, and formula (6), there are no infinite decreasing sequences in the set \( A \), and for any element \( a \in A^{(0)} \) the
sequence (1) is an infinite increasing sequence. Therefore, the mathematical induction technique can be applied as a mathematical proof on the set $A$. Applying this technique, and Axiom 2, the following proposition can be proved.

**Proposition 6.** For any elements $a, b \in A$ ($a \neq b$) the non-equality $a^{n_0} \neq b^{n_0}$ holds for all $n \in \mathbb{Z}_+$. Let

$$
\text{Cl}_0(a) = \{a^{n_0} | a \in A\}. \tag{7}
$$

Applying the mathematical induction technique, Propositions 2, 4 and 5, and formula (7), the following proposition can be proved.

**Proposition 7.** The sets $\text{Cl}_0(a)$ ($a \in A^{(0)}$) are non-empty pair-wise disjoint subsets, such that the following identity is true

$$
A = \bigcup_{a \in A^{(0)}} \text{Cl}_0(a). \tag{8}
$$

Summing up the result of the constructions which are carried out above, we conclude that:

1) the operation $\circ$ is the successor operation with natural interpretation $a^\circ = \{a\}$ in the Sets Theory, and the terms $\ldots \{a\} \ldots$ with infinite number of brackets are inadmissible;

2) the algebra $\mathfrak{A}_1 = (A; \{^{\circ}\})$ is the successor algebra, in which the mathematical induction technique can be applied as a mathematical proof.

6 Conclusions

Due to its algebraic form of presentation, the aggregate theory offers a constructive conceptual foundation for a computer framework for data structures management.

Due to a larger inventory of “building blocks” (sets, atoms, ordered pairs, multitudes) than the mainframe set theories, the aggregate theory is better suited than set theory for the role of foundation for “computer algebra”.

179
The aggregate theory is well suited both for point-free (continuous) mathematics – since it has no foundation axiom, and for the pointy (descrete) mathematics – you can postulate the Axiom 3 above or a similar one.

References


General Boolean Algebras, Boolean Rings and Extension Algebras

Kuznetsov Elena, Kuznetsov Eugene

Abstract

General Boolean algebras, Boolean rings and extension algebras are studied. The one-to-one correspondence between these objects are investigated.

Keywords: Boolean algebras, Boolean rings, extension algebras.

1 Introduction

Generalized Boolean algebras are one of the main algebraic objects used in the theory of modelling of artificial intelligence. These algebras are a natural generalization of classical Boolean algebras. They were introduced by M. Stone in [1, 2]. In the same place, he proved a one-to-one connection of these algebras with rings, namely, with Boolean rings (without unit element). He also proved the so-called Stone’s duality: if, from a generalized Boolean algebra, we construct the corresponding Boolean ring, and from the last, the corresponding new generalized Boolean algebra, then it coincides with the original.

In his works, J. Drugush introduced and studied a concept of extent algebra, which is also used for the theoretical modelling of artificial intelligence systems. The definition of this algebra can be found in [3]. The natural question arises about the correspondence between extent algebras and the above-mentioned generalized Boolean algebras and Boolean rings.
The authors of this paper managed to establish that extent algebras are in one-to-one correspondence with Boolean rings, and thus with generalized Boolean algebras. This is reflected in theorems 3.1 – 4.4.

2 Generalized Boolean Algebras

It is of some interest to examine the possibility of introducing a system with double composition which possesses most of the peculiar properties of Boolean algebras without containing a unit element.

The analogy with the theory of abstract rings suggests that the direct or indirect postulation of the unit element should be avoided; and examples from the theory of classes (for instance, the example of all the finite subclasses of a given infinite class, or the example of all the Lebesgue or Borel measurable sets of finite measure in \( n \)-space) indicate the existence of interesting algebras without unit. We shall therefore introduce the following definition:

**Definition 2.1.** A generalized Boolean algebra \( A \) is a system with double composition which satisfies the following postulates:

- **Postulate 1.** \( a \lor b = b \lor a \);
- **Postulate 2.** \( a \land (b \land c) = (a \land b) \land c \);
- **Postulate 3.** \( a(b \lor c) = a \land b \lor a \land c \);
- **Postulate 4.** There exists an element 0 in \( A \) such that \( a \lor 0 = a \) for every element \( a \) in \( A \);
- **Postulate 5.** If there exists an element 0 with the property required by Postulate 4 and if \( a \) and \( b \) are elements of \( A \) such that \( b \land a = a \), then there exists at least one such an element 0, independent of \( a \) and \( b \), that the simultaneous equations \( x \lor a = b \), \( x \land a = 0 \) have a solution in \( A \);
- **Postulate 6.** If there exists an element 0 with the property required by Postulate 4 and if \( a \) and \( b \) are elements of \( A \) such that \( a \land b = a \), then there exists at least one such element 0, independent of \( a \) and \( b \), for which the simultaneous equations \( x \lor a = b \), \( a \land x = 0 \) have a solution in \( A \);
- **Postulate 6.** \( a \lor a = a \);
Postulate 6.2. \( a \land a = a \).
Any element 0 satisfying Postulate 4 is called a zero.

3 Boolean Rings

Definition 3.1. A ring in which every element is idempotent, satisfying the law \( aa = a \), is called a Boolean ring.

Theorem 3.1. If \( A = (\Omega, +, \cdot, 0) \) is a Boolean ring, either with or without unit, the replacement of the operation + by the operation \( \lor \) defined by the relation
\[
(1) \quad a \lor b = a + b + ab,
\]
and the operation \( \cdot \) by the operation \( \land \) defined by the relation
\[
(2) \quad a \land b = a \cdot b,
\]
converts \( A \) into a system \( B = (\Omega, \land, \lor, 0) \) with the properties
\[
(1_1) \quad a \lor b = b \lor a;
\]
\[
(2_2) \quad a(bc) = (ab)c;
\]
\[
(3_1) \quad a(b \lor c) = ab \lor ac;
\]
\[
(4_1) \quad \text{there exists an element 0 such that } a \lor 0 = a \text{ for every } a;
\]
\[
(5_1) \quad \text{if } ba = a, \text{ there exists an element 0 with property } (4_1), \text{ independent of } a \text{ and } b, \text{ such that the equations } x \lor a = b, xa = 0 \text{ have a solution};
\]
\[
(5_2) \quad \text{if } ab = a, \text{ there exists an element 0 with the property } (4_1), \text{ independent of } a \text{ and } b, \text{ such that the equations } x \lor a = b, ax = 0 \text{ have a solution};
\]
\[
(6_1) \quad a \lor a = a;
\]
\[
(6_2) \quad aa = a;
\]
\[
(7) \quad a+b \text{ is a solution, necessarily unique, of the simultaneous equations}
\]
\[
x \lor ab = a \lor b, x(ab) = 0.
\]

Conversely, if \( B \) is a system with the indicated properties \((1_1)-(6_2)\), the replacement of the operation \( \lor \) by the new operation + defined by the relation \((7)\) and the operation \( \land \) by the new operation \( \cdot \) defined by the relation \((2)\), converts \( B \) into a Boolean ring \( A \) with the element 0.
of (41) as its zero element, the old operation ∨ is expressed in terms of
the new by the relation (1).

This theorem serves to identify Boolean rings with those systems
which we have called generalized Boolean algebras.

Let us denote as $A \xrightarrow{\mathcal{L}} A^*$ a functor of one-to-one correspondence
from general Boolean algebra to Boolean ring.

Let us denote as $A \xrightarrow{\mathcal{R}} A^*$ a functor of one-to-one correspondence
from Boolean ring to general Boolean algebra.

**Theorem 3.2.** If $A$ is a GBA, then $(A^\mathcal{L})^\mathcal{R} = A$.

**Theorem 3.3.** If $A^*$ is a BR, then $(A^*^\mathcal{R})^\mathcal{L} = A^*$.

# 4 Extensional Algebras

**Definition 4.1.** An extensional algebra (EA) $(A, +, \cdot)$ is a system
with double composition satisfying the following axioms:

1. $a - a = 0$,
2. $a + (b + c) = (a + b) + c$,
3. $a + b = b + a$,
4. $a + a = a$,
5. $(a + b) - c = (a - c) + (b - c)$,
6. $a - (b + c) = (a - b) - c$,
7. $a + (b - a) = a + b$,
8. $a + (a - b) = a$,
9. $(a - b) - c = (a - c) - (b - c)$,
10. $a - (b - c) = (a - b) + (a - (a - c))$

**Theorem 4.1.** Let $A = (\Omega, +, \cdot, 0)$ is a boolean ring, namely:
1. $(\Omega, +, 0)$ is a commutative group with all the elements of order 2;
2. $(\Omega, \cdot)$ is an idempotent commutative semigroup, $a \cdot a = a$;
3. $x \cdot 0 = 0 \cdot x = 0, \forall x \in \Omega$;
4. There are two distribution laws:
   
   a. $(x + y) \cdot z = x \cdot z + y \cdot z$ (distributivity to the right)
b. \( z \cdot (x + y) = z \cdot x + z \cdot y \) (distributivity to the left).

Then, if we define 2 new operations:

\[
\begin{align*}
     a \ominus b &= a + (a \cdot b) \\
     a \oplus b &= a + b + (a \cdot b)
\end{align*}
\]

we have \( B = (\Omega, \ominus, \oplus, 0) \) is an extensional algebra (EA).

**Theorem 4.2.** Let \( B = (\Omega, \ominus, \oplus, 0) \) be an extensional algebra (EA).

Let

\[
\begin{align*}
     a \cdot b &= a \ominus (a \ominus b) \\
     a + b &= (a \lor b) \oplus (a \land b) = (a \oplus b) \ominus (a \ominus (a \ominus b))
\end{align*}
\]

It can be proven that \( A = (\Omega, +, \cdot, 0) \) is a boolean ring, namely:

1. \( a + a = 0 \) (elements of order 2);
2. \( a + b = b + a \) (commutativity);
3. \( (a + b) + c = a + (b + c) \) (associativity);
4. \( 0 + a = a + 0 = a \);
5. \( a \cdot a = a \);
6. \( a \cdot b = b \cdot a \);
7. \( (a \cdot b) \cdot c = a \cdot (b \cdot c) \);
8. \( 0 \cdot a = a \cdot 0 = 0 \);
9. \( (a + b) \cdot c = a \cdot c + b \cdot c \).

Let us denote as \( B \overset{\mathcal{E}}{\rightarrow} B^* \) a functor of one-to-one correspondence from Boolean ring to extensional algebra.

Let us denote as \( B^* \overset{\mathcal{F}}{\rightarrow} B \) a functor of one-to-one correspondence from extensional algebra to Boolean ring.

**Theorem 4.3.** If \( B \) is a BR, then \( (B^E)^F = B \).

**Theorem 4.4.** If \( B^* \) is a EA, then \( (B^*^F)^E = B^* \).
5 Conclusion

The main problem was to find a correspondence between generalized Boolean algebras and extent algebras.

The authors of this paper had established that extent algebras are in one-to-one correspondence with Boolean rings and with generalized Boolean algebras, which was reflected in the Theorems 3.1 – 4.4.

References


Kuznetsov Elena¹, Kuznetsov Eugene²

¹Moldavian State University, MECC, Republic of Moldova
Email: elenacuznetova@rambler.ru

²Institute of Mathematics and Computer science, MECC, Republic of Moldova
Email: kuznet1964@mail.ru
Completeness of the Logic of Partial Quasiary Predicates with the Complement Composition

Mykola Nikitchenko, Oksana Shkilniak, Stepan Shkilniak, Tohrul Mamedov

Abstract

Partial quasiary predicates are mappings with non-fixed arity. They appear in a natural way in programming, where they represent program semantics, and in logic, where they formalize predicates over partial variable assignments. To increase expressibility of logic, oriented on such predicates, we extend the logic language with the complement operation (composition). Obtained logic is called first-order logic of partial quasiary predicates with the complement composition. We study properties of this logic, construct a sequent calculus for it and prove its completeness.

Keywords: partial predicate, quasiary predicate, program logic, predicate logic, soundness and completeness.

1 Introduction

Extensive usage of formal methods in Computer Science, Artificial Intelligence, and Software Development [1] leads to new logics that allow more adequate investigation of applied domains. Logic of partial quasiary predicates is one of such logics oriented on software verification. The class of partial quasiary predicates also appears in a natural way in other domains, in particular, in logic where it can be used for formalization of predicates defined over partial variable assignments. Algebras of such predicates serve as semantic base of logics of applied domains. An important question concerns expressibility of logic languages. It often happens that a chosen language is not expressive enough for effective usage. This question also concerns logics of partial quasiary predicates.
In our previous works [2, 3] we studied logics with tradition compositions of disjunction, negation, renomination, existential quantification. Application of such logics to software verification, in particular, to Floyd-Hoare program logic [4, 5], demonstrated that the logics are not expressive enough to construct a sound axiomatic system. This problem appeared due to necessity of introducing partial pre- and postconditions into Floyd-Hoare logic. Initially, this logic treats pre- and postconditions as total predicates, but being extended on class of partial predicate the logic becomes unsound [6, 7]. There are different methods to solve this problem, in particular, a sound axiomatic system can be constructed for the logic language extended with the complement composition (detailed discussion of the topic is presented in [8–10]). Introduction of this composition permits to modify rules of Floyd–Hoare logic in such a way that they become sound, but a negative side of this proposal is that the logic becomes more complicated. In this case, undefinedness conditions for predicates should be taken into account.

In [11] we constructed a sound and complete sequent calculus for a logic of renominative (quantifier-free) level. Here we generalize the obtained results for the first-order logic of partial quasiary predicates extended with the complement composition. We additionally study semantic properties of quantifier elimination, of variable assignment composition, of a ternary consequence relation with undefinedness conditions. We define a sequent calculus for this logic and prove its soundness and completeness.

Obtained results can be applied for software verification.

We use the following notations:  \( S \rightarrow^p S' \) (\( S \rightarrow^t S' \)) is the class of partial (total) mappings from \( S \) to \( S' \); \( p(d)\downarrow \) (\( p(d)\uparrow \)) means that \( p \) is defined (undefined) on \( d \). The terms and notations, not defined here, are treated in the sense of [10].

Proofs of lemmas and theorems are omitted in this paper.

2 First-order Logic of Partial Quasiary Predicates with the Complement Composition

We treat a logic \( L \) as a tuple \((\mathcal{A}, Fr, \mathcal{I}, \models, \models)\) [2, 3] where
A is a class of algebras of some signature;  
Fr is a language (based on the algebra signature);  
T is a class of interpretations;  
|= is a consequence relation;  
|– is an inference relation based on some calculus.

Here we consider only pure (without functions) logic \( L^{QEC} \). This logic is the next step of our construction of series of first-order logics of partial quasiary predicates.

Earlier, we started with a basic logic \( L^Q \) with compositions of disjunction \( \lor \), negation \( \neg \), renomination \( R_x^v \), and existential quantification \( \exists x \) [2]. This logic was not expressive enough to prove its completeness therefore a logic \( L^{QE} \) was constructed as an extension of \( L^Q \) with the null-ary parametric composition (predicate) \( E_z \) of variable assignment. (Also, variable unassignment predicate \( e_z \) can be used [10].) But again, logic \( L^{QE} \) was not expressive enough to construct sound program logics of Floyd-Hoare type therefore \( L^{QE} \) is extended to a new logic \( L^{QEC} \) by adding the composition of predicate complement \( \sim \) (discussion on the topic is presented in [9]).

2.1 Predicate Algebras with the Complement Composition

Let \( V \) and \( A \) be sets of names (variables) and values respectively. The class \( V^A \) of nominative sets (partial assignments, partial data) is defined as the class of all partial mappings from \( V \) to \( A \), thus, \( V^A = V \rightarrow^p A \).

The main operation for nominative sets is a total unary parametric renomination \( r_{v_1,\ldots,v_n}^{v_1,\ldots,v_n} : V^A \rightarrow^t V^A \), where \( v_1,\ldots,v_n, x_1,\ldots,x_n \) are names, and \( v_1,\ldots,v_n \) are distinct [2, 3]. Intuitively, given a nominative set \( d \) this operation yields a new nominative set changing the values of \( v_1,\ldots,v_n \) to the values of \( x_1,\ldots,x_n \) respectively. For this operation we also use simpler notation \( r_x^v \); \( x \in \overline{v} \) means that \( x \) is a variable from \( \overline{v} \); \( \overline{v} \cup \overline{x} \) is the set of variables that occur in \( \overline{v} \) and \( \overline{x} \); \( asn(d) \) is the set of assigned variables (names) in \( d \).
The set $Pr^V_A = V A \rightarrow^p \text{Bool}$ is called the set of *partial quasiary predicates*. For a predicate $p \in Pr^V_A$ its *truth*, *falsity*, and *undefinedness domains* are denoted $T(p)$, $F(p)$, and $\perp(p)$ respectively. Note, that a predicate $p$ is defined by $T(p)$ and $F(p)$ only, because $\perp(p) = V A \setminus (T(p) \cup F(p))$.

A name (variable) $z$ is *unessential* for $p \in Pr^V_A$, if for any $d \in V A$ the value of $p$ does not depend on the value of $z$ [10].

Operations over $Pr^V_A$ are called *compositions*. Basic compositions of first-order level of quasiary predicates are *disjunction* $\lor$, *negation* $\neg$, *renomination* $R^V_x$, and *existential quantification* $\exists x$.

We define them via their definedness domains ($p, q \in Pr^V_A$):
- $T(p \lor q) = T(p) \cup T(q)$, $F(p \lor q) = F(p) \cap F(q)$;
- $T(\neg p) = F(p)$, $F(\neg p) = T(p)$;
- $T(R^V_x(p)) = \{ d \in V A \mid r^V_x(d) \in T(p) \}$, $F(R^V_x(p)) = \{ d \in V A \mid r^V_x(d) \in F(p) \}$;
- $T(\exists x p) = \{ d \in V A \mid d \forall x \mapsto a \in T(p) \text{ for some } a \in A \}$, $F(\exists x p) = \{ d \in V A \mid d \forall x \mapsto a \in F(p) \text{ for all } a \in A \}$.

Please note that definitions of disjunction and negation are similar to *strong Kleene’s connectives*; their properties are described in [12].

Also we use *variable assignment predicate* $Ez$ defined as follows:
- $T(Ez) = \{ d \mid d(z) \downarrow \} = \{ d \in V A \mid z \in \text{asn}(d) \}$,
- $F(Ez) = \{ d \mid d(z) \uparrow \} = \{ d \in V A \mid z \notin \text{asn}(d) \}$.

At last, the complement composition is defined in the following way:
- $T(\neg p) = \perp(p)$, $F(\neg p) = \emptyset$.

We consider $\neg$ as a composition of propositional level. This composition differs from traditional compositions. The main difference lies in the fact that traditional compositions are applicative composition [13]. *Applicativity* of composition $C$ means that given predicates $p_1, \ldots, p_n$ the value of $C(p_1, \ldots, p_n)$ on some data is evaluated upon values of $p_1, \ldots, p_n$ on data from their definedness domains. The complement composition is not applicative because the value of $\neg p$ on some $d$ may
depend upon undefinedness domain of $p$. This fact complicates logics with such composition because the undefinedness domains should be explicitly involved in the definitions of consequence relations.

Note, that applicative compositions are monotone with respect to predicate graph inclusion; but composition $\sim$ is not.

A tuple $A^{QEC}(V, A) = \langle Pr_A^V, \lor, \neg, R_x^\top, \exists x, Ez, \sim \rangle$ is called a first-order complemented algebra of partial quasiary predicates.

A class of such algebras (with different $A$) forms a semantic base for logic $L^{QEC}$.

Now we describe the main properties of $A^{QEC}(V, A)$. We concentrate on the complement composition and do not formulate properties of disjunction, negation, renomination, existential quantification, and unassignment predicate [2, 10, 11].

**Lemma 1.** For any $p \in Pr_A^V$ we have

\begin{itemize}
  \item $\neg \sim p$; \quad $\neg \neg \neg \neg \neg p = \sim p$; \quad $\neg \neg \neg \neg \neg \neg p = \sim p$
  \item $T(\neg \sim p) = \emptyset$; \quad $F(\neg \sim p) = T(\sim p)$; \quad $\bot (\neg \sim p) = \bot (\sim p)$
  \item $T(R_x^\top (\sim p)) = T(\sim (R_x^\top (p)))$; \quad $F(R_x^\top (\sim p)) = \emptyset$
  \item $\bot (R_x^\top (\sim p)) = \bot (\sim (R_x^\top (p)))$
  \item $T(\neg \exists xP) = \bot (\exists xP)$; \quad $F(\neg \exists xP) = \emptyset$; \quad $\bot (\neg \exists xP) = T(\exists xP) \cup F(\exists xP)$
  \item $R_x^{\top, z, v}(Ez) = Ez$; \quad $R_x^{\top}(Ez) = Ez$, if $z \notin \{v\}$
\end{itemize}

### 2.2 Language (signature and formulas) of $L^{QEC}$

Let $V$ be an infinite set of names (variables) and $U$ be an infinite subset of $V$ called a set of unessential variables [10]. Let $Ps$ be a set of predicate symbols. A tuple $\Sigma^{QEC} = (V, U; \lor, \neg, R_x^\top, \exists x, Ez, \sim; Ps)$ is called the language signature.

For simplicity, we use the same notation for symbols of compositions and compositions themselves.

Given $\Sigma^{QEC}$, we define inductively the language of $L^{QEC}$ – the set of formulas denoted $Fr(L^{QEC})$ or simply $Fr$:

- if $P \in Ps$ then $P \in Fr$. Formulas of such forms are called atomic;
Formulas of the form $R^\varphi_x \Phi$ are called R-formulas.

2.3 $L^{QC}$-interpretations

Let $\mathcal{A}^{QC}(V, A) = < Pr^V_A; \lor, \neg, R^\varphi_x, \exists x, Ez, \sim >$ be a first-order complemented algebra of partial quasary predicates of a signature $\Sigma^{QC} = (V, U; \lor, \neg, R^\varphi_x, \exists x, \sim; Ps)$; $I^{Ps}_Q = Ps \rightarrow Pr^V_A$ be an interpretation mapping of predicate symbols that respects the set $U$ of unessential variables. Then a pair $J(\Sigma^{QC}) = (\mathcal{A}^{QC}(V, A), I^{Ps}_Q)$ is called an $L^{QC}$-interpretation. Note that this definition of interpretation is quite natural because the algebra $\mathcal{A}^{QC}(V, A)$ defines interpretations of composition symbols (logical symbols) and $I^{Ps}_Q$ defines interpretations of predicate symbols (descriptive symbols).

We simplify notation for $L^{QC}$-interpretation $J(\Sigma^{QC})$ omitting $L^{QC}$ and $\Sigma^{QC}$.

For a given interpretation $J$ and a formula $\Phi$, we can define by induction on the structure of $\Phi$ its value in $J$. Obtained predicate is denoted $\Phi_J$.

For $L^{QC}$ the notion of equivalent formulas becomes more complicated because of presence of unassigned and unessential variables.

Let $Un \subseteq V$ be a set of names treated as unassigned. Let $R^{\varphi, \pi, \gamma}_x \Phi$ be $R$-formula such that $\varphi \cup \pi \cup \gamma \subseteq Un$ and $(\varphi \cup \gamma) \cap Un = \emptyset$. Then expression $R^{\varphi, \pi, \gamma}_x \Phi$, where $\varepsilon \notin V$, is called $Un$-form of $R^{\varphi, \pi, \gamma}_x \Phi$. Here $\varepsilon$ indicates that a value of corresponding variable is not assigned. If $\Psi \in Fr$ is not $R$-formula, then its $Un$-form coincides with $\Psi$.

For any formula $\Phi$ its set $v(\Phi)$ of derived unessential variables is defined inductively:

- $v(P) = U$ for $P \in Ps$;
- $v(Ez) = \{x \in V \mid x \neq z\}$;
Completeness of the Logic of Partial Quasiary Predicates …

\[ - \nu(\Phi \lor \Psi) = \nu(\Phi) \cap \nu(\Psi); \]
\[ - \nu(\neg \Phi) = \nu(\Phi); \]
\[ - \nu(R_{x_1,\ldots,x_n}^{\nu_1,\ldots,\nu_n} \Phi) = (\nu(\Phi) \cup \{v_1,\ldots,v_n\}) \setminus \{x_i | v_i \notin \nu(\Phi), i \in \{1,\ldots,n\}\}; \]
\[ - \nu(\exists x \Phi) = \nu(\Phi) \setminus \{x\}; \]
\[ - \nu(\neg \neg \Phi) = \nu(\Phi). \]

We extend \( \nu \) on \( \Gamma \subseteq Fr \) defining \( \nu(\Gamma) = \bigcap_{\Phi \in \Gamma} \nu(\Phi) \).

Let \( R_x^{\nu_{x,y,z}}(\Phi) \) be \( R \)-formula such that \( \bar{u} \subseteq \nu(\Phi) \). Then \( R_x^\top(\Phi) \) is called \( Rs \)-form of \( R_x^{\nu_{x,y,z}}(\Phi) \).

\( R \)-formulas \( \Psi \) and \( \Xi \) are called \( Rs \)-Un-equivalent, if \( \Psi \) and \( \Xi \) have the same \( Rs \)-forms or these \( Rs \)-forms have the same \( Un \)-forms. Here \( Un \) is a parameter of equivalence relation and \( Rs \) is a sign of special equivalence.

If \( \Psi \) and \( \Xi \) are \( Rs \)-Un-equivalent, then \( -\Psi \) and \( -\Xi \) will be also called \( Rs \)-Un-equivalent.

**Lemma 2.** Let \( \Psi \) and \( \Xi \) be \( Rs \)-Un-equivalent, \( J \) be an interpretation. Then \( T(\Psi_J) \cap V^\mathbb{Un} A = T(\Xi_J) \cap V^\mathbb{Un} A \) and \( F(\Psi_J) \cap V^\mathbb{Un} A = F(\Xi_J) \cap V^\mathbb{Un} A \).

Informal treatment of this lemma is as follows: in the case of \( Rs \)-Un-equivalent formulas we have that \( \Psi_J(d) = \Xi_J(d) \) for any \( d \in V^A \) such that variables from \( Un \) are not assigned.

**2.4 Logical Consequence Relation under Conditions of Undefinedness**

Traditional consequence relations usually are binary relations. In our case, introduction of composition \( \sim \) requires more complicated ternary consequence relation because formulas, treated as undefined, should be explicitly taken into consideration. Here we introduce such consequence relation denoted \( |=_{IR\perp} \) between three sets of formulas. This relation will generalize binary irrefutability relation \( |=_{IR} \) [7].

Let \( \Sigma \subseteq Fr \) and \( J \) be an interpretation. We denote:
\[ \bigcap_{\Phi \in \Sigma} T(\Phi_J) \text{ as } T^\land(\Sigma_J), \quad \bigcap_{\Phi \in \Sigma} F(\Phi_J) \text{ as } F^\land(\Sigma_J), \quad \bigcap_{\Phi \in \Sigma} \perp(\Phi_J) \text{ as } \perp^\land(\Sigma_J). \]

Here \( \Sigma_J \) is the set \{ \( \Phi_J | \Phi \in \Sigma \) \}. Set \( \Sigma \) can be empty. In this case
\[ T^\land(\Sigma) = T^\land(\emptyset) = F^\land(\Sigma) = F^\land(\emptyset) = \perp^\land(\Sigma) = \perp^\land(\emptyset) = V^A. \]
Let \( \Gamma, U, \Delta \subseteq Fr \). Then \( \Delta \) is called an irrefutable consequence of \( \Gamma \) under undefinedness conditions \( U \) in interpretation \( J \) (denoted \( U / \Gamma \models_{IR} \Delta \)) if

\[
T \cap (\Gamma J) \cap \bot (U J) \cap F \cap (\Delta J) = \emptyset.
\]

\( \Delta \) is logical irrefutable consequence of \( \Gamma \) under undefinedness conditions \( U \) (denoted \( U / \Gamma \models_{IR} \Delta \)), if \( U / \Gamma \models_{IR} \Delta \) for any interpretation \( J \).

In particular, for \( U = \emptyset \), we get traditional logical irrefutability \( \Gamma \models_{IR} \Delta \).

Relation \( \models_{IR} \) is monotone in the following sense:

M) Let \( \Gamma \subseteq \Lambda, U \subseteq W, \) and \( \Delta \subseteq \Sigma \); then \( U / \Gamma \models_{IR} \Delta \Rightarrow W / \Lambda \models_{IR} \Sigma \).

For \( \models_{IR} \) the following properties of formula decomposition hold.

**Theorem 1.** For any \( U, \Gamma, \Delta \subseteq Fr, \Phi, \Psi, \Theta \in Fr \):

- \( \neg_L \) \( U / \neg \Phi, \Gamma \models_{IR} \Delta \Leftrightarrow U / \Gamma \models_{IR} \Delta, \Phi \);
- \( \neg_R \) \( U / \Gamma \models_{IR} \Delta, \neg \Phi \Leftrightarrow U / \Phi, \Gamma \models_{IR} \Delta \);
- \( \lor_L \) \( U / \Phi \lor \Psi, \Gamma \models_{IR} \Delta \Leftrightarrow U / \Phi, \Gamma \models_{IR} \Delta \) and \( U / \Psi, \Gamma \models_{IR} \Delta \);
- \( \lor_R \) \( U / \Gamma \models_{IR} \Delta, \Phi \lor \Psi \Leftrightarrow U / \Gamma \models_{IR} \Delta, \Phi, \Psi \);
- \( \neg_U \) \( U, \neg \Theta / \Gamma \models_{IR} \Delta \Leftrightarrow U, \Theta / \Gamma \models_{IR} \Delta \);
- \( \lor_U \) \( U, \Phi \lor \Theta / \Gamma \models_{IR} \Delta \Leftrightarrow U, \Phi, \Theta / \Gamma \models_{IR} \Delta \) and \( U, \Phi / \Gamma \models_{IR} \Theta, \Delta \) and \( U, \Theta / \Gamma \models_{IR} \Phi, \Delta \);
- \( \neg_U \) \( U, \neg \Phi / \Gamma \models_{IR} \Delta \Leftrightarrow U / \Phi, \Gamma \models_{IR} \Delta \) and \( U / \Gamma \models_{IR} \Delta, \Phi \);
- \( \lor_U \) \( U / \Phi, \Gamma \models_{IR} \Delta \Leftrightarrow U, \Phi / \Gamma \models_{IR} \Delta \);
- \( \sim_{EU} \) \( U, \sim E \phi / \Gamma \models_{IR} \Delta \Leftrightarrow U / \Gamma \models_{IR} \Delta \).

Let us consider properties of relation \( \models_{IR}^U \) for renomination composition. Each of the properties \( R \lor, R \rightarrow, RR, R \exists, R, RI, RU \) of Lemma 1 [10] induces three corresponding properties for \( \models_{IR}^U \), depending on the position of a formula (in the left side of \( \models_{IR}^U \), in the right side of \( \models_{IR}^U \), in the undefinedness conditions of \( \models_{IR}^U \)). Such properties are formulated in a similar way.

Let us consider two examples.

The distributivity property \( R \sim \) of renomination with respect to \( R^\sim \phi (P) = \sim R^\phi (P) \) induces the following properties \( R \sim_L, R \sim_R, R \sim_U \):

- \( R \sim_L \) \( U / R^\phi (\Phi), \Gamma \models_{IR} \Delta \Leftrightarrow U / \sim R^\phi (\Phi), \Gamma \models_{IR} \Delta \);
Completeness of the Logic of Partial Quasiary Predicates …

\[ R \sim_r \) \ U/\Gamma =_{IR} \Delta, R_x^\Phi (\neg \Phi) \iff U/\Gamma =_{IR} \Delta, \neg R_x^\Phi (\Phi); \]

\[ R \sim_u \) \ U, R_x^\Phi (\neg \Phi)/\Gamma =_{IR} \Delta \iff U, \neg R_x^\Phi (\Phi)/\Gamma =_{IR} \Delta. \]

For variable assignment predicate \( Ez \) we have the following properties:

\( \text{RE}_L \) \ U / R_x^\Phi (Ez), \Gamma =_{IR} \Delta \iff U / Ez, \Gamma =_{IR} \Delta, \text{ if } z \notin \bar{v}; \)

\( \text{RE}_R \) \ U / \Gamma =_{IR} \Delta, R_x^\Phi (Ez) \iff U / \Gamma =_{IR} \Delta, Ez, \text{ if } z \notin \bar{v}; \)

\( \text{RE}_U \) \ R_x^\Phi (Ez), U / \Gamma =_{IR} \Delta, \iff Ez, U / \Gamma =_{IR} \Delta, \text{ if } z \notin \bar{v}; \)

\( \text{RE}_{RL} \) \ U / R_{x,y}^\Phi (Ez), \Gamma =_{IR} \Delta \iff U / \Gamma =_{IR} \Delta, Ey; \)

\( \text{RE}_{RU} \) \ R_{x,y}^\Phi (Ez), U / \Gamma =_{IR} \Delta, \iff Ey, U / \Gamma =_{IR} \Delta; \)

\( \text{Ed} \) \ U / \Gamma =_{IR} \Delta \iff U / Ey, \Gamma =_{IR} \Delta \text{ and } U / \Gamma =_{IR} \Delta, Ey; \)

\( \text{Ev} \) \ U / \Gamma =_{IR} \Delta \iff U / Ez, \Gamma =_{IR} \Delta \text{ if } z \in fu(U, \Gamma, \Delta). \)

The following properties will induce the quantifier elimination rules:

\( \exists_L \) \ U / \exists x \Phi, \Gamma =_{IR} \Delta \iff U / R_x^\Phi (\Phi), Ez, \Gamma =_{IR} \Delta; \)

\( \exists_R \) \ U / R_x^\Phi (\exists x \Phi), \Gamma =_{IR} \Delta \iff U / R_{x,y}^\Phi (\Phi), Ez, \Gamma =_{IR} \Delta; \)

\( \exists_v \) \ U / \Gamma, Ey =_{IR} \exists x \Phi, \Delta \iff U / \Gamma, Ey =_{IR} \exists x \Phi, R_x^\Phi (\Phi), \Delta; \)

\( \exists_v \) \ U / \Gamma, Ey =_{IR} \exists x \Phi, R_x^\Phi (\Phi) \iff U / \Gamma, Ey =_{IR} \exists x \Phi, R_{x,y}^\Phi (\Phi); \)

\( \forall \) \ U, \exists x \Phi / \Gamma =_{IR} \Delta \iff U, \exists x \Phi, R_x^\Phi (\Phi) / Ez, \Gamma =_{IR} \Delta; \)

\( \forall \) \ U / R_x^\Phi (\exists x \Phi) / \Gamma =_{IR} \Delta \iff U, R_x^\Phi (\exists x \Phi), R_{x,y}^\Phi (\Phi) / Ez, \Gamma =_{IR} \Delta. \)

For \( \exists_L \) and \( \exists_U \) additional constraint is \( z \in fu(U, \Gamma, \Delta, \exists x \Phi) \); for \( \exists_R \) and \( \exists_R \) the constraint is \( z \in fu(U, \Gamma, \Delta, \exists x \Phi) \).

Properties \( \exists_L, \exists_R, \text{ and } \exists_U \) are special cases of \( \exists_R \), \( \exists_v \), \( \exists_R \) respectively.

Here \( z \in fu(U, \Gamma, \Delta, \exists x \Phi) \) and \( z \in fu(U, \Gamma, \Delta, R_x^\Phi (\exists x \Phi)) \) specify an unessential variable \( z \) that does not occur in formulas from \( U, \Gamma, \Delta, \) and in \( \exists x \Phi \) and in \( U, \Gamma, \Delta, \) and in \( R_x^\Phi (\exists x \Phi) \) respectively.

Theorems 2 and 3 define properties that guaranty validity of the consequence relation.
Theorem 2. Let $U, \Gamma, \Delta \subseteq Fr$, $\Phi$ and $\Psi$ be Rs-Un-equivalent formulas. Then the following properties hold:

- $C^{un}(U / \Phi, \Gamma |\equiv IR \perp \Delta, \Psi$, in particular, $U / \Phi, \Gamma |\equiv IR \perp \Delta, \Phi$;
- $C^{un,U_L}(U, \Phi / \Psi, \Gamma |\equiv IR \perp \Delta$, in particular, $U, \Phi / \Phi, \Gamma |\equiv IR \perp \Delta$;
- $C^{un,U_R}(U, \Phi / \Gamma |\equiv IR \perp \Delta, \Psi$, in particular, $U, \Phi / \Gamma |\equiv IR \perp \Delta$.

Theorem 3. Let $\Phi \in Fr$. Then the following properties hold:

- $C^{un}(U / \Phi |\equiv IR \perp \Delta, \neg \Phi$;
- $C^{un,U_E}(U, E\gamma / \Gamma |\equiv IR \perp \Delta$;
- $C^{un,\neg E}(U / \neg E\gamma, \Gamma |\equiv IR \perp \Delta$.

Let us admit that $C^{un,\neg E}$ is derivable from $C^{un,U_E}$ and $\neg L$.

3 Sequent Calculus for $L^{QEC}$

Usually, an inference relation $\vdash$ is defined by some axiomatic system (calculus). We present here a system that formalizes logical consequence relation between sets of formulas. Such systems are called sequent calculi.

We construct a sequent calculus $C^{QEC}$ for relation $\equiv IR \perp$.

The main objects of this calculus are sequents. Here we consider only the case with finite sequents. We treat them as finite sets of formulas signed by symbols $\vdash, \neg \vdash$, and $\perp$.

Formulas from $\Gamma$ are called T-formulas and are signed by $\vdash$, formulas from $\Delta$ are called F-formulas and are signed by $\neg \vdash$, and formulas from $U$ are called $\perp$-formulas and are signed by $\perp$.

Sequents are denoted $\vdash \Gamma \perp U \neg \perp \Delta$, in abbreviated form $\Sigma$.

The derivation in a sequent calculus has the form of a tree whose vertices are sequents.

The rules of sequent calculus are called sequent forms. They are syntactical analogs of the semantic properties of the logical consequence relation. Details of the definition of derivation tree can be found in [10].

Closed sequents are axioms of the sequent calculus.

A closed sequent is specified in such a way that the following condition should hold:

if sequent $\vdash \Gamma \perp U \neg \perp \Delta$ is closed then $U / \Gamma |\equiv IR \perp \Delta$.
A sequent calculus is defined by basic sequent forms and closure conditions of sequents.

The following conditions are induced by the properties $C_{\text{un}}$, $C_{\text{un}UL}$, $C_{\text{un}UR}$, $C_{\text{un}E}$, and $C_{\text{un}UE}$ respectively:

C) there are $Rs\text{-}Un$-equivalent $\Phi$ and $\Psi$ such that $\Phi \in \Gamma$ and $\Psi \in \Delta$;
C$_{\text{UL}}$) there are $Rs\text{-}Un$-equivalent $\Phi$ and $\Psi$ such that $\Phi \in U$ and $\Psi \in \Gamma$;
C$_{\text{UR}}$) there are $Rs\text{-}Un$-equivalent $\Phi$ and $\Psi$ such that $\Phi \in U$ and $\Psi \in \Delta$;
C) there is $\Phi$ such that $\sim \Phi \in \Delta$;
C$_{\text{UE}}$) there is $Ey$ such that $Ey \in U$.

For $C_{\text{QEC}}$ we take the following closure conditions:

sequent $|\Gamma \bot U \bot \Delta$ is closed if $C \lor C_{\text{UL}} \lor C_{\text{UR}} \lor C_{\text{E}} \lor C_{\text{UE}}$ holds.

Theorem 4. If sequent $|\Gamma \bot U \bot \Delta$ is closed then $U / \Gamma \models_{IR} \bot \Delta$.

Now we define the following groups of basic forms: decomposition forms, renomination forms, simplification forms, quantifier elimination forms, and variable assignment forms.

The sequent forms of decomposition for $\lor$, $\neg$, $\sim$ are induced by the corresponding properties of decomposition of formulas, in particular, basic sequent forms of $C_{\text{QEC}}$ calculus are induced by the formula decomposition properties $\neg_{L}$, $\neg_{R}$, $\lor_{L}$, $\lor_{R}$, $\neg_{U}$, $\lor_{U}$, $\sim_{U}$, $\sim_{L}$:

\[
\begin{align*}
\Gamma \vdash \Phi, \Sigma & \quad \Gamma \vdash \neg \Phi, \Sigma & \quad \Gamma \vdash \bot \Phi, \Sigma \\
\neg \Gamma \vdash \neg \Phi, \Sigma & \quad \neg \Gamma \vdash \neg \Phi, \Sigma & \quad \neg \Gamma \vdash \neg \Phi, \Sigma \\
\sim \Gamma \vdash \Phi, \Sigma & \quad \sim \Gamma \vdash \Phi, \Sigma & \quad \sim \Gamma \vdash \Phi, \Sigma \\
\sim \Gamma \vdash \sim \Phi, \Sigma & \quad \sim \Gamma \vdash \sim \Phi, \Sigma & \quad \sim \Gamma \vdash \sim \Phi, \Sigma \\
\sim \Gamma \vdash \sim \Phi, \Sigma & \quad \sim \Gamma \vdash \sim \Phi, \Sigma & \quad \sim \Gamma \vdash \sim \Phi, \Sigma \\
\sim \Gamma \vdash \sim E, \Sigma & \quad \sim \Gamma \vdash \sim E, \Sigma & \quad \sim \Gamma \vdash \sim E, \Sigma
\end{align*}
\]
The labels of the sequent forms are consistent with the labels of the properties of the decomposition of the formulas. Introduction of undefinedness formulas additionally leads to new sequent forms with three premises (rule $\bot \lor$).

For composition of renomination we use the following forms of equivalent transformations:

$$
\begin{align*}
|{-} R \lor & \frac{-R^\varphi_x(\Phi) \lor R^\varphi_y(\Psi), \Sigma}{-R^\varphi_x(\Phi \lor \Psi), \Sigma}; & |{-} R \lor \frac{-R^\varphi_x(\Phi), \Sigma}{-R^\varphi_x(\Phi \lor \Psi), \Sigma}; \\
& \frac{-R^\varphi_x(\Phi), \Sigma}{-R^\varphi_x(\Phi \lor \Psi), \Sigma}; & \frac{-R^\varphi_x(\Phi), \Sigma}{-R^\varphi_x(\Phi \lor \Psi), \Sigma}; & \frac{-R^\varphi_x(\Phi), \Sigma}{-R^\varphi_x(\Phi \lor \Psi), \Sigma}; & \frac{-R^\varphi_x(\Phi), \Sigma}{-R^\varphi_x(\Phi \lor \Psi), \Sigma}; & \frac{-R^\varphi_x(\Phi), \Sigma}{-R^\varphi_x(\Phi \lor \Psi), \Sigma}; & \frac{-R^\varphi_x(\Phi), \Sigma}{-R^\varphi_x(\Phi \lor \Psi), \Sigma}; & \frac{-R^\varphi_x(\Phi), \Sigma}{-R^\varphi_x(\Phi \lor \Psi), \Sigma}; \\
\end{align*}
$$

Forms of simplification:

$$
\begin{align*}
|{-} R & \frac{-\Phi, \Sigma}{-R(\Phi), \Sigma}; & |{-} R & \frac{-\Phi, \Sigma}{-R(\Phi), \Sigma}; & \frac{-\Phi, \Sigma}{-R(\Phi), \Sigma}; & \frac{-\Phi, \Sigma}{-R(\Phi), \Sigma}; & \frac{-\Phi, \Sigma}{-R(\Phi), \Sigma}; & \frac{-\Phi, \Sigma}{-R(\Phi), \Sigma}; \\
\end{align*}
$$
Completeness of the Logic of Partial Quasary Predicates

\[ \frac{\neg Ey, \Sigma}{\neg R^{\tau,y}(Ez), \Sigma} \]
\[ \frac{\neg Ey, \Sigma}{\neg R^{\tau,z}(Ez), \Sigma} \]
\[ \frac{\neg Ey, \Sigma}{\neg R^{\tau,z}(Ey), \Sigma} \]

For RU-forms there is additional constraint \( y \in \nu(\Phi) \); for RE-forms there is additional constraint \( z \not\in \{\nu\} \).

Forms of quantifier elimination:

\[ \frac{\exists \exists x \Phi, \neg Ey, \Sigma}{\exists x \Phi, \neg Ez, \Sigma} \]
\[ \frac{\exists \exists x \Phi, \neg Ey, \Sigma}{\exists x \Phi, \neg Ez, \Sigma} \]
\[ \frac{\exists \exists x \Phi, \neg Ez, \Sigma}{\exists x \Phi, \neg Ez, \Sigma} \]
\[ \frac{\exists \exists x \Phi, \neg Ez, \Sigma}{\exists x \Phi, \neg Ez, \Sigma} \]

For \( \exists \) and \( \exists \) the constraint is \( z \in fu(\Sigma, \exists x \Phi) \); for \( \exists R \) and \( \exists R \) the constraint is \( z \in fu(\Sigma, R^{\tau}(\exists x \Phi)) \); for \( \exists Rv \) and \( \exists v \) the constraint is \( \exists Ey \not\in \Sigma \).

Forms for variable assignment predicates:

\[ \frac{Ed \quad \neg Ey, \Sigma}{\neg Ey, \Sigma} \]
\[ \frac{Ev \quad \neg Ez, \Sigma}{\neg Ez, \Sigma} \]

For basic rules of \( C^{QEC} \) we have the following main properties [11].

**Theorem 5.** Let \( k \in \{1, 2, 3\} \) and \( \frac{\Gamma_1 \vdash U_1 \Delta_1 \ldots \Gamma_k \vdash U_k \Delta_k}{\Gamma \vdash \Gamma \Delta} \) be basic sequent form. Then

\[ \frac{U/\Gamma \vdash \Gamma \Delta \equiv U_1/\Gamma_1 \vdash \Gamma_1 \Delta_1 \\ldots \U_k/\Gamma_k \vdash \Gamma_k \Delta_k}{U/\Gamma \vdash \Gamma \Delta} \]

4 Soundness and Completeness of \( C^{QC} \)

First, we prove soundness of \( C^{QEC} \), then its completeness.

**Theorem 6 (soundness).** Let sequent \( \frac{\Gamma \vdash U \Delta}{\Gamma \vdash \Gamma \Delta} \) be derivable in \( C^{QEC} \). Then \( U/\Gamma \vdash \Gamma \Delta \).
Indeed, if $\vdash \Gamma \cup U \vdash \Delta$ is derivable then a finite closed tree was constructed. Therefore, for any leaf of this tree its sequent $\vdash \Lambda \cup W \vdash K$ is closed. Thus, by Theorem 5, $W / \Lambda \models_{IR} K$ holds. So, for the root of the tree (sequent $\vdash \Gamma \cup U \vdash \Delta$) we have that $U / \Gamma \models_{IR} \Delta$ holds.

The completeness is proved on the basis of theorems of the existence of a counter-model for the set of formulas of a non-closed path in the sequent tree. Such set are called model sets (Hintikka’s sets [14]).

Definition of a model set $H$ for $L^{QEC}$ is rather lengthy, therefore we give as an example only decomposition conditions for disjunction, negation, and complement compositions:

- $H \lor L$) If $\vdash \Phi \lor \Psi \in H$, then $\vdash \Phi \in H$ or $\vdash \Psi \in H$;
- $H \lor R$) If $\vdash \Phi \lor \Psi \in H$, then $\vdash \Phi \in H$ and $\vdash \Psi \in H$;
- $H \lor U$) If $\vdash \Phi \lor \Psi \in H$, then $\vdash \Phi \in H$ and $\vdash \Psi \in H$ or $\vdash \Phi \in H$ and $\vdash \Psi \in H$;
- $H \neg L$) If $\vdash \neg \Phi \in H$, then $\vdash \Phi \in H$;
- $H \neg R$) If $\vdash \neg \Phi \in H$, then $\vdash \Phi \in H$;
- $H \neg U$) If $\vdash \neg \Phi \in H$, then $\vdash \Phi \in H$;
- $H \neg \bot$) If $\vdash \neg \bot \in H$, then $\vdash \bot \in H$;
- $H \neg \bot$) If $\vdash \neg \bot \in H$, then $\vdash \bot \in H$ or $\vdash \bot \in H$.

More details on the definition of a model set can be found in [10, 11].

**Theorem 7.** Let $H$ be a model set for $L^{QEC}$. Then $H$ is satisfiable.

Proof goes similar to the proof of Lemma 2 [10] but additionally undefinedness conditions should be taken into account.

**Theorem 8.** Let $\varphi$ be unclosed path in a sequent tree for $\vdash \Gamma \cup U \cup \Delta$ and $H$ be the set of all formulas in $\varphi$. Then $H$ is a model set.

**Theorem 9 (completeness).** Let $U / \Gamma \models_{IR} \Delta$ hold. Then sequent $\vdash \Gamma \cup U \cup \Delta$ is derivable in $C^{QEC}$.

Indeed, let $U / \Gamma \models_{IR} \Delta$ and $\vdash \Gamma \cup U \vdash \Delta$ be not derivable. Then a sequent tree for $\vdash \Gamma \cup U \vdash \Delta$ is not closed. Thus, an unclosed path $\varphi$ exists in this tree. Let $H$ be the set of all formulas of this path. By Theorem 8, $H$ is a model.
set and, by theorem 7, $H$ is satisfiable. This means that a counter-model for $\vdash \Gamma \cup \Delta$ was constructed that contradicts to $U/\Gamma \models_{IR} \Delta$.

5 Conclusion

Extensive usage of logic in Computer Science leads to new logics that more adequately represent applied domains. Logic of partial quasiary predicates is one of such logics oriented on proving properties of programs. In our previous papers we studied a logic of renominative (quantifier-free) level. In this paper we have generalized the obtained results for the first-order logic of partial quasiary predicates extended with the complement composition. We have studied properties of this logic, constructed a sequent calculus for it and proved its soundness and completeness.

Obtained results can be useful for software verification; first steps were proposed in [15].

References


Reliability of Erasure Coding Schemes for Distributed Storage Systems

Andrei Bogdan Baran, Adrian Iftene, Hugues Mercier

Abstract

One of the most important aspects of a distributed storage system is the ability to ensure data reliability in case of a failure and the ability to recover it. One way is using replication, but a more efficient approach is using erasure codes, fragments (parity blocks) spread across multiple nodes in the storage system. In case of a node failure, it can be repaired using the fragments distributed in the network. Erasure codes can offer the reliability of data replication at a much smaller storage overhead. The objective of this paper is testing the reliability of a class of erasure codes by setting up a simulator that can be used to analyze different patterns and configurations. We will introduce a set of erasure codes and we will show how the reliability of the overall system can be improved, by having less redundancy and less storage overhead.

Keywords: Erasure correcting codes, distributed storage systems, reliability, simulator, large-scale graph

1 Introduction

Evolution of digital data has seen an exponential growth in the past decade and the need for extra storage for both structured and unstructured data has increased day by day. According to the statistics published by IDC Research, the compound annual growth rate will be 42% by the end of 2020[1] Storing high amounts of data involves
Andrei Bogdan Baran et al.

high costs and reliability issues that need to be addressed by proper storage design.

Reliability can be assured by adding redundancy in the system. The simplest method is replicating data using $n$-copies, a method adopted in many practical storage systems. A more efficient way is to store data using erasure codes, an approach that can offer the reliability of data replication at a much smaller storage overhead. The general idea is to break the original data into smaller blocks (fragments), expand and encode with redundant data (parity blocks) and spread them across the storage network.

2 State of the Art

In the IT history, many schemes have been created in order to protect data against storage failures, especially disk drive hardware. In order to assure the system reliability, two main approaches have been used: duplication of files, objects (backup, archiving, synchronization) or parity-based schemes at disk level such as RAID\[1\] or at object level using erasure codes, such as Reed-Solomon\[2\].

Reed-Solomon (RS) coding is the canonical erasure code and is one of the most widely used. RS code is part of the well-known family of Maximum Distance Separable (MDS) codes\[2\], which offers the maximum reliability for a given storage overhead. It is the only MDS coding technique for an arbitrary $n$ and $m$, that always tolerates $m$ erasures.

Recently, a lot of attention has been given to erasure codes that provide a better storage alternative compared to the traditional approach of data replication. Big distributed storage players such as Microsoft Azure or Apache Hadoop put their efforts in creating new storage schemes based on erasure codes, with the goal of achieving fault tolerance in a resource-efficient manner.

In Windows Azure Storage (WAS), the infrequently accessed and long-lived data (cool nodes) are stored using Local Reconstructions\[3\].

\[https://www.caringo.com/downloads/whitepapers/caringo_whitepaper_replication_and_erasure_coding.pdf\]
Codes (LRC)\textsuperscript{5} that can reduce the storage overhead to up to 1.33 times while maintaining higher reliability compared to a system with 3 replicas. Another property of the LRC codes is a slightly reduced number of coding fragments needed to reconstruct the original data.

\textbf{Hadoop Adaptively-Coded Distributed File System (HACFS)} is an erasure code scheme that dynamically adapts to work-load changes and uses two different erasure codes: fast code (for frequently accessed data) and compact code (for less frequently accessed data). The main goals for HACFS include fast degraded reads, low reconstruction time and low storage and network overhead. The HACFS extension\textsuperscript{6} brings some improvements by encoding the files based on some metrics such as read counts.

Another approach with focus on durability was proposed in 2013\textsuperscript{7}, called \textbf{Helical Entanglement Codes (HEC)}. Here the files are entangled with both previous-stored and forthcoming data, in other words, the old data is encoded with new data. The structure of these codes is based on a helical lattice that helps to propagate redundant encoded information using a chain of data and block parities (strands). The documents are entangled using simple encoding mechanism based on exclusive-or (XOR) operations. By creating intricate dependencies between the old and future blocks, the model can provide strong durability properties.

\section{Our Contributions}

In this paper, we propose a set of erasure codes, that can offer the reliability of data replication at a much smaller storage overhead. The idea is similar to HEC, but the novelty is represented by a practical implementation where we can demonstrate how the overall system can be made more robust by introducing different parameters for the entanglement process while keeping the performance at an acceptable level.

We implemented a simulator, that for a given set of parameters, can map a graph of parity blocks for small systems and it can be used to analyze different erasure patterns. Moreover, the simulator is able
to find what the minimum number of blocks from the entangled chain of parity blocks that, if removed, the data is permanently lost.

3.1 Storage structure

Let \( D \triangleq \{d_0, d_1, d_2, \ldots \} \) be a set of data objects of unit size, and let \( B \triangleq \{b_0, b_1, b_2, \ldots \} \) be a set of parity blocks of unit size. We want to store this set \( D \) of data objects in a distributed storage system using \( p \) parity blocks per object to improve storage reliability.

The \( p \) parity blocks of each data object are generated based on a series of positive parameters \( \alpha^q_p \), where \( q > 0 \) as follows:

\[
\alpha^0 = \{\alpha^1_1, \alpha^1_2, \ldots, \alpha^1_p\} \\
\alpha^1 = \{\alpha^2_1, \alpha^2_2, \ldots, \alpha^2_p\} \\
\vdots \\
\alpha^{q-1} = \{\alpha^q_1, \alpha^q_2, \ldots, \alpha^q_p\}
\] (1)

For document \( d_k \), the \( \alpha^k \mod q \) set of positive parameters are used to generated the parity blocks. The equations are constructed as follow:

\[
b_{pk} = d_k \oplus b_{pk-\alpha^k_1 \mod q} \\
b_{pk+1} = d_k \oplus b_{pk-\alpha^k_2 \mod q} \\
\vdots \\
b_{pk+p-1} = d_k \oplus b_{pk-\alpha^k_p \mod q}
\] (2)

To bootstrap the process, we use a bootstrap block \( b_{-1} \triangleq 0 \) such that \( b_{pk-\alpha^k_p \mod q} = b_{-1} \) if and only if \( pk - \alpha^k_p \mod q < 0 \). It’s worth noticing that for large values of \( \alpha^k_p \mod q \), the bootstrap block will be used in the generation of the other blocks, making the storage system less robust. Proper values need to be chosen for the \( \alpha \) parameters sets, in order to have a robust storage model. This will be a feature of the simulator described in section \ref{simulator}.
To calculate block parities we use simple XOR arithmetic operations. It’s easy to observe that encoding and decoding involves a solution of linear equations.

A graph representation of the storage model and how the parity blocks are linking with other parity blocks can be seen in Figure 1.

**Example.** For $p = 3$, $q = 2$ and $\alpha^0 = \{1, 3\}$, $\alpha^1 = \{4, 5\}$.

The parity equations for $d_7$ are $b_{21} = d_7 \oplus b_{18}$, $b_{22} = d_7 \oplus b_{10}$, and $b_{23} = d_7 \oplus b_{-1}$.

We consider two variants: when both data objects and party blocks are stored (**Variant 1**) and when only parity blocks are stored (**Variant 2**).

### 3.2 Storage reliability

The objective of the work is to find the reliability of this storage system given parameters $p$ and $\alpha^1, \alpha^2, \ldots, \alpha^{q-1}$. More precisely, the goal is to find the smallest number of blocks that cause irrecoverable data loss, knowing that:

- a data object $d_k$ is **locally irrecoverable**, if there is no option to recover it from the surrounding blocks;

- a data object $d_k$ is **globally irrecoverable**, if there is no option
to recover it recursively, in other words if there is a set of locally irrecoverable data objects, including $d_k$, that do not propagate to recoverable data objects.

Since the system is symmetric, the focus can be on data object $d_0$ and the smallest number of blocks required to make $d_0$ irrecoverable. Since block $b_{-1}$ cannot be erased, $p$ parity blocks generated by $d_0$ must be erased for Variant 1. For Variant 2, besides parity blocks, the data object $d_0$ must be erased also. The deletion of these blocks propagates to other data objects, which must be tackle recursively.

As it can be seen, it is very important how we choose the positive $\alpha$ parameters. The storage system is more solid if the connections between the parity blocks are multiple, increasing the number of possible blocks that can be used for document object reconstruction. In case of a failure (erasure) of a data object, there are multiple options to recover it, so we need a good simulator to test different cases, that can observe the smallest deletion pattern of parity blocks and suggest the best configuration that we can use for a storage system. In the next section, we will describe how the simulator works and how it was implemented.

### 3.3 Simulator

*ReliabilitySimulator*, a C++ console program that models the storage system for a given configuration and support different functionalities such as:

- finding the minimum number of blocks (or blocks and data objects) that need to be erased in order to make a data object globally irrecoverable;

- suggesting *alpha* positive parameters set that will give more robustness to the storage system

- varying the starting data object node in order to observe the impact over the deletion pattern;
Reliability of Erasure Coding Schemes...

- finding the optimal configuration for keeping the number of erased blocks to the minimum, being able to generate sets of $\alpha$ positive parameters based on some bounds;

- or offering different statistics such as the number of enqueues or execution time.

Multiple strategies have been used for analyzing the deletion pattern, such as Depth First Search, Branch and Bound or generic Backtracking. However the selection of next node to be processed plays a key role in determining what is the next subproblem to be explored, in order to reduce the search space as early as possible.

The choices are represented by parity block that makes a data object $d_k$ locally irrecoverable (i.e. cannot recover some of the blocks surrounding $d_k$). The following choice selection strategies have been implemented:

- **InOrder** - the choices are processed in order, default strategy (brute force search);

- **MinCost** - the choices with the min cost are processed first;

- **FutureFirst** - goes in the future as much as possible, selecting the choice that contains as many parity blocks from the left side of the equations;

- **PastFirst** - stays in the past as much as possible, selecting the choice that contains as many parity blocks from the right side of the equations;

- **Random** - the choices are processed in a random order;

- **MinCostExt** - in case there are multiple choices with same min cost, take first the one closest to either end ($d_0$ or $d_n$);

- **MinCostApp** - in case there are multiple choices with same min cost, take first the one with the smallest number of appearances in the right side of the equations.
3.4 Evaluation

The search space grows significantly when $p$ is increasing and in this section, we will focus on relatively small systems. Different metrics have been measured such as execution time or the number of data objects enqueued.

Let’s consider a system with $p = 3$ (three parity blocks per data object), $q = 1$, $\alpha^0 = \{1, 3, 8\}$, starting node $d_0$ and $n = 100$ data objects where we try different choice selection strategies in order to make $d_0$ globally irrecoverable. Using the Backtracking Strategy, the results can be seen in Table 1.

<table>
<thead>
<tr>
<th>Choice Selection Strategy</th>
<th>Execution Time (sec)</th>
<th>Enqueueings</th>
</tr>
</thead>
<tbody>
<tr>
<td>InOrder</td>
<td>0.198969</td>
<td>1332</td>
</tr>
<tr>
<td>MinCost</td>
<td>0.099548</td>
<td>740</td>
</tr>
<tr>
<td>FutureFirst</td>
<td>2.15464</td>
<td>13324</td>
</tr>
<tr>
<td>PastFirst</td>
<td>0.100671</td>
<td>580</td>
</tr>
<tr>
<td>Random</td>
<td>0.57088</td>
<td>3652.2</td>
</tr>
<tr>
<td>MinCostExt</td>
<td>0.105428</td>
<td>740</td>
</tr>
</tbody>
</table>

Table 1: Choice selection performance

As it can be seen, the fastest choice selection strategies are the one based on the min cost functions. Depending on how we choose the starting node, one choice selection can perform better than the other, as it can be seen in the Table 2.

<table>
<thead>
<tr>
<th>Choice Selection Strategy</th>
<th>Execution Time (sec) $d_{15}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>MinCost</td>
<td>0.733778</td>
</tr>
<tr>
<td>MinCostExt</td>
<td>0.85478</td>
</tr>
</tbody>
</table>

Table 2: Choice selection based on min cost having as starting node $d_{15}$

When considering a block to be deleted, the cost is calculated based on how the erasure propagates on the data objects already processed or are about to be processed. If the $\alpha$ parameters are not chosen wisely,
the parity blocks edges emerge more slowly and it takes more time to compute the solution.

Next, we will present what speed impact will have on the storage model if we increase the number of data objects. The results can be seen in Table \[3\].

<table>
<thead>
<tr>
<th>Number of data objects</th>
<th>Execution Time (sec)</th>
<th>Enqueueulings</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>0.00124</td>
<td>13</td>
</tr>
<tr>
<td>50</td>
<td>0.008483</td>
<td>74</td>
</tr>
<tr>
<td>100</td>
<td>0.022646</td>
<td>148</td>
</tr>
<tr>
<td>500</td>
<td>0.355892</td>
<td>759</td>
</tr>
<tr>
<td>1000</td>
<td>1.33229</td>
<td>1498</td>
</tr>
</tbody>
</table>

Table 3: Execution time and number of enqueuings - system with \(p = 2, q = 1\)

Let’s consider another system with \(p = 3\) (three parity blocks per data object), 40 data objects, with two scenarios for \(q\) parameter, as it can be seen in Figure \[2\]. The starting node has been varied in order to see how data objects are protected and which data objects are more protected compared to the others. Erasure patterns for both the parity blocks and data objects are analysed. Most of the data objects have the same level of protection. However, the data objects from the end of the chart, are less protected. Once the system will age, those data will have the same level of protection as the ones from the beginning. Also when \(q = 1\), it must be erased 7 parity blocks when the data objects are not stored or 7 parity blocks and 5 data objects when the data objects are stored. By having two sets of \(\alpha\) parameters \((q = 2)\), the robustness of the storage model increased: now 9 parity blocks must be erased when the data objects are not stored or 9 parity blocks and 6 data objects when the data objects are stored.

The coding rate is not influenced if the \(q\) parameter is increased. Considering the examples above, for \(p = 2\), we have a coding rate of \(R = 1/2 = 0.50\) or \(R = 1/3 = 0.33\) (data objects are stored) and for \(p = 3\), we have a coding rate of \(R = 1/3 = 0.33\) or \(R = 1/4 = 0.25\).
Figure 2: Erasure pattern for $p = 3$

(data objects are stored). A storage overhead of 1.33x is similar to the one offered by Reed-Solomon codes, but this new set of codes can offer higher reliability.

4 Conclusion

Replication is not a viable solution for long-term data, like archives, due to the storage overhead of keeping 3 replicas. For the cold data (less frequently accessed), erasure coding can offer higher reliability and significantly lower storage cost than replication. There are multiple erasure coding schemes such as Reed-Solomons or Local Recoverable Codes that can make the system more tolerant to failures assuring a high durability and reliability of user’s data.

In this papers, we have seen how a new set of values for $\alpha$ can improve the storage system reliability making the system more robust. Creating a simulator, gave us the possibility to analyze different erasure patterns and to validate different $\alpha$ parameters.

Being a symmetrical storage model, the same level of protection can be assured for most of the data objects. The oldest data objects will have the strongest protection, while the most recently added data
object will be less protected. For the first category, erasure codes can be used, while for the last types of data objects, replication techniques to keep high the level of data reliability.

Acknowledgments. EBSIS - Horizon 2020 European project (grant agreement No. 692178) has supported part of the research for this paper.

References


213
Andrei Bogdan Baran, Adrian Iftene, Hugues Mercier

1Faculty of Computer Science, UAIC, Iasi, Romania
Email: andrei.baran@info.uaic.ro

2Faculty of Computer Science, UAIC, Iasi, Romania
Email: adiftene@info.uaic.ro

3Departament of Computer Science and Mathematics, Neuchatel University, Switzerland
Email: hugues.mercier@unine.ch
Mathematical Formalization of
Local Sensitivity Analysis in
Computational Models

Julian Dimitrov

Abstract

In this paper we study two joined notions in modeling – continuous and discrete presentation of numerical model. Equivalence criteria for these two parts of applied numerical model are presented. Some non-infinitesimal properties have been investigated. A condition for recoverability is formulated and an example is present.

Keywords: digital continuity, relative distance, Hadamard product, condition for recoverability.

1 Introduction

The applied numerical models in technics are characterized by large errors of the parameters. However, these models are presented by continuous dependencies. To dealing with this problem in practice applies sensitivity analysis as local measure of the effect of a given input on a given output [9]. The estimation is made by computing the sensitivity coefficients as partial derivatives at input parameters. With sensitivity analysis is estimation the uncertainty in numerical models [1].

Directly using the partial derivatives, as it is done in [1], [2], [6], [9] and others, causes some disadvantages. For eliminate the uncommon units in practice used scaled sensitivity coefficients by the nominal parameter value [1]. But this approach requires mathematical formalization, including estimation of uncertainty and offering a suitable method for multivariate analysis. Phenomenological approach to this problem expressed in the “Principal of Uncertainty” [4] is not an optimal solution.
But such a solution should not be burdened with unnecessary generalized properties.

In this paper we introduce and study two parallel notions in modeling – continuous and discrete presentation of numerical model and non-infinitesimal properties have been obtained. Equivalence criteria for these two parts of numerical model are presented. The degree of equivalence is estimated with a confidence probability – the method allows for quantitative evaluation.

2 Numerical Model – Notation and Properties

Similarly of the operating in the area of numerical algorithms, the applied numerical models (and so technical models) transforms from input parameters (data) to output. Systematic operation through the model and the stochastic nature of the data leads to the existence of the properties such as measurement uncertainty of the parameters and transformation uncertainty of the model.

Definition 2.1. We denote with $K_m^+ = \{ x = (x_1, \ldots, x_m) \in \mathbb{R}^m, x_j > 0 \}$ the positive cone in $\mathbb{R}^m$. Let $X \subset K_m^+$ open connected space, $f : X \rightarrow Y$ is a continuous mapping and $Y = f(X) \subset \mathbb{R}^n$. Then we will say that domain $X$ is space of input parameters and range $Y$ - space of output parameters. So $X$, $Y$ and $f$ defines a transformation model with input and output parameters.

We use the estimation of relative errors in space of input parameters $X$ for unify these parameters. We will introduce evaluations that are independent of the physical dimensions of parameters and within the limits of specified numerical values.

Definition 2.2. Let $Y \subset \mathbb{R}^n$ is space of output parameters, $y_0 = (y_1^0, \ldots, y_n^0) \in Y$ and $\varepsilon_i$ is a permissible absolute error to $i$-component of points in $Y$. Then we will say that $\varepsilon = \sqrt{\varepsilon_1^2 + \ldots + \varepsilon_n^2}$ is admissible absolute error of $Y$. We denote as
Mathematical Formalization of Local Sensitivity Analysis in Computational Models

\[ \Sigma_{y_0}^n = \{ y = (y_1, \ldots, y_n) \in Y : y_i = y_i^0 + 2 j \varepsilon_i, \, i = 1 \div n, \, j = 0, \pm 1, \pm 2, \ldots \} \]

the grid of model elements in \( Y \) with origin \( y_0 \).

Let \( x_0 = (x_1^0, \ldots, x_n^0), \vec{x} = (\vec{x}_1, \ldots, \vec{x}_m) \in X \), where \( x_0 \) is the origin value and \( \vec{x} \) is the measured value. Then \( \delta_i = \frac{\vec{x}_i - x_i^0}{x_i^0} \) is the permissible relative error to \( i \)-component of \( x_0 \). We assume that for every \( x \in X \) we have a permissible relative error to \( i \)-component and \( \delta_i < 1 \). Then we will say that \( \delta = \sqrt{\delta_1^2 + \ldots + \delta_m^2} \) is admissible relative error in \( X \). Let \( \delta_i < 1, \, i = 1 \div n \). In accordance we have the grid of model elements

\[ \Pi_{x_0}^m = \left\{ x = (x_1, \ldots, x_m) \in X : x_i = x_i^0 \cdot \left( \frac{1 + \delta_i}{1 - \delta_i} \right)^j, \, i = 1 \div m, \, j = 0, \pm 1, \pm 2, \ldots \right\} \]

with origin \( x_0 \).

**Remark 2.1.** Grid of model elements \( \Pi_{x_0}^m \subset X \subset K_m^+ \) is a presented with a multiplication group in \( X \) with component wise group operation. By analogy the \( \Sigma_{y_0}^n \subset Y \subset R^n \) is additive group. The notation of grids in \( X \) and \( Y \), introduced in accordance of relative and absolute uncertainty that characterized input and output data respectively.

With grids \( \Pi_{x_0}^m \) and \( \Sigma_{y_0}^n \) we introduce discretization in \( X \) and \( Y \) formally similar to discretization of Euclidian space in [3]. But our goal is to work in continuous spaces \( X \) and \( Y \) instead of the so-defined discrete spaces:

1. In this Section is present the basic concepts and properties of applied numerical model.
2. In Section 3 we introduce condition for recovery from discrete presentation as ability of numerical model to recognize the parameters in input space. This concept is compared with introduced from Chen [3] “Digital continuity”.

217
(3) In Section 4 was developed the issue for estimation with a confidence probability the degree of recoverability.

In accordance to relative error in $X$ we will use the Hadamard product for multiplication of vectors.

**Definition 2.3.** Let $x_0 = (x_1^0, \ldots, x_m^0)$, $x = (x_1, \ldots, x_m) \in \mathbb{R}^m$. Following [7] we will say that the Hadamard product of $x_0$ and $x$ is the vector $\hat{x} = x_0 \odot x = (x_1^0 \cdot x_1, \ldots, x_m^0 \cdot x_m) \in \mathbb{R}^m$.

**Proposition 2.1. (Properties of Hadamard product).**

1. Hadamard product is associative, distributive and commutative.
2. Vector $e = (1, \ldots, 1) \in \mathbb{R}^m$ is identity element - $e \odot x = x \odot e = x$.
3. For vector $x = (x_1, \ldots, x_m)$ with nonzero components ($x_i \neq 0$, $i = 1 \div m$) there is unique element $x^{-1} = (1/x_1, \ldots, 1/x_m)$ for which $x \odot x^{-1} = e$ and $(x^{-1})^{-1} = x$.
4. With the Hadamard product $K_m^+$ is multiplicative group with identity element $e$.
5. Let we denote with $\|x\|$ the Euclidean norm. For every $x_0, x \in \mathbb{R}^m$ the inequality

$$\|x_0 \odot x\| \leq \|x_0\| \cdot \|x\|$$

is hold.

Proof of Point 5. Let $x_0 = (x_1^0, \ldots, x_m^0)$ and $x = (x_1, \ldots, x_m)$. Then $\|x_0 \odot x\|^2 = (x_1^0)^2 \cdot (x_1)^2 + \ldots + (x_m^0)^2 \cdot (x_m)^2 \leq (x_1^0)^2 + \ldots + (x_m^0)^2 \cdot (x_1)^2 + \ldots + (x_m)^2 = \|x_0\|^2 \cdot \|x\|^2$. ⇒ Ineq. (1).

□

**Definition 2.4. (Relative Distance).** Let $y_0, y \in Y$. We are introducing notation for *Euclidean distance* $b(y_0, y) = \|y - y_0\|$ and if $x_0, x \in K_m^+$ - relative *pseudo-distance* (relative distance) $b^r(x_0, x) = \|(x - x_0) \odot x_0^{-1}\|$.
Proposition 2.2. (Some Properties of Relative Distance). Let \( x_0, x \in K_m^+ \) then

1. \( b^r(x_0, x) \geq 0 \), \( b^r(x_0, x) = 0 \iff x_0 = x \).
2. \( b^r(\lambda x_0, \lambda x) = b^r(x_0, x) \) for \( \lambda \neq 0 \).
3. \( b^r(x_0^{-1}, x^{-1}) = \| x^{-1} \circ (x_0^{-1})^{-1} - e \| = \| x_0 \circ x^{-1} - e \| = b^r(x, x_0) \).
4. \( \frac{\| x - x_0 \|}{\| x_0 \|} \leq b^r(x_0, x) \leq \| x - x_0 \| \cdot \| x_0^{-1} \| \)

Proof of Point 4. We have \( \| x - x_0 \| = \| (x - x_0) \circ x_0^{-1} \| \) and from Ineq. (1) we receive \( \| x - x_0 \| \leq \| (x - x_0) \circ x_0^{-1} \| \cdot \| x_0 \| = b^r(x_0, x) \cdot \| x_0 \| \). \( \Rightarrow \) left inequality in 4.

In the other hand \( b^r(x_0, x) = (x - x_0) \circ x_0^{-1} \| \leq \| x - x_0 \| \cdot \| x_0^{-1} \| \) is similarly derived from the Ineq. (1) \( \Rightarrow \) right side of 4.

Proposition 2.3. (Topology in \( K_m^+ \)). From Point 4. In Proposition 2.2 \( \Rightarrow \) the relative distance \( b^r \) induced the same topology as Euclidean distance.

A mapping \( f : X \rightarrow Y \) is continuous in \( X \) with the relative distance \( b^r \) exactly where \( f \) is continuous (\( X \) is with Euclidean distance).

Definition 2.5. (Discrete presentation). Let in space of input parameters \( X \subset K_m^+ \) we have a grid of model elements \( \Pi_{x_0}^m \) and \( f : X \rightarrow Y \subset R^n \) is continuous. Then we will say that restriction \( \overline{f} = f\big|_G \), were \( G = \Pi_{x_0}^m \) is discrete presentation of mapping \( f \).

Remark 2.2. In real technical tasks the components of model elements in \( X \) may have a different structure. It is enough to have reason to put the permissible relative error \( \delta_i \) to \( i \) - component and admissible relative error \( \delta \) so that they are the same for each \( x \in X \).
We are discussing special case of numerical model when we have a model with input and output technical parameters and mapping that has discrete presentation. The input parameters have positive values $X \subset K_m^+$. So we denote the space of input parameters as triplet $(X, b^r, \delta)$ and the space of output parameters - $(Y, b, \epsilon)$. This notation includes space, distance and value of admissible error.

**Definition 2.6. (Numerical model).** It is transformation model with input and output parameters, continuous mapping $f : (X, b^r, \delta) \rightarrow (Y, b, \epsilon)$ and its discrete presentation $\bar{f} = f|_G, G = \Pi_{x_0}^m$.

**Proposition 2.4.** The so-defined in Definition 2.6. numerical model has the following essential properties:

1. The definition is total – We can chose $\forall x_0 \in X$ as origin point of $\Pi_{x_0}^m$ and then $y_0 = f(x_0)$ is the origin point of $\Sigma_{y_0}^n$.
2. Is independent from a fixed choice of origin point in $\Pi_{x_0}^m$.
3. Is formulated in terms of continuity $f : X \rightarrow Y$ with additional no infinitesimal properties. The last statement is based on the fact that by parameters $\delta$ and $\epsilon$ are defined finite structures as their characteristic dimensions. In this sense, parameters $\delta$ and $\epsilon$ can be perceived as measurement units. Through these units are determined 'absolute' distances $d_y = b(y_0, y)/\epsilon$ and $d_x = b^r(x_0, x)/\delta$ in $Y$ and $X$ respectively.
4. Basic properties of absolute distances:
   (i) $d_y(y_0, y) = b(y_0, y)/\epsilon$ in $Y$ is independent from translation by steps $\epsilon_i$ in every $i$ - component and from $\epsilon$.
   (ii) $d_x(x_0, x) = b^r(x_0, x)/\delta$ in $X$ is independent from component - wise multiplication with multiplier $\frac{1 + \delta_i}{1 - \delta_i}$ in every $i$ - component and from $\delta$. 

220
3 Condition for Recoverability and Digital Continuity

With the set \( x_0 x = \{ x_0 + t(x - x_0) : x_0, x \in X, t \in [0,1] \} \) we denote a line segment in \( X \). The linear segment \( y_0 y \subset Y \) is defined in a similar way. We say that the points are recognizable on the subject of study in two cases:

(i) in \( (X, b', \delta) \) if \( x_0, x \in X, \overline{x_0 x} \subset X \) and \( b'(x_0, x) > \delta \);

(ii) in \( (Y, b, \varepsilon) \) if \( y_0, y \in Y, \overline{y_0 y} \subset Y \) and \( b(y_0, y) > \varepsilon \).

Let \( X \) and \( Y \) are convex spaces. We say that the couple points \( x_0, x \in X \) or \( y_0, y \in Y \) are identical when they are not recognizable on the subject of study.

**Definition 3.1. (Condition for Recovery).** Let \( f : (X, b', \delta) \rightarrow (Y, b, \varepsilon) \) is continuous mapping and \( \overline{f} \) is the restriction of \( f \) on \( \Pi_{x_0} \subset X \). We say that the mapping \( f \) can recover from discrete presentation \( \overline{f} \) if for every two points, recognizable of subject of study, \( y_0, y \in Y \) the respective couples \( x_0, x \in X \) for which \( f(x_0) = y_0 \) and \( f(x) = y \) are recognizable of subject of study.

**Remark 3.1.** Let \( f : (X, b', \delta) \rightarrow (Y, b, \varepsilon) \) is linear mapping \( f(x) = k \cdot x, x \in X, k > 0 \). We explore the ability to perform the condition of recover – Definition 3.1.

We have \( b(y_0, y) = k \|x - x_0\| \). Let \( x_0 = e \) - identity element. Then \( b'(x_0, x) = \|x - x_0\| \) and \( b(y_0, y) = kb(x_0, x) \). \( \Rightarrow \) A necessary and sufficient condition for the recovery is \( k \leq \varepsilon/\delta \). That is, we get the inequality \( b(y_0, y)/\varepsilon \leq b'(x_0, x)/\delta \), which is an inequality between the “absolute“ distances in \( Y \) and \( X \). With this inequality is defined the Lipschitz condition for \( f \) with Lipschitz constant \( \varepsilon/\delta \). We introduce this property as a condition for recoverability.
Definition 3.2. (Condition for Recoverability).

We assume that continuous mapping \( f : (X, b', \delta) \rightarrow (Y, b, \varepsilon) \) is recoverability if it transforms every two points \( x_0, x \in X \) with their absolute distance \( d_X(x_0, x) \) in \( y_0 = f(x_0), y = f(x), y_0, y \in Y \) and \( d_Y(y_0, y) \leq d_X(x_0, x) \).

In other words, if the inequality

\[
\frac{b(y_0, y)}{b'(x_0, x)} \leq \frac{\varepsilon}{\delta}
\]

is hold for every \( x_0, x \in X \) then \( f \) is recoverability.

Following [3] we present a generalized definition of “digital continuous function”.

Definition 3.3. (Digital continuous mapping).

The points \( x_0, x \in (X, b', \delta) \) we will call adjacent if \( \delta < b'(x_0, x) \leq 2\delta \) and the linear segment \( x_0 x \subset X \).

The mapping \( f : (X, b', \delta) \rightarrow (Y, b, \varepsilon) \) is said to be digital continuous if for every two adjacent points \( x_0, x \) we have \( b(f(x_0), f(x)) \leq 2\varepsilon \). In other words if for \( x_0, x \in X, x_0 x \subset X \) and \( \delta < b'(x_0, x) \leq 2\delta \Rightarrow b(f(x_0), f(x)) \leq 2\varepsilon \).

Theorem 3.1. Let \( f : (X, b', \delta) \rightarrow (Y, b, \varepsilon) \) is recoverability (fulfills the condition of recoverability – Definition 3.2.) then

(i) \( f \) can recover from discrete presentation \( \overline{f} \) and

(ii) \( f \) is digital continuous mapping.

Proof.

(i) Let \( b(y_0, y) > \varepsilon, y_0 \neq y \Rightarrow b'(x_0, x) = \left(\frac{b'(x_0, x)}{b(y_0, y)}\right) \cdot b(y_0, y) > (\delta/\varepsilon)\delta = \delta \).
Mathematical Formalization of Local Sensitivity Analysis in Computational Models

(ii) Let \( \delta < b'(x_0, x) \leq 2\delta \). \( \Rightarrow \) \( 2\delta \geq b'(x_0, x) \geq b(y_0, y) \cdot (\delta/\epsilon) \) \( \Rightarrow \) \( 2\epsilon \geq b(y_0, y) \)

\[ \square \]

We denote a ball in \((Y, b, \epsilon)\) as \( B_{\epsilon}(y_0) = \{ y : b(y_0, y) \leq \epsilon \} \). In a similar way \( B_{\delta}^r(x_0) = \{ x : b'(x_0, x) \leq \delta \} \) is a ball in \((X, b^r, \delta)\).

**Remark 3.2.** In the case where the radius \( \delta \) of the ball \( B_{\delta}^r(x_0) \) is small enough we can assume that \( f \) is linear in \( B_{\delta}^r(x_0) \). Then we say that \( f \) is linear with sufficient precision.

**Example 3.1.**

Let’s discuss the principal for calculation with uncertainties measurement data: “When we divide measurements we add percent uncertainties.” We have \( u = f(x_1, x_2) = x_1/x_2 \). From Taylor’s formula and triangle inequality we obtain for the relative deviation \( \Delta u/u \leq \delta_1 + \delta_2 + 2\delta_1\delta_2 + 2\delta_2^2 \), where \( \delta_1, \delta_2 \) are relative errors of \( x_1, x_2 \) respectively. If \( \delta_1 \leq 10^{-1}, \delta_2 \leq 10^{-1} \) and the relative error of input \( \epsilon/u \geq 10^{-1} \) then \( 2\delta_1\delta_2 + 2\delta_2^2 < 10^{-1} \) and we have loss of order because of unit round off. \( \Rightarrow \) \( \epsilon/u = \delta_1 + \delta_2 \). In this case function \( u = f(x_1, x_2) \) is linear with sufficient precision.

4 Dealing with Transformation Uncertainty in Numerical Model

Without significant restriction and the possibility of application we look in this chapter only the case when \( y = f(x) : X \rightarrow Y \subset R \) is a smoothly function. Follow [2] the inaccuracy of result (transformation uncertainty) is estimated by corresponding confidence interval with confidence probability.

4.1 Gradient in Semi-logarithmic Derivatives.

Let in \( X \subset R^n \) is given a smoothly function \( y = f(x) : X \rightarrow Y \subset R \). We make the calculation of the gradient in semi-logarithmic
derivative $D_{ln} = \left( \frac{\partial f(x_0)}{\partial \ln x}, \ldots, \frac{\partial f(x_0)}{\partial \ln x_m} \right)$, ensuring accordance with the analytical expressions of relative distance. We denote $D_{ln}$ as semi-logarithmic gradient.

The deviation of the function $f(x)$ is given by Taylor’s formula [5] as $\Delta f = Df \cdot \Delta x + O(\|\Delta x\|^2) = D_{ln}f \cdot \Delta_{ln}x + O(\|\Delta x\|^2)$, where $\Delta x = x - x_0$, $\Delta_{ln}x = (x - x_0) \circ x_0^{-1}$, $Df$ is the functional gradient and $D_{ln}f$ semi-logarithmic gradient.

In our notation $|\Delta f| = b(y_0, y)$ and $\|\Delta_{ln}x\| = b^r(x_0, x)$.

**Theorem 4.1.** Let $f : X \rightarrow Y \subset R$ is a smoothly function in $X \subset R^m$.

(i) For every $x_0, x \in X$, $x_0 \neq x$ there exists $\hat{x} \in x_0x, \hat{x} \neq x_0$, $b^r(x_0, \hat{x}) \leq b^r(x_0, x)$ and such that

$$|\Delta f| = |f(x) - f(x_0)| \leq \|D_{ln}f(\hat{x})\| \cdot b^r(x_0, x).$$  

(ii) Equality is reached in Ineq. (3) when $\hat{x} = x_0 \in X$, $\delta$ is small enough so that $f$ is linear with sufficient precision in $B^r_\delta(x_0)$, $x \in B^r_\delta(x_0)$ is selected so that $b^r(x_0, x) = \delta$ and $\Delta_{ln}x = \lambda D_{ln}f(x_0)$, for some $\lambda \in R, \lambda \neq 0$.

Proof.

(i) From Mean Value Teorem [5] we have that for every points $x_0, x \in X$, $x_0 \neq x$ there exists $\hat{x} \in x_0x, \hat{x} \neq x_0, \hat{x} \neq x$ such that $\Delta f = D_{ln}f(\hat{x}) \cdot \Delta_{ln}x$. Then from Cauchy – Schwarz inequality $\Rightarrow$ (3).

(ii) Let $\delta$ is selected so that $f$ is linear with sufficient precision in $B^r_\delta(x_0)$. Then $\Delta f = D_{ln}f(x_0) \cdot \Delta_{ln}x$ for every $x \in B^r_\delta(x_0)$. The Cauchy – Schwarz inequality turns into equation when both the vector to which it relates, are parallel. We can choose $x \in B^r_\delta(x_0)$ such that
Mathematical Formalization of Local Sensitivity Analysis in Computational Models

\[ \Delta_m x = \lambda D_m f(x_0) \] and \[ b'(x_0, x) = \delta. \] This can be done by choosing \[ x = x_0 + \lambda D_m f(x_0) \circ x_0, \] where \[ \lambda = \delta/\|D_m f(x_0)\|. \]

We check the two conditions:
1. \[ b'(x_0, x) = \|(x - x_0) \circ x_0^{-1}\| = \lambda \|D_m f(x_0)\| = \delta \]
2. \[ \Delta_m x = (x - x_0) \circ x_0^{-1} = \lambda D_m f(x_0) \]
\[ \Rightarrow \] the proposition in part (ii) is hold. \( \square \)

4.2 Criterion for Recoverability.

**Remark 4.1.** The following Proposition 4.1. – 4.2. are introduced not infinitesimal properties for linear with sufficient precision function \( f \) because error \( \epsilon \) is with fixed value.

Let \( f \) is linear function with sufficient precision in \( B_\delta^r(x_0) \). Than \( \Delta f = Df \cdot \Delta x = D_m f \cdot \Delta_m x \) and from Cauchy – Schwarz inequality

\[ |\Delta f| = \|D_m f \cdot \Delta_m x\| \leq \|D_m f\| \cdot \|\Delta_m x\|, \] where \( \|\Delta_m x\| = b'(x_0, x). \)

**Proposition 4.1.** Let \( f(x) : X \rightarrow Y \) is linear function with sufficient precision in \( B_\delta^r(x_0) \). From (2) and (3) we obtain condition for recoverability

\[ k \|D_m f(x_0)\| \leq \frac{\epsilon}{\delta}, \] \( (4) \)

where \( k = \cos \psi, \) \( \cos \psi \in [0, \pi/2] \) and \( \psi \) is the angle of segment in the ball of errors \( B_\delta^r(x_0) \).

4.3 Final Form of the Condition of Recoverability.

**Proposition 4.2.** The condition for recoverability (4) is determined from three independent parameters \( k = \cos \psi, \) \( \epsilon = \sqrt{\epsilon_1^2 + \ldots + \epsilon_m^2} \) and \( \|D_m f\| \).

Would think that in every interval of these parameters is determine a critical part with probability \( p \). The probability of falling at the same time in all three critical areas is \( p^3 \) and then \( 1 - \bar{p} = 1 - p^3 \) is confidential.
probability for recoverability. Final form of the condition of recoverability is satisfied by

\[ k(1 - p)^2 \|D_{ln}f\| \leq \frac{\varepsilon}{\delta}, \]  

(5)

where \( k = \cos \psi \), \( \cos \psi \in [0, \pi/2] \).

4.4 Application of the Condition of Recoverability.

By statistical point of view we assume that the error is uniformly distributed. Average statistics can replace as expected value of the deviation \( \Delta f \) with its main value \( \Delta f = Df \cdot \Delta x = D_{ln}f \cdot \Delta x \).

Let denote with \( \nu_\delta = (\delta_1, ..., \delta_m) \) the vector of errors in \( X \). Following [2] the standard deviation of function in point \( x_0 \in X \) is present with the formula \( \sigma f = \|D_{ln}f(x_0) \circ \nu_\delta\| \). Than the inequality

\[ \sigma f \leq \varepsilon \]  

(6)

is condition for function \( f \) to be linear with sufficient precision in \( B^r_\delta(x_0) \), where \( \delta = \|\nu_\delta\| \).

When \( \sigma f > \varepsilon \) the function \( f \) is considered in the corresponding subset \( B^r_{\delta'}(x_0) \subset B^r_\delta(x_0) \), \( \delta' < \delta \). Due to the use of relative distance and semi-logarithmic derivatives \( f \), \( \sigma f \) and \( \|D_{ln}f(x_0)\| \) have the same physical dimension. This makes it easier to choose an appropriate elevated value of \( \|D_{ln}f(x_0)\| \) from data for adjacent points in grid \( \Sigma^n_{y_0} \subset Y \) [3].

5 Example

We are looking at the technical task for determination of the height limit \( H \) on the bench slope from the non-working board [8]. Height \( H \) in meters is given by

\[ H = \frac{2c}{\gamma} \cdot \frac{\sin(\alpha \cos \varphi)}{\sin^2\left(\frac{\alpha - \varphi}{2}\right)}, \]  

(7)
where $\gamma$ - volume weight of rock $kN/m^3$; $\alpha$ - angle of slope; $c$ - cohesion of rock $kN/m^2$; $\varphi$ - angle of friction.

We use the following samples of data for the parameters of the model: $\alpha=45^\circ, c=24 kN/m^2, \gamma=18 kN/m^3, v=\tan \varphi=0.32$. Parameters are presented with their absolute and relative errors respectively $c=[20,24,26] \pm 1=[24] \pm 4.2\%$ $[kN/m^2]$ and $v=[0.29,0.32,0.35] \pm 0.015=\{0.32\} \pm 4.7\%$. We consider two groups of data presented in Table 1 and Table 2. We calculate the gradient: $\frac{\partial H}{\partial \ln c} = H$ and $\frac{\partial H}{\partial \ln v} = H \cdot Q$, where $Q = \frac{\nu}{1+\nu^2} \left[cotg \left(\frac{\alpha-\varphi}{2}\right)-\nu\right]$. Then $\|D_{\ln}H\| = H \sqrt{1+Q^2}$. The permissible error of input data is $\varepsilon = \frac{\sqrt{4.2^2 + 4.7^2}}{100} = 6.3\%$.

For function $f = H$, with two inaccurate arguments $c$ and $\nu$, we receive $k = \cos \frac{\pi \rho}{2}$, coefficient values $k(1-p)$ and confidence probability $1-p = 1-p^2$ (Table 2).

<table>
<thead>
<tr>
<th>№</th>
<th>c $kN/m^2$</th>
<th>$\nu = \tan \varphi$</th>
<th>Q</th>
<th>$H[m]$</th>
<th>$|D_{\ln}H|$</th>
<th>$|D_{\ln}H| \cdot \varepsilon[m]$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>20</td>
<td>0.27</td>
<td>0.876</td>
<td>22.85</td>
<td>35.98</td>
<td>2.26</td>
</tr>
<tr>
<td>2</td>
<td>24</td>
<td>0.32</td>
<td>1.103</td>
<td>32.56</td>
<td>54.48</td>
<td>3.43</td>
</tr>
</tbody>
</table>

Table 2. Two groups of data - continuation

<table>
<thead>
<tr>
<th>№</th>
<th>$\delta$</th>
<th>$k(1-p)^2$</th>
<th>$P$</th>
<th>1-$\bar{P}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>0.442</td>
<td>0.30</td>
<td>97 %</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>0.292</td>
<td>0.40</td>
<td>94 %</td>
</tr>
</tbody>
</table>

Assuming that for parameter H admissible error is 1 meter, then for the experimental data of group №1 condition of recovery is filled with confidence 97% and for group №2 – with confidence 94%.
6 Conclusion
In this paper we establish condition for recoverability for measurement the uncertainty. For optimal evaluation of the dependencies, we use relative distance in space of input parameters and calculations with semi-logarithmic derivative. The proposed method takes into account the required accuracy of the result and gradient of functional dependence. An example is proposed for valuation the analytical expressions of a geotechnical model.

References

Julian Dimitrov
University of Mining and Geology, Sofia
E-mail:juldim@abv.bg
Computer Science in Applications
Recognition of Heterogeneous Documents: Problems and Challenges

Alexandru Colesnicov, Svetlana Cojocaru, Ludmila Malahov

Abstract

The paper deals with the problems at the recognition of documents that contain heterogeneous formalized script-presentable content (mathematical and chemical formulas, music scores, etc.). Due to its great diversity, recognition of such content can’t be performed by universal software system. Equally, there isn’t any uniform script presentation of the recognition results. State of the art in this area is reviewed. The relevant achievements are systematized. Problems, further development directions and possible solutions are identified. Functionality and implementation of Web platform for recognition of heterogeneous documents is discussed.

Keywords: information technologies, heterogeneous documents, optical recognition of heterogeneous content.

1 Introduction

At the revitalization of the cultural heritage, a need for optical recognition of heterogeneous contents in documents is met, for example, when trying to reuse text published without computers. We encountered this problem when re-publishing the book ”Numbers and Ideals” by V.A. Andrunachievici [1] in the modern Romanian Latin script. The text was scanned, recognized, transliterated and read by the editor in two man-days, while scanning, recognition and transliteration took less than three hours. The set of formulas required a long and painstaking
work of a highly qualified specialist, namely, 2 man-months, volumes of the text and formulas being approximately equal.

An additional complicating factor is a large volume of heterogeneous content. An example of such a task is the re-publishing of encyclopedias, in which one can meet mathematical and chemical formulas, music scores, diagrams, etc. besides the usual text. Thus, the problem arises of developing tools to support the process of recognition of heterogeneous (multi-source) content.

Let’s note that during the existence of mankind documents had been mostly in plain text interspersed with drawings. Even mathematical calculations had been described in words, although the use of standard phrases in this case can be considered as the beginning of formalism. In the process of canonization of religious cults, and later as science and arts developed, there was a need for a compact and instructive recording of various content elements. Recording of music had been the earliest\textsuperscript{1}.

There is a vast variety of content elements that obey certain formal rules mentioned above, and more: chemical structural formulas; chess notation; electronic circuits, etc. We will use the term heterogeneous or script-presentable content. The main characteristics of such content: 1) initially it is not a natural language text; 2) there is a scripting language for its description; 3) a graphical presentation can be reproduced from the script.

The existing term STEM\textsuperscript{2} content is limited to the affected areas\textsuperscript{3}.

Script description of heterogeneous content appeared in the development of its editors. It is a text in a formal language; generation from it the visual or even multimedia presentation of a given content item is possible using dedicated software. Its advantages are: ease of editing; compactness; rich search capabilities; ease of description exchange and reuse if the script language is standardized.

\textsuperscript{1}The oldest interpreted record of a religious song (“Hurrian Hymn”) is dated back to 1400 B.C.

\textsuperscript{2}STEM: Science, Technology, Engineering, and Mathematics

\textsuperscript{3}https://sites.google.com/site/stemteaching/stem-content
An example of such a scripting language is \LaTeX, which is widely used in the publication of mathematical works and in many other areas. This language is constantly evolving and complemented, in particular in the aspect of the presentation of non-textual elements. For example, in mathematics, very complex commutative diagrams, for which the available \LaTeX tools were lacking, had to be drawn in a vector graphics editor. Drawing of one chart took 2-3 hours. After the appearance of more advanced tools in \LaTeX, the description of the same diagram requires no more than 15 minutes.

In this paper, we describe State of the art in modern OCR, briefly outline the most well-known scripting languages for various fields (music, chemistry, mathematics, etc.), and formulate the main problems and challenges arising in the recognition of heterogeneous texts.

## 2 State of the Art in OCR

Content recognition began with text recognition. Currently it’s an advanced technology, and we describe state of the art in this area as a point of reference.

On OCR software, let us take as an example commercial ABBYY Finereader\(^4\) (AFR), and open source free OCRopus\(^5\) and Tesseract\(^6\).

Currently two basic technologies used in text recognition are character patterns and neural networks. Development of neural networks evolved text recognition from recognition of separate characters to the upper levels, namely, to intellectual character recognition and, further, to intellectual words recognition (recognition of whole phrases and lines in a text at once).

OCRopus and Tesseract permit to switch among these technologies. The current version AFR 14 uses character patterns. Nevertheless, it has been stated\(^7\) that “ABBYY has been in the field of neural net-
work since its foundation 25 years ago”, and “have developed solutions for various tasks”.

AFR offers to its user a convenient graphical shell from which the entire document recognition cycle is available, starting with scanning. It provides complex image correction (for example, correction of trapezoid distortion), page segmentation with automatic detection of the segment type (text, table, or picture), analysis of the table structure, dehyphenation, manual editing after recognition, etc.

AFR 14 recognizes texts in 192 languages in any combination, of which vocabulary support is provided for 48. The user can define a new language specifying its alphabet, as well as download an additional user dictionary. In difficult cases, AFR can be trained, and the accumulated character patterns can be downloaded and uploaded.

OCRopus and Tesseract do not provide a single product with a full recognition cycle, as well as a graphical shell. These are packages of separate programs managed from the command line. Ready models exist only for a small number of languages and fonts; for others, it is necessary to perform training and model building from scratch.

There are other text recognition systems, but many of them are no longer supported. This is because the OCR system is complex and time consuming to develop. If the system does not meet the enthusiasm of users or does not stand up to competition, the investment of forces and finances in the development instantly loses its meaning. This happened, for example, with CuneiForm that in the beginning of 2000s successfully competed against AFR but was discontinued since 2008.

Some OCR systems that were updated during 2017—2019 are: Tesseract, CIB OCR, AFR, OCR.space, Ocropy, Infty Reader, OmniPage, ReadIris, TopOCR.
3 Scripts for Heterogeneous Content

3.1 Musical scores

Most music editors use their own file formats. De facto exchange standards between systems are MIDI and scripting MusicXML. For example, an open source free music editor MuseScore 3 provides MusicXML and MIDI data exchange, as well as export and import of some other formats. One of its own formats mscx is also scripting.

MIDI is primarily an industry standard communication protocol for electronic musical instruments. Information is transmitted in the form of binary messages.

MusicXML is an XML file with specific tags and attributes for music. For example, meter signature 3/4 is given in the form:

```
<time symbol="common">
  <beats>3</beats>
  <beat-type>4</beat-type>
</time>
```

3.2 Chemistry

One of important issues for chemistry is the unequivocal identification of a chemical for production and trade. The assortment of chemicals is huge (millions). Each of them can have dozens of trade names, and, for structurally complex chemicals, tens, hundreds, and even thousands of image variants.

International industry standards in chemistry adopted by IUPAC. These are: InChI, IUPAC nomenclature for inorganic chemistry; IUPAC nomenclature for organic chemistry.

InChI describes one molecule as a single text string, and the description of each molecule is unique. For example, the description of

---

14 Musical Instrument Digital Interface
15 https://www.midi.org/
16 https://www.musicxml.com/
17 International Union of Pure and Applied Chemistry
18 International Chemical Identifier
ethyl alcohol $\text{CH}_3\text{CH}_2\text{OH}$ has the form:

\text{InChI}=1S/C2H6O/c1-2-3/h3H,2H2,1H3

Since these strings can be quite long, a hash code (InChI key) of constant length is generated from the InChI string. InChI key for ethanol is LFQSCWFLJHTTHZ-UHFFFAOYSA-N. InChI keys are used to search the chemical databases. In most cases, the search is performed with the first 14 characters of 27. False identification (ambiguity) is extremely rare.

InChI is a script, but its writing is quite difficult, so it is better to generate it using the molecular editor.

IUPAC nomenclatures, like InChI, give an unambiguous description of molecules, but in a more convenient form for human perception. According to these nomenclatures, ordinary water $\text{H}_2\text{O}$ is described as \textit{dihydrogen monoxide} (inorganic), and ethyl alcohol as \textit{ethanol} (organic). These were the simplest cases. Caffeine will be \textit{3,7-dihydro-1,3,7-trimethyl-1H-purine-2,6-dione}.

There are other scripting languages that allow to write chemical formulas.

\LaTeX\ contains more than 50 packages for chemistry\footnote{https://ctan.org/topic/chemistry} both for formulas and structural diagrams. Simple chemical formulas can be written in pure \LaTeX\ without additional tools, for example \texttt{H$_2$O} (water, $\text{H}_2\text{O}$) or \texttt{CH$_3$CH$_2$OH} (ethanol, $\text{CH}_3\text{CH}_2\text{OH}$).

\textit{SMILES}\footnote{Simplified Molecular Input Line Entry System} allows to write not only molecular formulas, but also reaction equations. The notation is ambiguous, but there is a \textit{canonical} unambiguous version. The principles of writing are very simple; therefore, SMILES can be easily created manually and is human-perceivable. The system is focused more on organic chemistry, in inorganic chemistry its scripts are redundant and cumbersome. Canonical SMILES for ethanol is $\text{CCO}$.

\footnote{https://ctan.org/topic/chemistry} \footnote{http://www.daylight.com/dayhtml/doc/theory/theory.smiles.html}
The scripting notation MDL Molfiles\(^{22}\) is very original: its scripts contain coordinates of the atoms in the 3D structure of the molecule, in angstroms, and description of the bonds. Unambiguity is not provided.

### 3.3 Mathematics

Standard scripting languages for mathematics are \(\LaTeX\) and MathML\(^{23}\). There are also scripting languages for computer algebra software like commercial \(\text{Maple}\)\(^{24}\) and \(\text{Mathematica}\)\(^{25}\), or free \(\text{Maxima}\)\(^{26}\).

In mathematics, we distinguish, firstly, formulas and equations, and, secondly, commutative diagrams. The latter are usually described by a special sublanguage. For example, in \(\LaTeX\), vector graphics packages \(\text{XYpic}\) and \(\text{tikz}\) contain separate subpackages (main package options) for commutative diagrams. In particular difficult cases, the non-specialized vector graphics subsystem \(\text{MetaPost}\) with its own script language is used. In general, \(\LaTeX\) is a huge collection of scripts and scripting tools, including even \(\text{MetaFont}\) to describe fonts.

### 3.4 Other Scripting Languages for Heterogeneous Content

We should mention here languages used with CAD software.

In mechanical engineering and building, the most known CAD is \(\text{AutoCAD}\). This is a commercial system, and its language doesn’t even have open specifications as it is used interactively. However, there are a lot of CAD products including free open-source ones, which makes obvious the necessity to develop standards to exchange CAD projects in the electronic form. For example, it was estimated\(^{2}\) that as on 2004 the USA capital facilities industry had lost annually $15.8 billions due to

\(^{22}\)https://docs.chemaxon.com/display/docs/MDL+MOLfiles%2C+RGfiles%2C+SDfiles%2C+Rxnfiles%2C+RDfiles+formats
\(^{23}\)https://www.w3.org/Math/
\(^{24}\)https://www.maplesoft.com/products/Maple/
\(^{25}\)https://www.wolfram.com/mathematica/
\(^{26}\)http://maxima.sourceforge.net/
inadequate interoperability arising from “the highly fragmented nature of the industry, the industry’s continued paper based business practices, a lack of standardization, and inconsistent technology adoption among stakeholders”.

Several project exchange languages for CAD are standardized on international or industry level. All are scripting, although some of them have two variants: textual (scripting) and binary. Thanks to this, CAD not only provides a graphical representation of design objects, but also, for example, strength calculations use the same script as the source code. It is also possible to generate commands for CNC machines or 3D printers. This proves the necessity and usefulness of the standard script description as opposed to any other.

Examples of manufacturer-neutral scripting standards for CAD file exchange are IGES and STEP\textsuperscript{27}. IGES scripts consist of 80-character records to be punched on cards. Since the initiation of STEP in 1994, IGES stopped its development, its last version being dated by 1996. Nevertheless, old IGES files since 1980s can be reused today.

Let us take the concept of BIM\textsuperscript{28}. In this concept, the only information model of the building is provided by a single script. It is used at all stages, from design and construction to the operation of the building inclusive. A three-dimensional model of a building object is associated with an information database, in which additional attributes can be assigned to each element of the model. The building object is actually designed as a whole. Changing anyone of parameters entails an automatic change of the other parameters and objects associated with it, up to drawings, visualizations, specifications, financial calculations and calendar schedules.

\textsuperscript{27}ISO 10303, Standard for the Exchange of Product model data.
\textsuperscript{28}Building Information Modeling, or Building Information Model
4 Optical Recognition of Heterogeneous Content

The need for this arises when working with sources that are available only in graphical form. Modern standards and traditions of graphic representation of industrial, scientific and technical information evolved over the 20th century in the absence of computer support. This implies that most paper sources of that and, possibly, earlier periods, may require such recognition.

For **musical scores**, the problem of recognition is solved. There are several free and commercial score recognition programs. They interpret even manually written scores. Examples:

- **SharpEye**[^29] is one of leading score scan and recognition programs. It saves result to computer format (MusicXML, NIFF, MIDI). The recognition is sometimes less successful on textual parts of music score as these programs are not oriented to OCR.

- Commercial score editor **Sibelius**[^30] “comes with two companion applications...With AudioScore Lite, you can input notes by singing or playing a monophonic instrument through a microphone. With PhotoScore & NotateMe Lite, you can turn printed, PDF, and JPEG sheet music into editable scores—and even hand-write music”.

- Site **Musipedia**[^31] proposes search for music in Internet. The melody can be played from MIDI or virtual keyboard, or whistled in front of microphone. There are also search by contour (graphical pattern formed by sequential notes) and search by rhythm tapped on a specified key.

For **chemistry**, formulas and equations in textual form can be processed by standard OCR systems. The recognition of chemical structural diagrams is an unsolved problem.

For **mathematics**, simple formulas can be OCR-ed, but all cur-

[^29]: https://www.mymusictools.com/eng/download/recognition-not/
[^30]: https://www.avid.com/sibelius-ultimate/features
[^31]: https://www.musipedia.org/query_by_tapping.html
rently existed systems fail on complicated formulas. The situation with commutative diagrams and geometrical drawings is the same as with chemical diagrams: the problem is unsolved.

For **graphical design content**, the situation is the same as with diagrams and drawings in chemistry and mathematics.

To recognize **chess diagrams** for the purpose of further position analysis, a program for the mobile phone ChessOCR\(^{32}\) is proposed. ChessOCR exports its result in the Portable Game Notation (PGN) format. This is a simple scripting format to record chess games. In addition to moves and variations, PGN file can contain metadata like name of the tournament, names of players, date, etc. A similar program Chessify\(^{33}\) has been announced, but its site has said “Coming soon!” on the corresponding page (as on June 12, 2019).

### 5 Problems and Challenges

#### 5.1 Page structure analysis

As we have seen, at present there are few possibilities to automate recognition of heterogeneous content. If the page contains such content, even the analysis of its structure with the detection of content types in blocks becomes very complex. It is necessary to apply a specific recognition program over each block corresponding to the type of content while in many cases the recognition task is not solved. Only semi-automation is possible here to support manual work.

And, finally, it is not clear what to do with the results of such recognition. How to integrate them into the script image of the page?

#### 5.2 Result format

Integration of heterogeneous content recognition results can be solved by export or conversion of those in the same format. The first candidate

\(^{32}\)https://chessocr.en.aptoide.com/

\(^{33}\)https://chessify.me
for this role is \LaTeX{} but the implementation may require significant improvements in it. The second obvious option is to use XML-based formats.

5.3 Uncovered issues

We list here some related topics that we haven’t discussed. This list don’t pretend to be exhaustive.

- Low-level conversions like conversion of raster images to vector graphics.
- Font generation from text image.
- Script description of the dependent parts of the document. Suppose we have a table, and its contents is visualized as a chart. A good script file shouldn’t duplicate the same information but simply accompany the data table with chart parameters like chart type, axes type, colors, thicknesses, markings, inscriptions, etc. The image will be regenerated each time we need it. The advantage is that if we change the table the chart will be corrected during regeneration.
- The ability to take data from other programs, for example, a table from Excel or Word.
- Geometrical scripting languages like older VRML\textsuperscript{34} or newer X3D\textsuperscript{35} etc.

5.4 Platform for Recognition of Heterogeneous Documents

The previewed minimal functionality of platform for recognition of heterogeneous documents is presented in Tab. 1

\textsuperscript{34}http://www.graphics.stanford.edu/courses/cs248-98-fall/Assignments/Assignment3/VRML2_Specification/spec/index.html

\textsuperscript{35}ISO/IEC 19775/19776/19777, http://www.web3d.org/x3d/what-x3d/
### Table 1. Recognition of heterogeneous documents: platform functionality

<table>
<thead>
<tr>
<th>Input data</th>
<th>Process</th>
<th>Resulting data</th>
</tr>
</thead>
<tbody>
<tr>
<td>A: Graphical document (paper, photocopy, etc.)</td>
<td>(P1): Imaging/scanning</td>
<td>B: Page image(s) in electronic form</td>
</tr>
<tr>
<td>B: Page image(s) in electronic form</td>
<td>(P2): Image quality improvement</td>
<td>B: Page image(s) in electronic form with better quality</td>
</tr>
<tr>
<td>B: Page image(s)</td>
<td>(P3): Page layout (structure) analysis</td>
<td>C: Page map(s)</td>
</tr>
<tr>
<td>D: Page fragment(s)</td>
<td>P4: Fragments recognition according to type(s) of fragment(s)</td>
<td>E: Script equivalent(s) of page fragment(s)</td>
</tr>
<tr>
<td>E: Script equivalent(s) of page fragment(s)</td>
<td>P5: Task distribution for manual verification</td>
<td>F: Extracted metadata</td>
</tr>
<tr>
<td>F: Extracted metadata</td>
<td></td>
<td></td>
</tr>
<tr>
<td>C: Page map(s)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>D: Page fragment(s)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>B: Page image(s)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>E: Script equivalent(s) of page fragment(s)</td>
<td>[P6]: Human verification; correction if necessary</td>
<td>E: Script equivalent(s) of page fragment(s)</td>
</tr>
<tr>
<td>F: Extracted metadata</td>
<td></td>
<td>F: Extracted metadata</td>
</tr>
</tbody>
</table>
It’s obvious that manual verification ([P6], [P11]) will be per-
formed by experts in the corresponding areas. It implies that the
platform will be Web-based like Wikipedia or version control systems.

The platform may be implemented using files, or using database.

In the first case, the set of page maps (in files) will be a base
to create the script presentation of the document. As recognition,
verification and correction are performed, these map files are filled with
additional information and, as the result, they became the full script
equivalents of the document pages. Use of the XML format may be
suitable.

In the second case, all extracted information beginning with page
maps is kept into DB records. Images and scripts may be kept also in
separate files while storing file names in the DB. The resulting script
equivalent of the document is generated from DB records.

6 Conclusions

OCR automation for documents with heterogeneous content is cur-
rently impossible. The goal should be to maximize the support of
manual work with further development to semi-automation under hu-
man control. We can preview several direction of development.

The first direction is effective and robust analyze of page structure.
Currently neural networks seem to be a perspective approach.

XML is the obvious presentation of the page analysis result. After
the recognition the generated scripts should be integrated into this
XML file. It’s necessary to develop an appropriate XML schema and
manipulation algorithms. Another direction is to convert all obtained
scripts into a universal format for a visualizer.

If LA T E X will be selected, additional macro packages for it should
be implemented, or adaptation of existing packages may be needed.

Another direction of development is the replacement of LATEX. (The
LaTeX3 Project seems to be hopelessly long-term.) Yes, LATEX is
much more than a simple text-formatting tool but it isn’t a book pub-

\url{https://www.latex-project.org/latex3/}
lishing system. \LaTeX{} has a number of fundamental flaws, especially its typographical (non-computer) paradigm, in which the ink applied to the page cover the previous layer and can’t be made transparent. Integration with Postscript removes this problem only partially.

Finally, recognizers should be developed that are absent today, for example, for structural formulas of arbitrary complexity in chemistry, formulas of arbitrary complexity in mathematics, commutative diagrams in mathematics, geometrical and technical drawings, etc. A desirable feature of recognizers is metadata extraction, at least semi-automatically. Again, neural networks provide some hope here.

The coordination of this activity is also necessary. Web platform for recognition of heterogeneous documents could be implemented to integrate all used tools.

References


Early Detection of Signs of
Anorexia in Social Media

Ciprian-Gabriel Cuşmuliuc, Lucia-Georgiana Coca, Adrian Iftene

Abstract

In recent years social media has started to play a crucial role in every person's life. People spend more and more time on it and so it is an online portrait of our persona. From the information we post and our interest a psychological profile can be created using specialized techniques in order to early identify different signs of diseases such as: anorexia, depression, suicide risk and many more. This paper aims to analyze different techniques of detecting early signs of anorexia in social media, their performance and how we can fine tune them in order to improve the actual results.

Keywords: Anorexia, Social Media, Machine Learning.

1 Introduction

The interest in exploiting information in social networks has increased more and more in recent years. There are currently several evaluation exercises using social networking data such as SemEval, where there are tasks on identifying feelings, aggressive language, offensive, or hate identification and CLEF, where there are tasks to identify protests, and the rapid identification of various symptoms.

Given how in recent years social media has become more and more part of people’s lives [1] and how the online persona is resembling more and more to the real life persona a psychological profile can be created using specialized techniques [2].

This profile can serve a vital part in our health as it can signal some early signs of mental illness and could help us better overcome our problems from developing stages before it’s too late. Such diseases are depression and anorexia, they are common illnesses that negatively affect
feelings, thoughts and behaviors and can harm regular activities like sleeping and social interaction. It is a leading cause of disability and many other diseases [3]. According to WHO (World Health Organization)\(^1\) statistics, more than 300 million people over the world are affected by depression and in each country at least 10% are provided treatment, anorexia is also very dangerous, according to National Eating Disorder Association, USA, 70 million people of all ages suffer from anorexia\(^2\).

For depression poor recognition and treatment may aggravate heart failure symptoms, precipitate functional decline, disrupt social and occupational functioning, and lead to an increased risk of mortality [4] and for anorexia the physical consequences are even more devastating, the consequences range from bone fragility to the shutdown of major body systems.

From the arguments above it can clearly be seen why early detection of these disease is so important and in this paper we will be detailing how we can do such a thing with different machine learning frameworks. The aim is to train multiple classifiers using the training set to identify anorexia or depression of the individual documents of the test sets. The performance of a text classification technique is highly dependent on the potential features of a corpus.

The dataset was provided by the CLEF eRisk\(^3\) organization, one dataset being for depression and one for anorexia. This paper focuses on anorexia but these techniques can also be used for depression as they are very similar and the algorithms would be the same.

This paper is organized as follows: Section 2 describes similar solutions; Section 3 gives a description of the task. Section 4 details the model we developed and then Section 5 details the results we obtained; finally Section 6 concludes this paper and describes our future work.

---

1 \[https://www.who.int/mental_health/management/depression/en/\]
2 \[https://www.nationaleatingdisorders.org/CollegiateSurveyProject\]
3 \[https://early.irlab.org/\]
2 Similar Solutions

Similar solutions that tackle the same problem can be found in the CLEF 2018 eRisk Task one. The best submissions are from the following sources [5]:

1. FH Dortmund with the BCSGD submission [6];
2. Universidad Nacional de San Luis, Argentina with algorithms: UNSLB and UNSLD [7].

There are multiple submissions, such from: Universitat Pompeu Fabra (Spain) [8], Tokushima University (Japan) [9] and other 5, but the best three are the above mentioned.

The best technique, FH Dortmund with BCSGD, uses a CNN neural network with a 300 dimensional fastText word Embedding feature matrix. The performance of this model is: ERDE5 has an error of 12.15% and on ERDE50 is 5.96% (which is the best result of them all), the F1 is 0.81%, the Precision is 0.75% and the Recall is 0.88%.

The second best model, the UNSL with UNSLB and UNSLD submissions, this team implemented a variant based on a model of flexible temporal variation of terms (FTVT) [10] and another variant based on sequential incremental classification (SIC).

The first model follows a semantic representation of documents that explicitly considers that the information available at each chunk is partial.

The second model is a novel text classification approach that incrementally estimates the association of each individual to each class based on the accumulated evidence.

The results are the following: ERDE5 has an error of 11.40% and on ERDE50 is 7.82% (which is the best result of them all), the F1 is 0.61%, the Precision is 0.75% and the Recall is 0.51% for UNSLB and for UNSLD the ERDE5 has an error of 12.93% and on ERDE50 is 9.8% (which is the best result of them all), the F1 is 0.79%, the Precision is 0.91% and the Recall is 0.71%.

Other models used to solve this problem are: TF-IDF with CNN, Latent Dirichlet Allocation with Multilayer Perceptron and flexible temporal variation of terms and another variant based on sequential incremental classification.
3 Task Description
3.1 Dataset
The anorexia dataset is a collection of posts or comments from a set of users over Reddit [5]. This corpus is divided into two categories - posts from users suffering from anorexia and posts from users that are a control group and do not have anorexia.

The following table better describes the main statistics of the train and test collection:

<table>
<thead>
<tr>
<th></th>
<th>Train</th>
<th>Test</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Anorexia</td>
<td>Control</td>
</tr>
<tr>
<td>Num. subjects</td>
<td>20</td>
<td>132</td>
</tr>
<tr>
<td>Num. submissions (posts &amp; comments)</td>
<td>7,452</td>
<td>77,514</td>
</tr>
<tr>
<td>Avg num. of submissions per subject</td>
<td>372.6</td>
<td>587.2</td>
</tr>
<tr>
<td>Avg num. of days from first to last submission</td>
<td>803.3</td>
<td>641.5</td>
</tr>
<tr>
<td>Avg num. words per submission</td>
<td>41.2</td>
<td>20.9</td>
</tr>
</tbody>
</table>

Table 1: Statistics of the train and test dataset.

The data is available via XML format and a small example of a user post is the following - from subject9942_1.xml:

```xml
<WRITING>
  <TITLE></TITLE>
  <DATE> 2015-02-09 13:50:12 </DATE>
  <INFO> reddit post </INFO>
  <TEXT> a system based on rules and works to get you 'whatever the goal is': enlightenment, nirvana or salvation in the case of Christianity</TEXT>
</WRITING>
```

Figure 1: Example of user post.

It can clearly be seen that we are provided a user post title, date of post and post content. It is very important that since the posts are in order they should also be processed in the same manner, as this shows the
evolution of the user’s disease. The same can be said for the depression dataset, the only difference would be that the statistics are a bit different.

3.2 Evaluation metrics

The evaluation metrics are of two types: standard one that we used and the CLEF eRisk ERDE metric [5]. The standard types of metrics that were used are the following: Precision, Recall, and F1 and to combine the result of negative and positive classifications micro avg., macro avg. and weighted avg. were used.

The CLEF ERDE metric was introduced because standard classifications measures are time-unaware and do not penalize late decisions. Therefore, the evaluation must have a new measure of performance that rewards early alerts. More specifically ERDE, an error measure for early risk detection [5] for which the fewer writings required to make the alert, the better. For each user the evaluation proceeds as follows. Given a chunk of data, if a system does not emit a decision then it has access to the next chunk of data (i.e. more submissions from the same user) thus the system gets a penalty for late emission.

ERDE takes into account the correctness of the decision and the delay taken by the system to make the decision. The delay is measured by counting the number of distinct submission seen before taking the decision. The calculation formula is the following:

\[
ERDE_o(d, k) = \begin{cases} 
  c_p & \text{if } d = \text{positive AND ground truth = negative (FP)} \\
  c_f & \text{if } d = \text{negative AND ground truth = positive (FN)} \\
  I_c(k)c_p & \text{if } d = \text{positive AND ground truth = positive (TP)} \\
  0 & \text{if } d = \text{negative AND ground truth = positive (TN)}
\end{cases}
\]

Figure 2: ERDE calculation formula [5].

The minimum value of \( o \) is considered as 5 and the maximum value as 50. Note that ERDE lies in range \([0, 1]\). A low value of ERDE is desirable as this is a measure to find error in the system.

4 Models

The models that have been created to solve this problem will be described in this section combined with methods of preprocessing the data and extracting features. To achieve this several Python libraries have been
used: sklearn, PySpark with PySpark MLlib and Tensorflow with Keras. There are 5 models with the following characteristics: sklearn Multilayer Perceptron Classifier and on feature extraction TF-IDF and Latent Dirichlet Allocation, Convolutional Neural Network and Bidirectional Long Short Term Memory Network is used (CNN-LSTM) with fastText feature extraction based on Tensorflow with Keras and finally Pyspark Multilayer Perceptron Classifier but with three different types of feature extraction: CountVectorizer with IDF, TF-IDF and Word2Vec. Based on these five models a performance score will be generated and a comparison between our result and others will be performed in order to create better models in the future.

4.1 Preprocessing

Before feeding the data to the machine learning algorithms we had to preprocess the text and extract features [11]. This section describes in detail this process in order to fully understand the training data feed to the algorithms.

In order to find the best solution for the classification problem, multiple types of preprocessing and feature extraction techniques were used such that the best could be selected similar to [12], [13].

For preprocessing a “Tokenizer” was used and applied for each line, taking text (such as a sentence) and breaking it into individual terms (words). After the tokenization process the stop words were removed with the “StopWordsRemover” class, given the fact that the texts are in English we only removed stop words from this language. Irony or sentiment analysis was not taken into account in the preprocessing part as feature extraction could represent this indirectly, it was a known risk that this could impact certain edge cases of the classification.

After the preprocessing of the text it was necessary to extract features from it, a multitude of methods were used, trying to find the perfect fit for every algorithm. As stated in the introduction of this section almost every algorithm uses a different feature extraction method in order to maximize its performance.

4.2 Feature Extraction

For the first algorithm, a Sklearn implementation of Multilayer Perceptron Classifier (short MLP), it was used as feature extractor TF-IDF and after that Latent Dirichlet Allocation. The LDA model is applied on a term-
document matrix of the users, where the element at position $ij$ is the relative frequency of term $i$ in document $j$. The term-document matrix was restricted to 3,000 most frequent n-grams of length 1 to 3 (without stop words). Experimentally it was found that the model worked best on 30 topics.

The second algorithm, an implementation of Convolutional Neural Network and Bidirectional Long Short Term Memory Network is used (CNN-LSTM) in Tensorflow with Keras, uses a new feature extraction technique called fastText that is a library for learning of word embedding’s and text classification created by Facebook’s AI Research lab. It creates a word embedding vector from a corpus. The settings of this feature extractor are the following: size of features 100 (for dimensionality), the context window size is 50, the minimum word count is 5 and the iteration number is 50.

The third set of algorithms, based on PySpark and the base classification model is the same, Multilayer Perceptron, but the feature extraction methods are slightly different, depending on what configuration. The first variation uses a CountVectorizer with IDF, CountVectorizer aims to help convert a collection of text documents to vectors of token counts, the model produces sparse representations for the documents over the vocabulary, the settings of CountVectorizer are the following: minimum term frequency is 1 and so is the minimum definition frequency, the maximum definition frequency is $2^{63}-1$ and the vocabulary size is 30,000 and IDF has a minimum doc frequency of 2. These settings were determined experimentally. The second variation uses the classical TF-IDF with the PySpark implementation HashingTF, this would create a feature map where a raw feature is mapped into an index (term) by applying a hash function, after which the IDF would take the generated term frequency vectors to fit which scales each feature and down-weighs features that appear frequently in the corpus. For settings, the feature no is 6,000 (as this would force us to create the same number of input layers for the neural network, we had to scale it down to this value). The last preprocessing technique uses Word2vec model, it is very similar to fastText, and it provides an efficient implementation of the continuous bag-of-words and skip-gram architectures for computing
vector representations of words. The settings for word2vec are: the feature vector size is limited to 500 features.

4.3 Models

To continue what was described in the last section, after the preprocessing and feature extraction the classification algorithms that would predict the risk of anorexia would be needed. This part describes what algorithms were used and how we combined them to get the results in the next section.

The first model, the Sklearn MLP, ran a pipeline consisting of TF-IDF after which the output would be fed to LDA and in the end to a Multilayer Perceptron Classifier with the following settings that were experimentally found: the maximum iterations are set to 1500, the activation function is set to identity and the weight optimization is set to adam and the hidden layers are 2 with dimensions of 60 and 30 perceptron’s.

The second model, based on Tensorflow and Keras, implements a combination Convolutional Neural Network and Bidirectional Long Short Term Memory Network is used (CNN-LSTM) that takes input from fastText feature extractor that is based upon a tokenized corpus, and thus the feature vectors are fed to the algorithm to produce a classification. The settings of the model are the following: the batch size is set to 500, the number of epochs is 100. To prevent overfitting if the model does not begin to improve training will come to an end (using the EarlyStopping callback function). The model has 3 convolutions of a one dimension (Conv1D in keras) (since we have a feature vector not a matrix) with kernel size three, 32 output filters, the first two have ELU activation function and the last layer has a relu activation function. After these three convolutions a Max pooling is done on the output in order to downsample the data, this feeds the next component which is a Bidirectional LSTM neural network that outputs a 512 sized vector. The output of the LSTM is fed to three neural layers that output a 512 sized vector and has a sigmoid activation function; in the end this input is sent to a neural layer that outputs a two sized vector with softmax activation, this would be the final, classification layer.

The third model, based on Pyspark, also uses MLP but with three different feature extractions. The first one, using CountVectorizer, the
settings were the following: maximum iterations were set to 2,000, the input layer was of size 30,000, it had two hidden layers of 80 and 100 and the optimization was set to l-bfgs. The second model, that uses TF-IDF, has settings very similar to the first one, the only differences being that the input layer size was of size 6,000. The same can be said for the last one, the Word2vec, here the vector feature size is only 500. It was tried to have very similar settings between these as a more accurate measurement could be performed.

These models can also be applied on the depression dataset without any changes and see what result it yields, the next section focuses on the results of the anorexia dataset however we also applied these algorithms with success on the aforementioned dataset.

5 Results

In this section the results of the algorithms will be measured. Before starting the algorithms will be labeled for simplicity, the first implementation with sklearn will be called LSLDA, the second model CNNLSTM and the last one will be called: for the CountVectorizer will be called CVMLP, the TF-IDF will be called IDFMLP and the Word2Vec W2VMLP.

The next tables will show the performance all the algorithms based on Precision, Recall, F1 and confusion matrices:

<table>
<thead>
<tr>
<th>LSLDA</th>
<th>Precision</th>
<th>Recall</th>
<th>F1</th>
<th>support</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0.95</td>
<td>0.65</td>
<td>0.77</td>
<td>279</td>
</tr>
<tr>
<td>1</td>
<td>0.24</td>
<td>0.78</td>
<td>0.37</td>
<td>41</td>
</tr>
<tr>
<td>Micro avg</td>
<td>0.66</td>
<td>0.66</td>
<td>0.66</td>
<td>320</td>
</tr>
<tr>
<td>Macro avg</td>
<td>0.60</td>
<td>0.71</td>
<td>0.57</td>
<td>320</td>
</tr>
<tr>
<td>Weighted avg</td>
<td>0.86</td>
<td>0.66</td>
<td>0.72</td>
<td>320</td>
</tr>
</tbody>
</table>

Table 2: LSLDA results.
LSLDA confusion matrix:

<table>
<thead>
<tr>
<th></th>
<th>LSLDA Predicted No</th>
<th>LSLDA Predicted Yes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Actual No</td>
<td>180</td>
<td>99</td>
</tr>
<tr>
<td>Actual Yes</td>
<td>9</td>
<td>32</td>
</tr>
</tbody>
</table>

Table 3: LSLDA confusion matrix.

<table>
<thead>
<tr>
<th>CNNLSTM</th>
<th>Precision</th>
<th>Recall</th>
<th>F1</th>
<th>support</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0.87</td>
<td>1</td>
<td>0.93</td>
<td>279</td>
</tr>
<tr>
<td>1</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>41</td>
</tr>
<tr>
<td>Micro avg</td>
<td>0.87</td>
<td>0.87</td>
<td>0.87</td>
<td>320</td>
</tr>
<tr>
<td>Macro avg</td>
<td>0.44</td>
<td>0.50</td>
<td>0.47</td>
<td>320</td>
</tr>
<tr>
<td>Weighted avg</td>
<td>0.76</td>
<td>0.87</td>
<td>0.81</td>
<td>320</td>
</tr>
</tbody>
</table>

Table 4: CNNLSTM results.

CNNLSTM confusion matrix:

<table>
<thead>
<tr>
<th>CNNLSTM</th>
<th>Predicted No</th>
<th>Predicted Yes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Actual No</td>
<td>279</td>
<td>0</td>
</tr>
<tr>
<td>Actual Yes</td>
<td>41</td>
<td>0</td>
</tr>
</tbody>
</table>

Table 5: CNNLSTM confusion matrix.
Early Detection of Signs of Anorexia in Social Media

<table>
<thead>
<tr>
<th>CVMLP</th>
<th>Precision</th>
<th>Recall</th>
<th>F1</th>
<th>support</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0.96</td>
<td>0.98</td>
<td>0.97</td>
<td>279</td>
</tr>
<tr>
<td>1</td>
<td>0.86</td>
<td>0.73</td>
<td>0.79</td>
<td>41</td>
</tr>
<tr>
<td>Micro avg</td>
<td>0.95</td>
<td>0.95</td>
<td>0.95</td>
<td>320</td>
</tr>
<tr>
<td>Macro avg</td>
<td>0.91</td>
<td>0.86</td>
<td>0.88</td>
<td>320</td>
</tr>
<tr>
<td>Weighted avg</td>
<td>0.95</td>
<td>0.95</td>
<td>0.95</td>
<td>320</td>
</tr>
</tbody>
</table>

Table 6: CVMLP results.

CVMLP confusion matrix:

<table>
<thead>
<tr>
<th>CVMLP</th>
<th>Predicted No</th>
<th>Predicted Yes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Actual No</td>
<td>274</td>
<td>5</td>
</tr>
<tr>
<td>Actual Yes</td>
<td>11</td>
<td>30</td>
</tr>
</tbody>
</table>

Table 7: CVMLP confusion matrix.

<table>
<thead>
<tr>
<th>IDFMLP</th>
<th>Precision</th>
<th>Recall</th>
<th>F1</th>
<th>support</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0.94</td>
<td>0.97</td>
<td>0.96</td>
<td>279</td>
</tr>
<tr>
<td>1</td>
<td>0.76</td>
<td>0.61</td>
<td>0.68</td>
<td>41</td>
</tr>
<tr>
<td>Micro avg</td>
<td>0.93</td>
<td>0.93</td>
<td>0.93</td>
<td>320</td>
</tr>
<tr>
<td>Macro avg</td>
<td>0.85</td>
<td>0.79</td>
<td>0.82</td>
<td>320</td>
</tr>
<tr>
<td>Weighted avg</td>
<td>0.92</td>
<td>0.93</td>
<td>0.92</td>
<td>320</td>
</tr>
</tbody>
</table>

Table 8: IDFMLP results.
IDFMLP confusion matrix:

<table>
<thead>
<tr>
<th>IDFMLP</th>
<th>Predicted No</th>
<th>Predicted Yes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Actual No</td>
<td>271</td>
<td>8</td>
</tr>
<tr>
<td>Actual Yes</td>
<td>16</td>
<td>25</td>
</tr>
</tbody>
</table>

Table 9: IDFMLP confusion matrix.

<table>
<thead>
<tr>
<th>W2VMLP</th>
<th>Precision</th>
<th>Recall</th>
<th>F1</th>
<th>support</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0.97</td>
<td>0.96</td>
<td>0.97</td>
<td>279</td>
</tr>
<tr>
<td>1</td>
<td>0.77</td>
<td>0.83</td>
<td>0.80</td>
<td>41</td>
</tr>
<tr>
<td>Micro avg</td>
<td>0.95</td>
<td>0.95</td>
<td>0.95</td>
<td>320</td>
</tr>
<tr>
<td>Macro avg</td>
<td>0.87</td>
<td>0.90</td>
<td>0.88</td>
<td>320</td>
</tr>
<tr>
<td>Weighted avg</td>
<td>0.95</td>
<td>0.95</td>
<td>0.95</td>
<td>320</td>
</tr>
</tbody>
</table>

Table 10: W2VMLP results.

W2VMLP confusion matrix:

<table>
<thead>
<tr>
<th>W2VMLP</th>
<th>Predicted No</th>
<th>Predicted Yes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Actual No</td>
<td>269</td>
<td>10</td>
</tr>
<tr>
<td>Actual Yes</td>
<td>7</td>
<td>34</td>
</tr>
</tbody>
</table>

Table 11: W2VMLP confusion matrix.

The algorithms were also ran with the official ERDE metrics with \( o = 5 \) and \( o = 50 \) and the script also has some F1, Precision and Recall that seem to have a bit of a different calculation from the sklearn as the result have a small difference, nonetheless this will be taken into consideration when a conclusion has been made.
To analyze the performance of the algorithms, it can clearly be seen from all the data that CVMLP that uses CountVectorizer and Multilayer Perceptron has the best results, on the contrary the worse algorithm is CNN with LSTM that is not able to classify well positive cases and this would require further fine tuning.

To compare it with the best results, from FH Dortmund and UNSL, the FH Dortmund had an ERDE$_5$ of 12.15% where the CVMLP had 12.95% which is a very close call and on the long run, the ERDE$_{50}$ where they had 5.96% and we had 3.33%, almost two time improvement over the best algorithm at this lab, the F1 of our version is the save 0.81 and their 0.81, our precision 0.86 over 0.75 and the recall of 0.76 being lower than 0.88. For the UNSL with the UNSLB implementation where they had an ERDE$_5$ of 11.40%, slightly lower than our 12.95%, the ERDE$_{50}$ of 7.82%, way higher than our 3.33% and the F1 of 0.61 lower than our 0.81, the precision was 0.75 versus ours of 0.86 and the recall of 0.51, lower than our 0.76; for the UNSLD with an ERDE$_5$ of 12.93% being the save as our 12.95%, ERDE$_{50}$ of 9.85% which is almost three times higher than our 3.33% and the F1 of 0.79 versus our 0.81, the Precision of 0.91 which is a bit better than our 0.86 and finally a recall of 0.71, slightly lower than our 0.76.

To conclude, our best algorithm, CVMLP, has one of the best ERDE$_{50}$ from all the implementation, average ERDE$_5$ and also high Precision, Recall and F1.
6 Conclusions

To conclude, in this paper we mainly discussed about ways to early detect anorexia in social media but also we highlighted how we can apply the same methods on depression, we saw multiple implementations in order to measure their performance and compare to see which one is better, the CVMLP algorithm was the best and compared with other submissions it is up there with the best, the worse algorithm CNNLSTM has weak results not being able to classify an anorexia case needing further improvement.

As future work we would like to improve on the latter algorithm knowing it is high potential but also fine tune our existing ones in order to perform much better.

Acknowledgments. We thank all the volunteers involved in the development of this application. This work is partially supported by POC-A1-A1.2.3-G-2015 program, as part of the PrivateSky project (P_40_371/13/01.09.2016).

References


Early Detection of Signs of Anorexia in Social Media


Ciprian-Gabriel Cușmuliuc, Lucia-Georgiana Coca, Adrian Iftene
Faculty of Computer Science, “Alexandru Ioan Cuza” University of Iasi, Romania
E-mail: {cipriancus, cocageorgiana}@gmail.com, adiftene@info.uaic.ro
Building a Diachronic Parallel Corpus for the Alignment of the Old Romanian Texts

Tudor Bumbu

Abstract

Building parallel corpora is the focus of many groups in Corpus Linguistics because data based approaches within Natural Language Processing requires large amounts of linguistic resources in the language studies. In this paper, we describe our processing steps to build a diachronic parallel corpus based on the oldest canonical gospels printed using Romanian Cyrillic alphabet. Our data resources are the scanned images of the New Testament printed in 1648 and the electronic version of the New Testament translated and revised in Romanian in 1994. The corpus will serve as a fundamental resource for analyzing the diachronic aspects of the language and for aligning old Romanian text form to a modern text form.

Keywords: diachronic parallel corpus, old text alignment, Romanian Cyrillic.

1 Introduction

Old historical texts differ from modern texts by the absence of standardized writing conventions. This phenomenon occurs because the natural languages are living organisms and they are ever developing in terms of phonology, lexicon, sentence structure and many other linguistic features. In most cases, the development of the languages happens too slowly and it cannot be observed across adjacent generations. However, taking into account a long period we can easily notice the differences
between the text written 300 years ago and the same thoughts put into words these days.

The goal of our research is to build a diachronic parallel corpus of the New Testament book of the 17th century and the New Testament of the 20th century and further to perform diachronic analysis of the Romanian as well as to align the old text to the modern text form.

This paper presents only the process of building a parallel diachronic corpus of old Romanian texts using particular technics for optical character recognition, transliteration and fuzzy metrics to align the sentences in the corpus.

It is a part of a bigger research project and the overall result is a tool pack, which will serve at automation of the alignment of old text to the modern text. Aligning old texts to their modern representation means translating them into a contemporary language by replacing the phrases that are out of use and archaisms with modern expressions. Any person speaking a contemporary language should easily interpret the resulted text. The work further presented in this paper is based on the process of developing the diachronic parallel corpus.

2 Steps in Diachronic Parallel Corpus Building

In respect to diachronic corpora, there were several conferences, such as Diachronic Corpora, Genre, and Language Change, Nottingham, 8-9 April 2016, Digital Access of Textual Cultural Heritage DATeCH Gettigen, 1-2 June 2017, Language Change in English and Beyond: Linguistic Theory and Historical Corpora, Athens, 14-15 March 2019.

Europeana Collections brings together digitized cultural heritage material from over 3,500 cultural heritage institutions.

Modern natural language processing studies focus on diachronic and dialectal variations of languages based on historical corpora and grow very fast in importance. Diachronic corpora are created for many languages. At the International Pragmatics Conference in Manchester in 2011 there are papers on diachronic interpretation of corpora: Taavitsainen et al., 2014 [1]. The alignment of the oldest New Testament is found in Haug et al., 2008 [2].
Building a Diachronic Parallel Corpus for the Alignment of the Old Romanian Texts

The paper Malahov L. et al, 2017 [3] is the main guideline of our research and it is intended for enlarging the entire diachronic Romanian corpus.

3 Steps in Diachronic Parallel Corpus Building

The steps in diachronic parallel corpus building vary by case. Our main source is the scanned images of the Bible of the 17th century printed using Romanian Cyrillic alphabet. The title of the book is “Noulu Testamentu sau Împacarea, au Leagea noao a lui Is. Hs.” and it was printed in 1648 at Cetatea Belgradului, Ardeal [4]. The second resource is the electronic version of the New Testament from the 1994 edition printed by the Interfaith Bible Society in Romania reproducing the text of the 1982 edition [5].

Taking into account the fact that the alphabet used in the main resource is old and unused, it has to pass through several steps before it can be used as an editable text with Latin alphabet.

The first step is to transform the scanned images into editable text format. It is a challenging task because there is no OCR program, which has old Romanian Cyrillic language of the 17th century. We had to train a new language using ABBYY FineReader software [6]. The printed book is going to be entirely converted into editable Cyrillic alphabet and then in the next step transliterated to the Latin alphabet. The optical character recognition of this book is a very long and difficult process because the scanned images are in a bad condition and the printing has a lot of specific features such as: letters over other letters, wide usage of accents, etc.

Figure 1. A fragment from New Testament page (1648)
We have focused only on the base text, the text from the first column of the table shown in the figure 1. The recognition accuracy was above 75% and the correction of the errors was done manually by comparing it with the scanned image. We obtained 3779 sentences in Cyrillic Romanian, which cover the four canonical gospels – Matthew, Mark, Luke and John and saved it to txt format.

The next step is to convert the Cyrillic alphabet into Latin. At this iteration, we used a transliteration tool developed at the Institute of Mathematics and Computer Science “V. Andrunachievici” [7]. A sample of transliterated text is shown in the figure 2. We obtained the same amount of sentences but in the Latin alphabet. This is the moment when the main resource reached the level of being processed together with the second resource.

Table 1. Sample of (a) OCRed text and (b) Transliterated text

<table>
<thead>
<tr>
<th>a) OCRed Text</th>
<th>b) Transliterated text</th>
</tr>
</thead>
</table>
| Nu giudecareț casă nu fiț giudecaț.  
Că cu ce giudecată veți giudeca, giudecăveț: și cuce măsură veți măsura, săva măsura voao. | 
| Hi giudecareț căță hi fițć giudecăț.  
Ră kă țe giudecaț țe căță giudeca, giudecaț țe căță: iși kă țe mășcăță țe căță mășcăță, țeca mășcăță coluș. | 

The third step is to align the sentences from the main resource to the sentences from the second resource. We took 3500 sentences from each resource in part. Each sentence corresponds to a verse. The sentences were selected based on their order in the book. 1006 sentences from the Gospel of Mathew starting from the 4th chapter, 675 sentences from Mark, 1151 sentences from Luke and 668 from John. The structure and the lexicon of the same sentence in the old and modern resource vary. They can be very unalike but at the same time, there are sentences very close to each other. See table 1.
Table 1. Sample of aligned sentences from Gospel of Mark, chapter 1

<table>
<thead>
<tr>
<th>No.</th>
<th>Old text</th>
<th>Modern text</th>
<th>English text</th>
</tr>
</thead>
<tbody>
<tr>
<td>9.</td>
<td>Și fu îzilele acelia, veni Iisus de în nazarethul galileei, și să boteză dela Ioan în Iordan.</td>
<td>Și în zilele acelea, Iisus a venit din Nazaretul Galileei și S'a botezat în Iordan de către Ioan.</td>
<td>It happened in those days, that Jesus came from Nazareth of Galilee, and was baptized by John in the Jordan.</td>
</tr>
<tr>
<td>10.</td>
<td>Și aciși eşind de în apă, văzu deșchise ceruirele, și duhul ca un porumbu, pogorând spre El.</td>
<td>Și în data, ieșind din apă, a vazut cerurile deschise și Duhul ca un porumbel pogorându-Se peste El.</td>
<td>Immediately coming up from the water, he saw the heavens parting, and the Spirit descending on him like a dove.</td>
</tr>
<tr>
<td>11.</td>
<td>Și glas fu den ceruire, tu ești fiul meu cel iubit, întru carele bine voescu.</td>
<td>Și glas s'a făcut din ceruri: &quot;Tu ești Fiul Meu Cel iubit, întru Tine am binevoit&quot;.</td>
<td>A voice came out of the sky, &quot;You are my beloved Son, in whom I am well pleased.&quot;</td>
</tr>
<tr>
<td>12.</td>
<td>Și numai decât scoase pre El duhul împuștie.</td>
<td>Și în data Duhul L-a scos în pustie.</td>
<td>Immediately the Spirit drove him out into the wilderness.</td>
</tr>
<tr>
<td>13.</td>
<td>Și era acolo împuștie, patruzece de zile și patruzece de nopti, ispiti de satana: și era cu fierile, și îngerii slujia lui.</td>
<td>Și a fost în pustie patruzece de zile, fiind ispiti de Satana. Și era împreuna cu fiarele, și îngerii Îi slujeau.</td>
<td>He was there in the wilderness forty days tempted by Satan. He was with the wild animals; and the angels were serving him.</td>
</tr>
<tr>
<td>14.</td>
<td>Iară după prinsoarea lui Ioan, veni Iisus îngalilea, propoveduind Evanghelie a Împărăției lui dumnezeau.</td>
<td>După ce Ioan a fost întemnițat, Iisus a venit în Galileea, propovăduind Evanghelia împărăției lui Dumnezeu</td>
<td>Now after John was taken into custody, Jesus came into Galilee, preaching the Good News of the Kingdom of God</td>
</tr>
</tbody>
</table>

To measure the similarity between the two sentences we use Fuzzy string matching based on Levenstein Distance implemented in the
fuzzywuzzy library in Python. The similarity ratio is measured in percent and we compared different combinations in order to obtain the best ratios.

Table 3. Similarity ratios between old text and modern text

<table>
<thead>
<tr>
<th>No.</th>
<th>Ratio type</th>
<th>Similarity (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>Simple ratio</td>
<td>68</td>
</tr>
<tr>
<td>2.</td>
<td>Token sort ratio</td>
<td>61</td>
</tr>
<tr>
<td>3.</td>
<td>Token set ratio</td>
<td>71</td>
</tr>
</tbody>
</table>

Simple ratio compares the entire sequence similarity, in order. Using it, we obtained a similarity of 68%.

Token sort ratio compares the entire sequence similarity but ignores word order.

Token set ratio compares the entire sequence similarity but ignores duplicated words and word order. Token set ratio gives us the result of 71% of similarity and it is the best fit for our text. According to this result, we decided to apply token set ratio to align the sentences which exceed the threshold of 70%. As we need the corpus for the task of making the automatic alignment of old text to modern text, the similarity between sentences should be high. We obtained more than 2700 sentence in our diachronic parallel corpus. The corpus is saved in txt format and is not annotated. In the next section we present some related and recent work in this direction.

4 Conclusion and Future Work

In this paper, we focused on creation of a diachronic parallel corpus of the New Testament book of the 17th century and the New Testament of the 20th century. We have presented the process of building a parallel diachronic corpus of old Romanian texts using two main series of actions. The first part was to digitize the images of the old text basing on: optical character recognition and transliteration. The second part was to apply fuzzy metrics based on Levenstein Distance to align the sentences in the corpus.
The main usage of the corpus is a further research in aligning the old texts to modern text form. It is a good idea to compare two texts from a diachronic point of view. From this comparison, we can partially generalize how the written language has developed and how to apply these differences in order to transform the old text into a modern one.

References


Tudor Bumbu

Vladimir Andrunachievici Institute of Mathematics and Computer Science
E-mail: tudor.bumbu@math.md
Iasi City Explorer
Camelia Miluț, Adrian Iftene, Daniela Gîfu

Abstract

Iasi City Explorer is a voice first application, developed as an Amazon Alexa skill. This application is the first Alexa skill, dedicated to the touristic potential of Iasi. Iasi City Explorer can offer all the information that you need to plan a very nice stay in this city. It includes recommendations for restaurants, cafes, car rentals, information about weather and it suggests activities suitable for a certain time of the day, including main attractions and places where you can pleasantly spend your time.

Keywords: Amazon Alexa, Amazon Echo, Speech recognition.

1 Introduction
Alexa is a smart personal assistant with voice interface, developed by Amazon. It is integrated with many physical devices such as smart home appliances (refrigerators, bulbs, thermostats, etc.), automotive, surveillance cameras, locks, mobile phones, but the most used are the Echo smart speakers launched by Amazon in 2014 [1]. Alexa listens to the voice commands received from the user and provides an appropriate response through which it performs a task [2]. Alexa for mobile devices allows the user to activate the embedded device by a keyword or by clicking a button [3, 4]. Alexa's name appeared for two reasons. The first, because of the consonant X found in name, the sound of which, the name does not resemble those of similar devices, to prevent their accidental activation. And the second reason would be for the connection to the Library of Alexandria - which was considered to be the keeper of the knowledge of all time [4, 5].
2 Similar Skills

Skills for Alexa, similar to Iasi City Guide, are:

- *Bucharest Guide*\(^1\): offers information about 33 points of interest in Bucharest.
- *Rome tour*\(^2\): A guide to tourist attractions in the capital of Italy. It contains the most popular locations in Rome, such as monuments or museums. Skill sends location addresses to companion application on mobile or TV.
- *NYC Guide*\(^3\): Provides information to tourists in New York such as tourist attractions or information about transportation.

3 Proposed Solution

The main purpose of *Iasi City Explorer* is to enhance the tourist experience in Iasi and also to help the newcomers to explore the city and find location easier, without the need to make intermediate search.

*Iasi City Explorer* is available on the Amazon Echo device, but also on web browsers, Android and iOS platform, via Amazon Alexa app, where the user can interact with the application through voice commands. The main reason for picking Alexa as a development platform is the increasing popularity of the Artificial Intelligence, implicitly of the digital assistants and voice first applications.

3.1 Architecture

\(^1\) https://www.amazon.com/Catalin-Batrinu-Bucharest-Guide/dp/B074WBNM7R
\(^2\) https://www.amazon.com/Chris-Cinelli-Rome-tour/dp/B07475HGSS
\(^3\) https://www.amazon.com/AlexSantisteban-New-York-City-Guide/dp/B076QGS2PS
At the development level, the application follows a cloud-based architecture [6], orchestrating services provided by Amazon Web Services [7] with Google Maps Platform and Yahoo Weather. The interaction between services can be seen in the Figure 2.

![Figure 1. System architecture.](image)

The user interacts with Alexa-integrated devices such as Amazon Echo, mobile applications, etc. The query is intercepted by Alexa Voice Services, and is forwarded to the Alexa Skills Kit that launches the Lambda function [8, 9]. Depending on the request received, Lambda takes data from Yahoo Weather API\(^5\), Google Places API\(^6\) or DynamoDB [10]. DynamoDB data is also obtained through the Google Places API. To monitor the Lambda function, CloudWatch metrics are recorded.

To access DynamoDB and CloudWatch, the Lambda function gets some permission through Identity and Access Management (IAM). The response processed using the Lambda function is passed back to the user through the Alexa Voice Service and the device through which the interrogation was made. In the following, we will present the technical details of this process.

### 3.2 Developing the Skill

An Alexa skill is composed of two parts:

\(^5\) https://developer.yahoo.com/weather/
\(^6\) https://developers.google.com/places
• **Skill interface**: The user interaction side, configured by Alexa Skill Console. This section defines how the user's voice commands are directed to the Skill Service.

• **Skill service**: contains the logic of the application, hosted on a remote server.

### 3.3 Extracting Data

To populate the database, a separate section of the application was created. This section includes a Node JS server that contains a DynamoDB client linked to the PLACE LIST table and a list of location types made by Google Places queries through Place Search. In order to get more information on the locations obtained from the query, they are provided as a Place Details function. The results are filtered and serialized in JSON format. Finally, serialized results are inserted into the table through the DynamoDB client.

### 4 Conclusion

Alexa was chosen as the application development platform because voice first applications are becoming increasingly popular, and one of the reasons is ease of use. An application that can be used by voice interaction can be easily used by visually impaired people or with motor disabilities or by people who are reluctant to interact with a touchscreen (the elderly, or children).

Another advantage of using Alexa as a development platform is portability. Besides the popular Echo devices, Alexa is integrated with many IoT devices, but also with classic smartphones, making it accessible at all times and in any place.

This application can contribute, also, to the popularity level of the city, bringing an Iasi related application to a new platform, with increasing popularity.

**Acknowledgments.** We thank all the volunteers involved in the development of this application. This work is partially supported by POC-A1-A1.2.3-G-2015 program, as part of the PrivateSky project (P_40_371/13/01.09.2016) and by the README project “Interactive and Innovative application for evaluating the readability of texts in Romanian

References


Camelia Miluț¹, Adrian Iftene¹, Daniela Gifu¹,²,³

¹Faculty of Computer Science, “Alexandru Ioan Cuza” University of Iasi, Romania
² Institute of Computer Science, Romanian Academy, Iasi branch
³ Cognos Business Consulting S.R.L., Bucharest, Romania
E-mail: {camelia.milut, adiftene, daniela.gifu}@info.uaic.ro @info.uaic.ro
Pocket World with Augmented Reality

Roxana Pantea, Adrian Iftene, Daniela Gîfu

Abstract

Currently, there are many devices and technologies that evolve across time. Augmented reality sits at the top of the list of the most used technologies as much in the gaming industry as well as for applications that are based on the educational software concept. This domain is appealing for both children and adults through the innovation that it brings to the traditional learning system. The application described in this paper has a target group that consists of users who wish to travel and discover new places. Each country has a rich historic content and at least a city with famous landmarks. We might find that we are not always properly informed regarding those landmarks. Pocket World is an application that helps recognizing these landmarks and finding out essential information about them as well as other adjacent landmarks.

Keywords: Augmented reality, Android, eLearning.

1 Introduction

A major limit in traditional learning system is that students, and sometimes educators, often are not enough prepared to engage themselves in a real world setting. A solution is the constructivist approach, being a well-established methodology for learning [11; 12]. In other words, students are working in teams to incorporate augmented reality (AR) technology, aiming to improve education experience [1; 13].

The authors of 2010 Horizon Report [8] also assert that the AR has a huge potential to provide a useful context for education, allowing learning and discovery experiences connected to real world information. Mixed
realities have been used in education for a long time [10]: the augmentation of wall paintings in caves is an approach to the transfer of knowledge about hunting and survival; Heilig’s patent for Sensororama describes the need for a solution for teaching, training and educating people in armed forces, industries and schools [6]; Ivan Sutherland saw the augmented image as a solution to give the user a “chance to get acquainted with concepts that cannot be achieved in the physical world” [14]; and the pioneering activity of Caudell and Mizel in Boeing’s augmented reality were designed to teach workers how to assemble complex components into aircraft [2]. ELearning applications using AR in recent years have focused on subjects such as chemistry [9], geography [3], history [7], astronomy [5], and lessons on mobile devices (smart phones or tablets) contain animations, movies, sounds, and so on.

The proposed software, called Pocket World AR, falls into the category of augmented reality systems for learning and teaching, which benefit from a multi-user online graphical environment. Basically, it can be used as a way to navigate through the landmarks from a city as well as a fast and fun learning method for children who are fascinated by the pictures they usually find in books. We chose to use augmented reality because it makes the user experience more interesting and it’s a very easy way to learn information about a landmark that the user might be sitting next to or has seen in a photograph. Pocket World provides the possibility to learn information about both a certain landmark and the adjacent sites by scanning or browsing through a vast list. The application also suggests nearby landmarks when the GPS is activated.

The paper is structured as follow: Section 2 refers to the Pocket World AR architecture and each main feature individually. The paper conclusions are given in section 3, mentioning some future directions for the development of this augmented learning.

2 SYSTEM ARCHITECTURE

This chapter describes the architecture of the application and each main feature individually. Figure 1 shows the four main features: Scan Landmark, Search Landmark, Lookup for Closest Landmarks and Add new landmark.
Figure 1. Global architecture of the application.

The main page of the Pocket World Application
The purpose of the main page of the application is to provide easy access to the main features of the application. It contains the application logo and 4 buttons, each triggering a certain module (see Figure 2).

Figure 2. The main page of the Pocket World Application and its layout.
Scan and identify the landmark

This module uses augmented reality to provide useful information when we are near a landmark. After selecting or identifying a landmark, the application opens a page with essential information and a representative photo of it. The information is extracted from the Wikipedia page of each landmark. Apart from the landmarks that are already in the database, the user can also add other landmarks that he sees relevant. He might want to mark the coordinates of a cozy coffee shop he likes or a place he heard about from a friend. Thus, all these points will be collected in a very friendly, easy to use application that is available for everyone.

The basis of the augmented reality component is Vuforia SDK\(^1\), which is a software development kit that uses the Computer Vision technology for image and 3D objects recognition. It is available for Android, iOS and the Unity3D game engine, being capable of maintaining sufficient performance with interactive frame rates over 20 Hz in most cases [4]. After creating an account on Vuforia site, we have registered the application and obtained a license key. It was used to integrate the application with Android Studio. Then we built the database for registering marker type images, important for the augmented reality (see Figure 3).

![Figure 3. Vuforia database from the Pocket World application.](image)

\(^1\) https://developer.vuforia.com
In the touristic_objectives database, each image has an augmentation level; level 5 for a certain image means that it can be successfully used as a marker in our application.

In Figure 4, we can see how this module works based on the Vuforia component. Due to the high number of TrackableResult objects being transmitted to the application with a high frequency (few dozen per second), we have used a specific method to ensure the existence and the persistence of the identified object. The method consists of counting the identified objects up to a given limit, if they are under the same name. It is also allowed to have a small deviation.

For instance, if we continuously identify trackers with the colosseum_x name and there is a new tracker under a different name, the latter will be ignored, because we are processing at high speeds and situations like this can occur, thus they will be treated as errors. The counting is restarted when the number of different trackers is higher than a given limit.

When the same object has been identified for a number of times higher than a predefined limit, the application accesses the LandmarkInfoActivity activity, which is essentially a page that extracts information from Wikipedia and displays it in real time (see Figure 5).

The correlation between the tracker names, the time-stamp from the database and the image is done based on some naming rules. Let’s follow the landmark with the Statue of Liberty name from the database correlated
to all the images registered in Vuforia as *statue_of_Liberty_x* (names that only include lowercase letters, words delimited by “_” and with the “_x” suffix, where “_x” is an integer number) and associated image from ListView will be found under the name *statueofliberty* (lowercase letters and delimiting removed).

![Image of Colosseum](image)

**Figure 5.** Page displaying information about the object identified as Colosseum.

From this point on, the user has the possibility to identify the landmarks located in the same city as the landmark registered in the LandmarkInfoActivity activity instance (pressing the Find closest landmarks button) or to erase it from the database permanently (pressing the Remove item button).

**Lookup a landmark**

Because we do not always have an image of a landmark or we are not near it to be able to scan and identify it, we can select landmarks from a provided list. This list can gather points of interest that are usually found
as landmarks, like statues or buildings, as well as POI’s that are not necessary landmarks, like parks (that cannot be identified only based on some trackers).

We have created a module with an easy function: to offer a list of all the landmarks stored in the database and an efficient search bar to be able to filter them by name (see Figure 6).

![Figure 6. Searching and selecting landmarks.](image)

**Finding nearby landmarks**

Most of the time, when we find ourselves in a new place or city, we are not always aware that we might also find some less known landmarks around (maybe due to the fact that we focus on the well-known attractions, we could lose the opportunity to visit other interesting landmarks). This is why it is very useful to have an application which lists the nearby attractions and the distance to them just by pressing a button (see Figure 7).
Figure 7. Displaying the landmarks that are nearby the user’s location.

3 Conclusion and Future Work

The AR applications are endless, as are those of virtual reality. Probably the AR will stay on mobile platforms, since everyone now has a smartphone. It is very accessible. Modern day technologies are permanently changing and this comes with numerous benefits. Apart from the easy information transmission, the focus of the educational software users is also the new information display methods.

*Pocket World AR* is an interactive and fun environment for learning that can help increase the interest and motivation of both children and adults. They can learn more in this way and it is easier for them to link places and facts. On a small scale, the application also facilitates learning basic information about foreign countries and places. Another important advantage of the application was helping users with memory exercises, through the use of augmented reality. Due to the fact that by simply scanning a landmark we can obtain information about it and we can also link it to other nearby landmarks by simply pressing a button, we add more consistency to the information from the Wikipedia page (images, events and basic descriptions). Basically, a trip to a new city can become the perfect time to learn information about the landmarks that define it, without too much effort and too many searches.
This application can become much more complex and it can expand its usage area. First, we intend to migrate the application from the local level to the network level, where any new landmark added will be stored on a server, verified and after an acceptance process, it will be integrated in the application instances on all mobile devices. Second, we want to integrate the application with Google Maps for the following reason: along with the list of the nearby landmarks, we want to add them to a map, together with route suggestions for how to reach them. Third, we can link the application to the official websites of all the landmarks in the database, to be able to offer users information like opening hours, visitation schedules and admittance prices.

Acknowledgments. We thank all the volunteers involved in the development of this application. This work is partially supported by POC-A1-A1.2.3-G-2015 program, as part of the PrivateSky project (P_40_371/13/01.09.2016) and by the README project “Interactive and Innovative application for evaluating the readability of texts in Romanian Language and for improving users’ writing styles”, contract no. 114/15.09.2017, MySMIS 2014 code 119286.

References


Roxana Pantea et al.


Roxana Pantea¹, Adrian Iftene¹, Daniela Gifu¹,²,³

¹Faculty of Computer Science, “Alexandru Ioan Cuza” University of Iasi, Romania
²Institute of Computer Science, Romanian Academy, Iasi branch
³Cognos Business Consulting S.R.L., Bucharest, Romania
E-mail: {roxana.pantea, adiftene, daniela.gifu}@info.uaic.ro
Smart Museums with Augmented Reality

Andrei Porfíreanu, Alexandru Ungurean, Andreea Dascălu, Adrian Iftene, Daniela Gifu

Abstract

Nowadays, the evolution of technology is crucial to humans for several reasons, most of them will require time and training. Augmented Reality (AR) occupies a top place in the list of the most used technologies as much in the gaming industry as well as for applications that are based on the educational software concept applied in the cultural heritage. Smart Museums project aims to attract people of all ages to visit museums more often. The proposed application uses the capabilities of AR technologies to transform a visit to a museum in an attractive experience full of memorable memories. Basically, we can have access to textual or audio information about an artwork or its creator, or we can have access to image galleries with works of the same author.

Keywords: Augmented reality, Museum, Usability testing.

1 Introduction

What brings Augmented Reality (AR) into our lives? The immediate answer will be that AR provides a unique experience to all [1]. For instance, we can enjoy diverse opportunities of spending leisure time (e.g. a visit to a museum). Today AR has an extraordinary potential for our lives, being a technology that complements the real world with text or graphics objects in real-time. Basically, AR provides solutions to issues of space limitation, exhibitions costs, etc. [2]. In this context, AR allows us to see an augmented environment, with information generated by a computer, usually across sensors connected to a desktop computer or through mobile devices like smart phones or tablets.
One of the aims of AR is to help us to understand better the physical objects from real environment like museums, historical objectives, restaurants, etc., providing additional information about them.

The development of Smart Museums application started in 2017 at Faculty of Computer Science of Iasi, as a group project, meant to encourage teamwork between students and also make them discover and use new technologies. Now we have reached the point where our project has a practical dimension and can be easily integrated in the experience of visiting a local museum.

How? Simply view the exhibit you are interested in through your phone or your tablet screen and instantly multiple pieces of information will pop up on the screen. The application user will be able to choose between a text description, audio files, browsing through an image gallery of related exhibits, thus finding out even more details of the author’s work. In a society where technology has been gaining ground in more and more areas, we believe our application is a great way to ensure a friendly and smart experience, regardless the visitor’s age or the contact he has had with similar applications until then.

The paper is structured as follow: Section 2 refers to the Smart Museums architecture and its options, and section 3 describes how to test the system usability using a questionnaire designed to give a quantitative score of it. Finally, the paper conclusions are given in section 4, mentioning some future directions for the development of this augmented learning.

2 Smart Museums Architecture

This section describes the architecture of the application based on Client-Server model (see Figure 1).
In the process of creating the application, we used Vuforia, Unity and a client-server protocol to achieve three of the main functionalities.

Android is a software platform developed by Google. Android is open source and is designed for use on mobile devices, smartphones, tablets, GPS, TV. It provides a rich framework with which to create different applications or games in Java programming language. The framework offers numerous bookstores, documentation, tutorials and an emulator that simulates a mobile device on its own development platform.

Vuforia SDK is an augmented reality software development kit that uses Computer Vision technology to recognize simple 3D images and objects. This image capture capability allows developers to accurately identify real world elements when viewed through a mobile device’s camera.

---

1 https://www.vuforia.com/
2 https://unity3d.com/
3 https://www.android.com/
4 https://www.sciencedaily.com/terms/computer_vision.htm
Unity is a gaming platform developed by Unity Technologies, which is mainly used to develop both three-dimensional and two-dimensional video games, as well as computer simulators, consoles, and mobile devices. It was first announced only for OS X at the Apple Developer World Conference in 2005. Since then, it has been extended to target 27 platforms. Six major unit releases were released.

2.1 Client – Server Model

How we can see in Figure 1, the Server component (written in C#) provides information to the Client according to the following types of requests: login (only the admin of the museum and the user of the application), get-museum (allow to the user of application to get only one museum at a time and to download it on his device), insert-museum (the administrator can insert new museums in the application), delete-museum (also at the administrator level, allow deletion of museums from database) and modify the database with exhibits (allow to administrator to change details for one exhibit).

The Client component was written also in C# and it was integrated into the Android application, in order to communicate with server and to get information. User interaction is through the graphical interface. Also, at Client side we have a Java application, for the administrator of museum who can manage the information about the museum (insert, update, delete).

The client is a compatible Android application on all mobile devices that have a camera and Android support. When we start the application, the first screen of the Android application offers the main options: Visit museum or Download museums (see Figure 2).

Figure 2. Main options of the Android application.
In order to use the option Visit Museum, the user must:

- Download the package for the museum on the mobile device. The package can be downloaded in advance at home or in other places or can be downloaded in the museum. Of course, we recommend doing this in advance, because in some cases it is possible to take a bit depending on the speed of the Internet or the availability of the data transfer.
- Have the location enabled so that the application allows it to log in and to use this option (see Figure 3);
- Be within the museum to allow the application to use the information about this museum.

Figure 3. Location Issue.

Once it enters the application, the camera can recognize the targets using an API from Vuforia. Targets are artworks, sculpture paintings or exhibits from the museum. When a target has been recognized, the application shows a number of buttons for doing various activities (see Figure 4 left).
At the top of the application, we have the 3 buttons to help the user to do different things:

- **Text** if the user wishes to read details about the opera or about the artist (see Figure 4 right);
- **Audio** if the user wants to hear the text from above option instead of reading it;
- **Gallery** if the user wants to see pictures of similar exhibits with the recognized one (see Figure 4 right).

At the center of the screen, the user sees the name of the exhibit that was scanned with the phone (see Figure 4 middle).

In the bottom of the application are two buttons. The left button opens a Quiz. This questions the user about the exhibit and tests his knowledge. The button on the right opens a panel of other works of art similar to the one scanned by the user.

### 3 System Usability

To test the system usability we use the System Usability Scale (SUS) [3]. It is a questionnaire designed to give a quantitative score of usability, based on a user’s subjective experience with a system. The scale is
independent of technology or interface types, so it allows any digital system, or iterations of a single system, to be compared. SUS spans only 10 Likert scale questions, and was described by Brooke (1996) as “quick and dirty” [3]. It does not offer an exhaustive evaluation, but “[...] a general indication of the overall level of usability of a system compared to its competitors or predecessors” with low costs, and minimal strain on participants [3]. SUS scores range from 0 to 100 (with 100 being the highest score).

A large empirical evaluation of the SUS (based on 2300 individual surveys from over 200 studies) found the scale to be reliable and accurate for measuring usability, and to be useful for iterative design processes. It further introduced a set of acceptability ranges and “adjective ratings” as a tool for interpreting SUS scores. Figure 5 displays the original definition of these ranges, as described in [4]. Both Brooke [3] and Bangor [4] warned against doing analysis on individual SUS questions, as they are very co-dependent. Furthermore, in [4] authors recommend that SUS scores should be regarded together with more specific observations and qualitative user statements.

![SUS Scores](image)

Figure 5. SUS Scores.

We have selected pieces of information in text format and photos that are relevant to the exhibits. For the audio format, we have used the same text. All resources were available in Romanian and in English.

The ten issues addressed to the users involved in the evaluation were:

1. I think that I would like to use this system frequently.
2. I found the system unnecessarily complex.
3. I thought the system was easy to use.
4. I think that I would need the support of a technical person to be able to use this system.
5. I found the various functions in this system were well integrated.
6. I thought there was too much inconsistency in this system.
7. I would imagine that most people would learn to use this system very quickly.
8. I found the system very cumbersome to use.
9. I felt very confident using the system.
10. I needed to learn a lot of things before I could get going with this system.

Measuring the usability of our application was done in two stages. First measurement was done after we had the first working prototype of the application outside of the museum. The System Usability Scale gave us a score of 76, which according to the SUS scores from Figure 5, means a good usability. The second measurement was done after the application integrated all the components specified in the requirements in the museum “Mihai Eminescu”. This time the score was 78, which was not far from the first result. However, the improved mark showed little signs of progress in the good direction. During our experiments, the museum curators showed interest in using new technology to promote art.

More testing needs to be done in order to assess the true usability of the application. A big bias that influenced the scores, pushing them higher than in a real scenario. It might have been the users who have done the testing part. The application was tested mainly by the members of our group, directly involved in the development process. Testing needs to be done exclusively on users not involved in the development steps, in a museum scenario where the application is supposed to be used in the first place.

4 Conclusion

The paper presents our work in Smart Museums project, which has the aim to attract more peoples (children and adults) in museums from Iasi. Augmented Reality can help the visitor to understand better an artwork or to obtain easier information about creator.

Today eLearning applications using AR are focused on subjects such as chemistry [5], geography [6], history [7], and astronomy [8]. Lessons and evaluation games are available on mobile devices (smart phones or tablets) and contain animations, movies, sounds, and so on.
Current and future work is focused on creation of games to verify the attention of visitors or to involve them in collaborative puzzling activities.

**Acknowledgments.** We thank all the volunteers involved in the development of this application. This work is partially supported by POC-A1-A1.2.3-G-2015 program, as part of the PrivateSky project (P_40_371/13/01.09.2016) and by the README project “Interactive and Innovative application for evaluating the readability of texts in Romanian Language and for improving users’ writing styles”, contract no. 114/15.09.2017, MySMIS 2014 code 119286.

**References**


Andrei Porfireanu¹, Adrian Iftene¹, Daniela Gifu¹,²,³

¹Faculty of Computer Science, “Alexandru Ioan Cuza” University of Iasi, Romania
² Institute of Computer Science, Romanian Academy, Iasi branch
³ Cognos Business Consulting S.R.L., Bucharest, Romania

E-mail: {andrei.porfireanu, adifene, daniela.gifu}@info.uaic.ro
Solar System Explorer

Diana Elena Gușă, Adrian Iftene, Daniela Gîfu

Abstract

Academic research through to engineering, business, design, entertainment, etc. shows that virtual reality (VR) simulations improve collaboration and knowledge construction. The application described in this paper has a target group that consists of users (e.g. students), who wish to learn and understand better the Solar System. Naturally, I called it Solar System Explorer and it makes use of the virtual reality and sensors of the mobile phone. It provides in interactive way information about planets, videos, animations, statistics, and social networking posts for a quick learning. It also offers the opportunity to evaluate the knowledge.

Keywords: Virtual Reality, Solar System, Android.

1 Introduction

Virtual Reality (VR) is a state-of-the-art technology and it is now enjoying its incredible development. Through virtual reality we can create a reality that does not exist, being separate from the physical one. Virtual reality can be seen as an imaginative reality, the virtual term assuming the absence of visual, palpable boundaries, everything being related to our imagination. In technical terms, virtual reality describes a three-dimensional environment generated by a computer; virtual reality describes a three-dimensional environment generated by a computer, an environment for translating the app. Once transposed into virtual world, we can interact with objects, virtual actors, and other real people (arrived there in the same way we do) to work together to accomplish tasks.

Virtual reality can be applied in various fields, such as architecture (modeling, simulation and visualization), sports, medicine (experiments and simulations), simulators (pilots, astronauts, drivers), art, entertainment (games and movies) [2], [4] and [5]. We can reach new discoveries in
areas that have a direct impact on our everyday lives [1]. This technology is more and more common and we can expect more and more innovations from this sphere in the coming period [3].

Smart phones, tablets and other mobile devices have been provided in recent years with sensors that enable us to simulate the virtual world. The most common sensors are the magnetometer, the accelerometer and the gyroscope. These are very important for generating virtual reality, because they allow us to simulate the movement in space that can be transferred to the virtual world. Solar System Explorer is an application that uses the mobile phone’s motion sensors to create a virtual environment that offers the possibility of knowing the Solar System. The application is an educational but also entertaining way of interacting with the user. It can be used especially to help students improve their knowledge of the Solar System and learn about each and every planet. Obviously, the application can be used by users of any age who want to improve their knowledge of our solar system. Users can also evaluate their knowledge gained through the tests provided, with the choice of several degrees of difficulty. The application allows you to browse the list of planets and select the desired one, search them through a search engine or voice commands, or directly access by clicking the screen. For each object in the Solar System, information, statistics, video documentation taken from YouTube, 3D animations, and current Twitter posts are provided.

The paper is structured as follow: Section 2 presents a short overview of VR systems in order to clarify the importance of them in order to encourage students to use these methods that individualize learning, while Section 3 refers to the Solar System Explorer 3D architecture and its functionalities. Finally, the paper conclusions are given in section 4, mentioning some future directions for the development of this virtual learning.

2 Similar Applications

In order to understand the extraordinary potential of applications based on VR, we need to understand its significant potential to transform the way we learn, helping us understand complex topics to facilitating virtual lessons. We describe some applications that use VR to explain why software based on it was needed?
2.1 Star Chart
The Star Chart\textsuperscript{1} application is designed for Android and allows stars and planets to be viewed on the mobile device as well as viewing within a planetarium. It offers an animated and illustrated visual experience. The application allows you to enlarge the images for observing the details. By simply clicking on a particular object, you can get information about it. The application needs the location of the device, and users have to enter the latitude and longitude, or they can give them access to the GPS. Thus, when the device moves, it changes and the map of space gives the impression of a real exploration of the sky.

2.2 Solar System Explorer 3D
The Solar System Explorer 3D\textsuperscript{2} application is also created for Android and offers the possibility of exploring the Solar System. There is a list of planets and, depending on the object chosen on the screen, its image is displayed along with information and options for it. The Solar System map can also be traced by dragging or rotating the image with your fingers on the screen. The app also offers the Flight option to explore the space with a left-right, up-down, and a rotating button.

2.3 Sky Portal
The SkyPortal\textsuperscript{3} application is for both Android and iOS. It is one of the best star simulators. We can look what's in the sky above us or we can move the device to explore the space. There is also an option to search for objects and get all the information about them. The application allows you to choose different dates and times than the current one to see what changes over a certain period of time. The app also has Night mode for eye protection. Another feature provided by this application is the use of the compass to position where we want on the map.

2.4 Sites in VR

\textsuperscript{1} https://play.google.com/store/apps/details?id=com.escapistgames.starchart
\textsuperscript{2} https://www.amazon.com/Burlock-IT-Pty-Ltd-Explorer/dp/B00PE9ISMY
\textsuperscript{3} https://play.google.com/store/apps/details?id=com.celestron.skyportal
The Sites in VR\textsuperscript{4} app is for both Android and IOS. It uses virtual reality and gives the user the chance to translate into new places, different cities and unique landscapes. The application does not particularly focus on space exploration, but there is also the option to see museums, palaces, castles, or other tourist attractions in different cities around the world. The desired location can be chosen from several countries or cities provided by the application, but also from a range of tourist attractions that can be found in some parts of the world. Once the desired landscape has been selected, the user can explore the surroundings by moving the device to the desired area or using Google Cardboard virtual reality glasses.

3 Proposed Solution

Solar System Explorer combines the elements of the Android platform with frameworks and APIs for various services such as Virtual Reality, YouTube, Twitter, making it an easy-to-use application. To make the solar system model, the Autodesk Maya which has a 3D objects modeling software was used. The application contains two parts of virtual reality: one that was made using mobile phone sensors and the other one made with Google VR. After the application is launched, the sensors start to record motion. When changes occur they are transmitted and the image is drawn to the new coordinates. The virtual reality part created with Google VR is faster in motion detection and it allows the use of Cardboard glasses.

The application also provides a way to connect to the latest news and events related to the solar system by presenting posts from Twitter. Thus, the user is connected to an important source of information, already centralized and filtered. The most interactive part of the application is the test provided, which allows the evaluation of the knowledge acquired. There are three levels of difficulty: easy, medium and hard. The obtained score helps to track the progress of the study. The application could be used as an assessment system for astronomy classes.

The user can search a planet using the first menu button or the item that can be chosen from the list provided. The second menu button is the

voice command function that can be used to search planets. The user has to say the name of the planet and he will receive more details about it. After selecting a planet, the menu changes. It provides features specific to the objects. Here are some documentaries that have been downloaded using a YouTube API. The application offers 3D animations for a clear view of planets and the space that surround them. Also, there are real and important text information retrieved from the NASA website and the links connect directly to a secure web page for detailed articles.

3.1 Application Architecture

Main module
When the app starts, the main activity that contains the menu buttons, along with the two parts of virtual reality is launched. The first is created using sensors and image redraw when changes are made from them, and the second is created using the Google VR SDK, also offering the possibility to use Cardboard glasses. The application can be used in Full-Screen mode, and the orientation is Landscape, except the activity that makes posts on Twitter where Portrait is allowed. Figure 1 in left shows how the application looks after it is released, with all the details above.

Figure 1. Solar System Explorer Home Page (left) and launch the specific activity for the Sun after it was selected from the main image (right).

VR using device sensors
Also, once the app is turned on, the phone sensors are also switched on to receive the move. An axle system is used, where a point has 3 coordinates: $x$, $y$, $z$. Thus, every motion detected on the $Oy$ axis will be sent to redraw the background image to the new coordinates, giving it the impression of moving it. The main screen image will be the one used by the sensors, being the image of the Solar System drawn with the Maya
program. Users will be able to zoom in and zoom out. They will also be able to access each planet by simply touching it on the screen.

Once a planet is clicked, another activity will start, and another menu will appear on the screen with the image specific to each object. As can be seen in Figure 1 right, the Sun was selected, and the main menu and image changed. Thus, when the user is interested in finding out more about a particular planet, he can always click on it and document it. The specific menu for Solar System objects will be described in the next section.

Search in the list of objects
By exploring the left-hand menu, the first button represents the Search function of the planets. Figure 2 left shows that the user is provided with a list of solar system objects and a border (at the top of the screen) where he can enter text, representing the name of the planet being searched for. Thus, a planet can be chosen by scrolling through the list presented, or it can be searched more quickly by entering a part of the name of the planet we are looking for in the Search bar.

Voice commands
The second button in the main menu of Solar System Explorer is the ability to search for an object in the Solar System with voice commands. Thus, at the touch of a button, the user will be able to see a check box confirming that the device’s microphone is recording his commands. This can be seen in Figure 2 right.

When it comes to obtaining information about a particular planet, its English name should be spoken and its specific activity will be opened. This search method gives the user an easier way to find what he wants.
When the name of a planet is spoken, the device’s microphone may not receive the command due to its distance or due to surrounding noise. In this case, a message will appear in which the user is asked to repeat the command. So he has to make sure the name is clearly spoken, close to the microphone, and that there are no other noises.

**Twitter posts**
The third menu button allows the user to interact with the Twitter social network. Through the Solar System Explorer, the user is given access to real-time, solar system tweets on accounts like NASA, without having to create a Twitter account or connect to the personal.

At the same time, posts are already filtered and centralized, so you can access the tweets about each planet or use the *All* option to browse all existing Solar System posts. Figure 3 left shows that the list of planets is available, but also the option to show all posts in descending order.

![Figure 3. List of planets for getting posts on Twitter (left) and Post static on Twitter (right).](image)

A list of posts about the latest solar system news is available. Information can be updated in real time. Users can give a Refresh page and new tweets will be loaded into the app if posted. If a user clicks on a post, he will be redirected to the Twitter Web site where he can log in, and view the posts in his own account. For the variation where all the tweets about the Solar System are presented, a button can be seen in the bottom right - down corner, which will redirect the user to a post statistics.

By accessing it, we will notice, as in Figure 3 right, how often a planet has been posted compared to the other. On the right, we have the name of the planets and what color is representative for each, and in the
center of the screen is the Pie Graph. In the upper left corner there are two options that allow you to customize the chart by using two Ratio Bars.

**Quiz**

The fourth button on the left side of the menu is the most interactive part of Solar System Explorer. As can be seen in Figure 4 left, the functionality of the button is to redirect the user to a test to verify his / her knowledge of the Solar System gained through the application. Thus, 10 questions will be made to highlight the most important aspects to be known about the planets. Tests offer 3 levels of difficulty: *Easy*, *Medium* and *Hard*, so the same set of questions can be used to assess your knowledge by the degree of difficulty you want. At the same time, depending on the level of difficulty chosen, the question-answer model differs.

![Solar System Quiz](image)

Figure 4. Solar System Quiz for knowledge evaluation (left) and Easy level - single-choice model (right).

For the *Easy* level, the user is provided with a variety of single-choice responses, and he or she has to select a single option. This form of test has a low difficulty level, giving him 3 possible answers to choose from, as can be seen in Figure 4 right. If checked the correct answer, a corresponding message will be displayed on the screen. If the answer is wrong a message will be, again, displayed on the screen indicating this, along with the correct one.

*Medium* difficulty level is a medium difficulty assessment mode where the user needs to find the answer without giving him a list of possibilities. The text entered can be written in uppercase or lowercase, or a combination of these. If it is the correct answer, the user will be notified
by a message. The wrong answer is notified by an appropriate message along with the correct variant.

The **Hard** level is the most difficult level founded on the same question model used for the Medium level. But additionally a timer is added. Thus, not only is there no longer a list of possible answers, but the user has to answer even before the allocated time expires, otherwise he will lose the question and get 0 points.

## 4 Conclusion

The Solar System Explorer offers various ways the user can study the Solar System. Two virtual reality methods are provided that make interaction with the application more interesting, due to the possibility of exploring the space by receiving the movement of the mobile device. At the same time, accessing the planets and deepening their knowledge is done in an easy and pleasant way by simply clicking on them, using the Search function, or using the voice command. The application also provides a way to connect to the latest news and events related to the Solar System by presenting posts from a popular social network, Twitter. The user does not have to connect to an account, or to search for specific tweets, the information being centralized and filtered already. It is possible to obtain specific posts on a particular planet, or all those related to the Solar System. For the second variant there is also a statistic of the tweets, and so we see how often things were posted about a planet compared to the others.

The most interactive part of the application is the test made available to evaluate the knowledge gained during the use of the application. There are three levels of difficulty: easy, medium and hard to determine how the questions will be asked and given answers. For Easy, one-choice response variants are used, Medium uses *EditText* in which the response is to be written, and for Hard is used the same model as the Medium level, the difficulty being added by the presence of the timer. The score obtained helps to see the progress made after the study.

In the future, the app could also be developed for other platforms like IOS. Star, constellations, satellites, or other elements of the Solar System could also be added for a more sophisticated knowledge base. The application could also be used as an assessment system for astronomy
classes, eventually adding the possibility of creating an account for both students and teachers, and the latter having access to student results and adding questions similar to [6].

**Acknowledgments.** We thank all the volunteers involved in the development of this application. This work is partially supported by POC-A1-A1.2.3-G-2015 program, as part of the PrivateSky project (P_40_371/13/01.09.2016) and by the README project “Interactive and Innovative application for evaluating the readability of texts in Romanian Language and for improving users’ writing styles”, contract no. 114/15.09.2017, MySMIS 2014 code 119286.

**References**


Diana Elena Gușă¹, Adrian Iftene¹, Daniela Gîfu¹,²,³

¹Faculty of Computer Science, “Alexandru Ioan Cuza” University of Iasi, Romania
² Institute of Computer Science, Romanian Academy, Iasi branch
³ Cognos Business Consulting S.R.L., Bucharest, Romania

Faculty of Computer Science, “Alexandru Ioan Cuza” University of Iasi, Romania
E-mail: {diana.gusa, adiftene, daniela.gifu}@info.uaic.ro
Research centered on investigating the ethical attitudes towards information technology

Liliana Mâţă, Ioana Boghian

Abstract

The main purpose of this research is to explore current studies aimed at measuring the ethical use of information technology. Depending on the research methodology used, three categories of studies are quantified: quantitative research based on the construction of different questionnaires to measure attitudes or perceptions towards ethical use of information technology, qualitative research centered on the creation of ethical scenarios or the use of interview and mixed research that integrates both qualitative exploration methods and quantitative investigation methods. The results of the study highlight the need for new research based on mixed methodologies to investigate ethical attitudes towards information technology.

Keywords: ethical attitudes, information technology, mixed research, qualitative research, quantitative research

1 Introduction

The massive integration of information technology (IT) in universities brings new implications for the ethical use of these resources by students, teachers and administrators. All the education system has registered an explosive growth in internet use and computer technology. There is a high connection between students and different technologies due to unlimited access to information, which could also lead to a number of negative effects on the educational process along with their benefits. The lack of scientific integrity in educational environments as regards the
use of technology is a problem that should be taken into account. Ethical issues related to the use of information technology require a professional and sensitive approach to making students aware of them. There are misconceptions about copyright laws and ethical concerns about the use of information technology [1] by students and sometimes even by teachers. If these ethical issues are ignored, they can have a negative impact on learning. The general purpose of information ethics is to integrate technological and human values in a way that advances and protects human values rather than harm them.

The increasing use of new technologies in the educational process forces students to get informed about computational ethics and related social and legal issues [2] so that technology rewards can be accessed by everyone. Teachers in higher education have to turn to information technology to prepare lectures, download materials from the Internet, publish works online, send emails, and provide students with different web links related to course content [3]. Under these circumstances, Johnson and Simpson [4] highlight that educators need to be aware of the legal and illegal use of information technology to provide ethical models for students.

It is important to draw the attention of the authorities to the training of university professors and students in order to raise awareness of the appropriate and efficient use of information technology. As Rolstad [5] also mentions, it is necessary to include the ethics of information technology in curricula at universities, as new digital tools are widely used in the educational environment. Sargolzaei and Nikbakht [6] proposed a set of strategies for preventing ethical abuse in the field of information technology. Introducing a course that includes ethical issues related to the information technology for license, masters and doctoral students becomes essential to provide appropriate responses to the ethical challenges of IT. Masrom et al. [7] recommends the organization of workshops in higher education in the field of computerized ethics for all technology users and the continuous achievement of new research to monitor the awareness of the ethical use of technologies in educational process. Therefore, the purpose of this paper is to analyze the studies centered on investigating the ethical use of information technology.
Research centered on investigating the ethical attitudes towards information technology

2 Investigating the attitudes towards the ethical use of information technology

Inappropriate use of technology in education is one of the recent themes that attract the attention of researchers in the field of computer science [8]. Although new technologies are increasingly integrated into higher education, research aimed at investigating the ethical attitudes regarding the ethical use of information technology is rather low. There have been selected articles published in international journals that include research on the ethical use of information technology over the last 15 years.

Most studies are predominantly quantitative as a result of the use of questionnaire-based surveys. Siegfried [9] applied a questionnaire to assess the attitudes of students regarding software piracy. The results showed that there has been few if any changes in student opinions regarding the unauthorized duplication of copyrighted materials. Lorents et al. [10] measured the students’ ethical attitudes towards computer use. The main research instrument was the questionnaire, which was required to rate the behaviour described in several scenarios on a 7-point Likert scale. The results show that the median rank for all situations ranges between the unethical and the very unethical. Masrom et al. [11] investigated the ethical awareness of computer use among undergraduate students from public universities. Their research tool was the questionnaire. The results indicate that male students and senior undergraduate students’ beliefs were clearer about ethical perceptions on computer use ethics. Pupovac et al. [12] used the Attitude toward Plagiarism questionnaire among pharmacy and medical biochemistry students. Results revealed moderate positive attitude and subjective norms toward plagiarism and moderate to high negative attitude. Chiang and Lee [13] identified, based on a questionnaire, the intentions toward observing ethics in the area of digital rights. There were developed the following aspects: freedom of expression, freedom of association, equal access to information, confidentiality, security, and protection of intellectual property while using computers. Özer et al. [14] adopted the version of the Cyberethics Questionnaire developed by Yamano (2004) to measure computer teachers’ attitudes, awareness, and teaching practices regarding computer ethics. Hashim and Hassan [15] developed a questionnaire to investigate the cyber ethics aspects in using Facebook for Social
Networking. Jamil et al. [16, 17] explored the attitudes of students and teachers from different public and private sector universities towards ethical use of information technology. Sargolzaei and Nikbakht [6] designed a questionnaire to evaluate the ethical issues of information technology. The main ethical and social issues indices in the field of information technology are the following: anti-religious propaganda in the cyberspace, online theft, copyright violation, addiction to social networks, rapid formation and spread of rumors and false news, interest in violent computer games, verbal attacks, online gambling, negative behavioral and personality effects, forging digital documents, ethical abuse of the camera, violation of privacy, hacking.

Acilar and Aydemir [18] conducted a study using a survey method to collect data among freshman business administration students at a public university in Turkey, with the aim to investigate whether age, gender and duration of computer usage in a week have a significant impact on freshman students’ ethical judgments regarding computer and Internet usage. They found that age is an important factor: the more mature students are, the better their understanding of ethics; female students are more sensitive than male students regarding the unethical use of computers; more computer experienced students show less ethical attitudes toward computer usage; differences between female and male students were investigated more closely in a subsequent study of Acilar and Yörük [19]. Surveys of student attitudes about unethical uses of information technology applied at two different institutions and, based on the data collected, Etter et al. [20] found that students at a private church-affiliated college rated cheating behaviors as more offensive than their counterparts at a regional campus of a major research university. In their study, Gecer and Tosun [21] adapted the Internet ethical attitude scale for students from secondary school to university and compared the ethical attitudes of participant students according to their genders and departments; the data was collected through the Internet Ethical Attitude Scale (IEAS) and a personal information form designed by researchers. The results showed that female students are more conscientious than male students in the issue of Internet ethical attitudes and attitudes related to “homework plagiarism” are higher in female students than male students.
Hosny and Fatima [22] investigated the attitude towards cheating and plagiarism among female students in the College of Computer and Information Sciences (CCIS) at King Saud University, Riyadh, Saudi Arabia and found that both cheating and plagiarism are common practices among the students, despite the fact that most students believe that they are unethical and against religious values. In their study, the researchers Leonard and Cronan [23] attempted to identify factors influencing an individual’s attitude toward ethical behavior in information systems environment; discriminate analysis was used to process the environmental factor influences assessed by a sample of university students: societal, belief system, personal, professional, legal, and business), moral obligation, consequences of the action, and gender. The findings show that the many factors influencing attitude toward ethical decisions also depend upon the type of ethical issue involved and may shift over time; also, attitude towards unethical use of IT depends on sex. By examining current views of IT property as measured by a series of six current IT related property ethical issues, Peslak’s study [24] is surveys a cross-section of students, faculty, and professionals and analyzes recognition of an ethical issue for each individual topic: the findings confirm that all the studied information technology property issues are generally recognized as important ethical topics and significant differences were found by age and gender for some but not all property issues.

The study by Quah et al. [25] discusses the ethical orientations of students (ethical idealism, ethical relativism and Machiavellianism) towards their attitude to plagiarize and also the moderation effect of religious orientation on the relationship of the independent variables towards students’ attitude toward plagiarism. The results showed that ethical relativism and Machiavellianism had a positive relationship with students’ attitude towards plagiarism and religious orientation was found to have no moderating effect on the relationship between ethical idealism, ethical relativism and Machiavellianism and students’ attitude towards plagiarism. The study by Voutsa et al. [26] presents the results of a survey conducted based on the questionnaire published by ETHICOMP in 1998 applied to the students of the department of Informatics at the Alexander Technological Institute of Thessaloniki. The findings reveal students’ lack of deep awareness in deontological and ethical issues. Yi-Chih and Wei-
Li [27] examined the correlation between risk perception, knowledge, social influence, self-efficacy, and cyber bullying behavior from the perspective of the attitude-social influence-efficacy model among adolescents who have had cyber bullying behavior or have witnessed their peers’ cyber bullying behavior; the results showed that attitude towards cyber bullying affected cyber bullying intention, and that intention also influenced cyber bullying behavior. Social influence also had an impact on cyber bullying intention and cyber bullying behavior.

There are also qualitative research studies based on creating ethical dilemmas or interviewing. Teston [28] designed a survey tool, which consisted of various stories dilemmas to measure ethical values. Statistical results regarding the attitudes to software piracy indicated that 48% of students consider it legal. In their research, Liu and Chen [29] explored cross-cultural differences between American and Chinese business students with respect to their attitudes towards information ethics and investigated the association between their ethical evaluations of the target act and their morality judgments of the target actor in the questionable information-handling issues. Based on case-studies and computer-ethics scenarios, they found that Chinese students tended to be more ethical in the questionable privacy and access issues, whereas their American counterparts were more likely to be ethical in the questionable property issue. Also, female students, regardless of their culture backgrounds, were more ethical than male students; the students’ ethical evaluations of the target act were significantly related to their morality judgments of the target actor. Masrom et al. [30] address computer-related scenarios that can be used to examine the computer ethics and identify the ethical issues involved; they also review several measures of computer ethics in different setting and perceptions of various dimensions of ethical behaviour in IT that are related to the circumstances of the ethical scenario.

Unlike quantitative research, there are very few studies that are based on mixed methodologies. Verecio [31] used questionnaires, interviews, and observations to assess the level of awareness in computer ethics. Researchers Alakurt et al. [32] used questionnaires and interviews to investigate ICT-related ethical problems. The computer-related scenarios were designed based on Mason’s [33] four ethical issues: privacy,
Research centered on investigating the ethical attitudes towards information technology accuracy, property and accessibility. By using a research methodology based on focus group and questionnaire, Freestone and Mitchell [34] addressed Internet related ethics and identified 24 aberrant behaviours of young consumers, grouped into five factors: illegal, questionable activities, hacking related, human Internet trade and downloading; those perceived as least wrong were; downloading movie and music files from the Internet for free. Karim et al. [35] used the case-study approach and surveys to highlight the connection between students’ unethical Internet use and the big five personality variables; personality traits such as agreeableness, conscientiousness and emotional stability were found to be negatively correlated with student unethical Internet behavior.

The main categories of research identified are qualitative, quantitative and mixed research (Table 1).

Table 1. Types of research depending on research methodology

<table>
<thead>
<tr>
<th>Categories of research</th>
<th>Research methods, authors</th>
</tr>
</thead>
<tbody>
<tr>
<td>Quantitative research</td>
<td>questionnaire (Chiang and Lee, 2011; Hashim and Hassan, 2013; Jamil et al., 2013, 2014; Lorents et al., 2006; Masrom et al., 2009; Özer et al., 2011; Pupovac et al., 2010; Sargolzaei and Nikbakht, 2017; Siegfried, 2004; Acilar and Aydemir, 2009; Acilar and Yörük, 2010; Etter, Cramer and Finn, 2007; Gecer and Tosun, 2015; Hosny and Fatima, 2014; Leonard and Cronan, 2005; Peslak, 2007; Quah, Stewart and Lee, 2012; Voutsa et al., 2006; Yi-Chih and Wei-Li, 2018)</td>
</tr>
<tr>
<td>Qualitative research</td>
<td>ethical dilemmas (Teston, 2008); case-study/computer-ethics scenario; Liu and Chen, 2012; Masrom et al., 2010)</td>
</tr>
<tr>
<td>Mixed research</td>
<td>questionnaire, interviews, and observations (Verecio, 2016); case-study approach and survey (Karim et al., 2009); questionnaire and interview (Alakurt et al., 2012); focus group and questionnaire (Freestone and Mitchell, 2004)</td>
</tr>
</tbody>
</table>
3 Conclusion

One of our conclusions is that quantitative studies predominate, at a significant difference in number compared to qualitative and mixed types of research. The qualitative studies are highly important because they highlight the types of unethical issues occurring in the IT use, the attitudes towards unethical use of IT, as well as the factors influencing such attitudes. For example, findings of studies that attitude towards unethical use of IT depends on sex and may shift over time support the elaboration of training programs that need to focus on the different influencers for males and females and, also, that organizations must periodically reassess their employees’ ethical climate and adjust their ethics’ programs as attitude influencers change. Another conclusion is that more qualitative research is needed to highlight possible measures for intervention. Also, mixed research may connect results from both qualitative and quantitative studies, to propose solutions and course of action.

Acknowledgments. „This work was supported by a grant of Ministry of Research and Innovation, CNCS - UEFISCDI, project number PN-III-P1-1.1-TE-2016-0773, within PNCDI III”.

References


Research centered on investigating the ethical attitudes towards information technology


Research centered on investigating the ethical attitudes towards information technology


Liliana Măţă¹, Ioana Boghian²

¹„Vasile Alecsandri” University of Bacău, Romania
E-mail: liliana.mata@ub.ro

²„Vasile Alecsandri” University of Bacău, Romania
E-mail: rahela_bac@yahoo.co.uk
Achievements – a way to engage students in educational process

Olesea Caftanatov

Abstract

In this work, we explore the role of achievements in the educational software. Our aim is to analyze the best requirements for creating and implementing achievements in educational software, in order to ensure that children will not be distracted from the learning process but still be engaged in it. Additionally, we involved a group of children in designing collectible items that will be implementing in our educational software GeoMe.

Keywords: achievements, gamification, rewarding systems, badges.

1 Introduction

During recent years, the practice of adding game elements ("gamification") to non-game services has gained a relatively large amount of attention, especially in educational system, because it is obvious that when students are engaged, they learn. We also understand that games, in any form, increase motivation through engagement. Moreover, one of the key features of a game is to keep players engaged and wanting to play more, otherwise the game will not be considered a success and after a short play time it will be deleted and forgotten.

Nevertheless, regarding engagement nowhere else is more important than in education. Nothing demonstrates better than striking high school dropout rates because of a general lack of student’s motivation. According [1] nationwide, an estimated 1.3 million students dropped out from the class of 2010 without earning a diploma. As the gamer generation moves into the mainstream workforce, including myself, we are willing and eager
to apply some game’s culture, for example achievements, into learning process.

In this paper, our aim is to analyze the requirements for designing achievements for our educational software GeoMe. GeoMe stand for “geometry for me”. It is software designed to help pupils in learning geometry by personalizing their learning paths. Pupils are unique, and each one of them likes to learn in their own style. GeoMe application focuses mainly on four styles: reading, listening, visualizing and practicing. Our main goal is to generate the content based on user’s preferences. If the user loves to listen than more content with audio style will be recommended, on the other hand, if the user likes to read than the next lessons will be in text format, and so on. GeoMe has six categories; each contains learning material regarding subjects such as point and line, angle, triangle, square, circle and other shapes. Our GeoMe application is still in process of development, but below (see Figure 1) we present few screenshots from early versions of our app.

Figure 1. a) The list of categories that can be scrolled down; b) The layout with video lessons regarding angles; c) example of bite size chunks lesson (by the way, avatar can be clicked to start the audio record of the below text); d) one of quizzes’ type regarding shape’s identification.
In order to make children’s learning process more enjoyable we designed tutor’s avatar [2]. Moreover, we intend to add rewarding systems. For this purpose, we involved a group of children in creating collectible items as rewards for unlocking achievements.

2 Short history regarding achievements

The idea for game achievements can be traced back to early 80’s, with Activision’s Patches for high scores. According to [3] every Activision game gave players the option of taking a photo of their high score and sending them to the publisher to receive a patch commemorating their accomplishments. In early 1980s, achievements were very limited and quite often not a focal part of tempting the players to return. Nowadays, games have certainly taken achievements to a completely new level.

The “achievements” terminology was not introduced until 2005 when Microsoft introduced the “Gamerscore” system for the Xbox 360 platform. Achievements are a bonus feature included in games that unlock when the player completes a certain task and sometimes it’s purpose are solely for showing off player’s gaming skills. Otherwise stated, meeting the fulfillment conditions encoded by the game’s developer is referred to as unlocking achievements.

**Definition 1 [4].** An achievement in a video game is a reward or recognition earned by players for an in-game accomplishment.

According to [5] achievements are the building blocks that enable someone to construct a sense of them as a success. There are various types of achievements based on what players are engaged. Bartle [6] proposes one of the simplest taxonomies (see Figure 2).
Consequently, game developers target a specific type of player and design for them a set of achievements. For better understanding, we will bring some example of achievements from a Chinese MMORPG Revelation [7]. For *killers*, there is such type of achievements as making triple kills without being killed or participate for 500 times in an immortal battleground, for *socializers* one of the achievements consists in accumulating 3k interaction points with each type of class. By doing some main line quests *explorers* can get achievements. Regarding *achievers*, there are designed special types of events like on Halloween they need to accomplish specific task to get achievements, or collect different types of mounts, wings, styles and so on. As we can see, achievements range from simple tasks the player would do anyway, such as completing a level or defeating a dungeon boss, to more complicated challengers like finding X hidden chapters that respawn randomly and complete a book, and so on.

Anyway, we need to understand that there are specific risks that achievements can bring. It is not sufficiently to create an achievement in learning applications and consider that users will be engaged. Categorically, no! If the requirements for getting achievements are too easy, then players will see them as trivial. Moreover, with too many trivial achievements the users will fell unrewarded. However, if we make mission too hard, then users will become frustrated. Another risk can be, when one of the achievements is impossible to get, then users may lose
their interest and motivation to accomplish requirements for getting other achievements.

Nevertheless, the strength of achievements should never be underestimated. There are many players obsessed with accomplishing 100% achievements and trying to get in the top 10 players on leaderboards. One of features regarding ratings that I like in Revelation game is that they offer the top rating between your friends lists. In such a way, it motivates to get better score, better skills then your friends do. Your friends are people that you often interact, so usually you care more about their opinion regarding you, then some stranger’s opinion about you. Additionally, when you see that your friends’ achievements are better than yours are, and you feel like you are not worse than they are, you are being engaged to demonstrate it by improving your performance.

3 Achievements requirements
According to [8-12] in order to design a better assignment that will produce some achievements there are many requirements, some of them are:

- **Attractive achievements work better.** In one of our researches [13] we observed that attractive interface works better. We understand that not only interface but also other attractive things work too. So making an attractive achievements is must do, because users like to display attractive rewards. According to my personal experience in many types of games, I can say with certainty that people like to differentiate themselves from others by wearing different types of titles, styles, trophies, badges with beautiful designed icons. So, one of the most important requires regarding achievements it to make them more attractive.

- **Combine easy successes with hard challenges.** Another good practice is to combine easy to get achievements with the hard one. According to [14] persons with „high in achievement” describe their failure as lack of effort, on the other hand, individuals with „low in achievement” describe their failure as lack of ability. In order to not demotivate users we need to design achievements with different grade of difficulties. By making easy
achievements to accomplish, users will not lose momentum. However, it is important to not develop too many of them. For users that love to collect achievements it is better to create achievements that are harder to challenge, that require more time and effort to achieve. Nevertheless, know the limits, by creating to hard requirements, users will lose they motivation to accomplish achievements.

- **Score achievements proportionality.** Achievements points should be proportional to the amount of time or skill required to earn that achievement.

- **List of the available achievements.** Users need to know which ones are available to them even if they are not unlocked yet. Those that they achieved should be colorful, but those unlocked it is better to keep with grayed-out until their display has been earned.

- **Think like an achievement hunter.** Some users will attempt to earn every achievement that a created. Therefore, it is better to provide achievements that target this type of players. It is important to avoid creating achievements that rely too much on elements beyond user’s control, like, too high random rate. Moreover, by adding some hidden achievements will make them surprised and happy.

- **Add special rewards in achievements.** Usually users try to get achievements because of their special rewards. Some of rewards can be mounts, important items for character’s development, experience, money, skills and of course titles.

- **Incremental achievements.** Achievements can be designed as standard or incremental. Incremental achievements involves users progress overtime, for beginner till the highest level that application can offer. Therefore, by keeping track of users progress can be a good choice to design achievements.

4 **Collectible achievements**

Some users respond to opportunities of winning, earning and collecting awards that can be displayed to other community members. In order to develop collectible achievements for our educational application we organized a workshop with participation of pupils from the 2nd grade. At
Achievements – a way to engage students in educational process

this event, we asked children to draw imaginary characters, objects but keeping in mind that all their works should contain some geometrical forms. Afterword’s we selected their drawings and asked illustrator to convert them into digital format, so that we can use them as items for collectible achievements. Below there are presented several examples (see Figure 3).

![Figure 3. Kids drawing & digital format](image)

While looking at kids drawings, our illustrator got some insights and draw a few items too (see a few examples in Figure 4).

![Figure 4. Examples of items for collectible achievements. Illustrator: Doina Seiciuc](image)

By using kids’ drawings, we opt to collect some funny, crazy ideas that kids would like. Moreover, we try to make them to look great, because attractive awards work better and users would be proud to show them off to fellow community members.

5 Example of achievement implementation

As we mention previously, by adding achievement in learning process pupils can be more engaged. For instance: there are pupils that learn
faster, others slower. For the slower category we can design the following achievement. Let us say we have a quiz with five questions, for each question user has 30 sec. time to give an answer. If the user give all correct answers on 29 second, than he or she will get the achievement: “Slower than a turtle”. Regarding this achievement, user will get the turtle’ collective item, see figure 3 and green title: “Slowpoke”. The effect of title are: when users wear this title they can get additionally +5 more seconds on answering quizzes. Thus, next time when doing a quiz, user will have 35 sec. time to answer a question.

6. Final thoughts
Our main project is developing an education application with a personalized content and design. In order to attract and keep user’s attention by creating delightful experience that affects users in the virtual and real world we intend to add one of game’s features such as achievements. For this purpose, we analyzed the criteria for designing better achievements, for creating enjoyable assignments that will help pupils to motivate and engage in their learning process. Moreover, in order to create attractive collectible achievements that children will like, we involved a group of pupils in designing achievements. Some of their works a presented in this paper.

References
Achievements – a way to engage students in educational process


Olesea Caftanatov¹

¹Affiliation/Institution: Vladimir Andrunachievici Institute of Mathematics and Computer Science, Chisinau, Moldova
E-mail: olesea.caftanatov@math.md

329
AI Based Multilayered Approach for Management of Mass Casualty Situations

Constantin Gaindric, Svetlana Cojocaru, Stefan Pickl, Iulian Secrieru, Olga Popcova

Abstract

Triage of casualties is the key element in the management of mass casualty situations. In addition, casualty evacuation without an effective coordination may lead to hospitals overcrowding and, therefore, to an increased number of deaths. So, guidance for rapid transportation (according to triage priorities, available ambulances and human resources and destination hospitals capabilities) is needed as well. We propose to enhance the efficacy of crisis management response in large-scale disasters by designing a decision support framework for management of mass casualty situations at collection points via an artificial intelligence (AI) based multilayered approach.

Keywords: decision support, multilayered approach, victim’s data flow, AI & medical informatics, emergency ultrasound, operations research.

1 Introduction

Very often disasters result in mass casualty situations, which trigger a complexity of decisions to be made at casualty collection points and advanced medical posts.

Triage and evacuation of casualties are the most important elements in disaster management chain. In addition, casualty evacuation without an effective coordination may lead to hospitals overcrowding and, therefore,
to an increased number of deaths. So, guidance for rapid transportation (according to triage categories, needed/available ambulances and human resources and destination hospitals capabilities) is needed as well. The medical decision-making process at casualty collection points becomes complicated due to the fact that significant bleeding into the peritoneal, pleural, or pericardial spaces may occur without visible warning signs. The efficacy of medical first response, especially in mass casualty situations, is extremely dependent upon the time of treatment delivery.

In this research we propose to enhance the efficacy of crisis management response in large-scale disasters by designing a decision support (DS) framework for management of mass casualty situations at collection points via an artificial intelligence based multilayered approach. It will support decision-makers/end-users (healthcare personnel and aides), who are dealing in disaster area under time pressure with a considerable number of casualties and have limited resources such as ambulances, available nearby medical centers, and personnel.

2 DS framework based on multilayered approach

The proposed solution is aimed to implement emergency ultrasound in injury assessment at disaster site using portable ultrasound scanners, and offering easy-to-use computer-aided tools for mobile devices, which will help to perform triage (based on vital signs) and more accurate re-triage of casualties with injuries of thorax and abdomen (taking into account level of urgency determined by emergency ultrasound), will suggest efficient therapeutic decisions (life-threatening interventions and emergency diagnostics), and will assist the coordinated evacuation of the injured persons.

The data, analyzed in the framework of this study has multilayered structure. It is obtained during all stages of casualty management at collection points, from treatment to evacuation: registration, triage, re-triage, live-threatening interventions and transportation.

The main objectives proposed within this research are the following:  
0 – equipping medical first aid personnel with modern portable medical ultrasound scanners to be used at collection points;  
1 – development of software for mobile devices with a simple user interface (via voice registration) to gather and organize the primary
AI Based Multilayered Approach for Management of Mass Casualty Situations

medical data of casualties, including assessment of triage priority – **LAYER 1**.

2 – creation of 3 modules of the DS framework:
2.1) **LAYER 2.** DSS for more accurate re-triage of casualties with injuries of thorax and abdomen, based on use of emergency ultrasound equipment, including suggestions of live-threatening interventions to be done before transportation.
2.2) **LAYER 3.** DSS for medical evaluation of stabilized casualties, based on ultrasound features, during transportation or in clinical conditions.
2.3) **LAYER 4.** DSS for efficient placement and transportation of casualties, offering guidance for rapid transportation, based on innovative AI inference and transportation systems frameworks.

3 – Training of all personnel, involved in the process of mass casualty situations response in various modes (from virtual simulation up to "in the real site" exercises).

![Victim flow management. Example Scenario.](image)

**Figure 1.** Victim flow management. Example Scenario.

### 3 Methodology

At the stage of acquisition of professional knowledge in the field of disaster medicine and emergency medical care and creation of knowledge base, the following artificial intelligence methods and algorithms will be
used: advanced technologies for expert medical knowledge acquisition; algorithms for precedents analysis in order to identify new knowledge (patterns); methods of structurization and creation of medical taxonomies; representation of the acquired knowledge in the form of semantic networks, cognitive matrices, and ontologies.

At the stage of collection and storage of information and data, methodology and practice from the domain of relational databases will be used.

For data classification and processing the methods of classical medical statistics will be used.

The DS framework will consist of 4 modules (levels) that can interact (by transferring data automatically between the levels):
1. Module supporting primary triage and registration of casualties.
2. Re-triage module based on mobile ultrasound analysis and suggestions for life-threatening interventions.
3. Module supporting in-depth analysis of injuries of internal organs based on mobile/stationary ultrasound analysis.
4. Module for decisions support of coordinated evacuation of casualties.

At the stage of the user interface development the existent practices, principles and approaches that are used in modern medical information systems web- and mobile applications, will be taken into account.

The software application will be implemented on mobile devices (tablets or mobile phones) using mobile software, embedded SQL database engine, taking into account peer-to-peer connection (via Wi-Fi, Bluetooth or Near Field Communication) and delayed data transmission in case of total network failure. Also we plan to implement on-demand data synchronization between mobile devices to conserve device battery (instead of instant wireless communication).

To develop the decision support framework we will use only OpenSource software products, storage and data exchange protocols, specific for medical information systems. This corresponds to the international practice in the development of medical software, which is planned to be widely disseminated.

In the development process of the decision support framework for management of mass casualty situations at collection points via an
artificial intelligence based multilayered approach, the specific and special competence and experience obtained previously will be used:

- Operations research [1-3];
- Artificial intelligence for medical informatics applications [4-6];
- Emergency ultrasound and disaster medicine [7].

4 Conclusion

The proposed AI Based Multilayered Approach will enhance the response procedures at collection points during disaster situations by implementing the best practices/protocols and technologies from emergency diagnostics, medical informatics and operations research.

The following positive impacts on security issues planned to be achieved:

- Fast registration and triage priority assessment, using the proposed DS framework.
- Implementation of emergency ultrasound protocols and suggestions for life-threatening interventions, given by the proposed software, will imply accurate casualty triage re-assessment and more effective emergency therapy before further transportation and will minimize over- and under-triage.
- Suggestions for emergency diagnostics, given by the proposed software during transportation, will enhance further clinical treatment.
- Software for evacuating casualties will help in efficient distribution of the available resources.

Acknowledgments. The Institutional Projects (Supreme Council for Science and Technological Development) – 15.817.02.02A, and the International Collaboration Project – 17.820.5007.02/Danube has supported the research for this paper.

References


Constantin Gaindric\textsuperscript{1}, Svetlana Cojocaru\textsuperscript{1}, Stefan Pickl\textsuperscript{2}, Iulian Secrieru\textsuperscript{1}, Olga Popcova\textsuperscript{1}

\textsuperscript{1}Vladimir Andrunachievici Institute of Mathematics and Computer Science, Chisinau, Republic of Moldova
E-mail: constantin.gaindric@math.md

\textsuperscript{2}Institute for Theoretical Computer Science, Mathematics and Operations Research, Bundeswehr University Munich, Germany
Performing cricothyroidotomy incision using Artificial Intelligence

Gheorghe-Catalin Crisan

Abstract

The goal of this article is to present the benefits of using Artificial Intelligence for developing a medical application for cricothyroidotomy interventions. Our scope is to create a device that can be used in emergency situations to make the incision in the neck of the victim. The device can be used by any person so that the intervention can take place in a very short time. Therefore, the person attending the accident should not be a specialist to save the person's life. We also aim to show how we can train such a neural network for these interventions.

Keywords: artificial intelligence, medical, first aid, cricothyroidotomy.

1 Introduction

The cricothyroidotomy is an emergency procedure that is performed on patients in respiratory distress [1] where one cannot obtain a definitive airway through orotracheal intubation (from the mouth) or nasotracheal intubation (from nose). Intubations can be blocked for various reasons, anatomy difficulties or physical obstruction, so it can happen anywhere and at any time to any person. The cricothyroidotomy must be performed in a well-defined neck point called the median cricothyroid ligament. If the incision is inadequate, other cartilage and neck ligaments may be affected which may lead to complications. Thus, there is the possibility of severe mutilation.
Using neural networks, we can perform a matrix training process representing the anatomical structure of the neck. Thus, based on the training we can predict exactly which point is the incision. In this way the device could sign the point where the incision is to be made or even more, the device itself to make the incision.

2 Problem definition
The problem is that orotracheal intubation or nasotracheal intubation can block for various causes that can even cause the death of a person, such as:

- the existence of some anatomy difficulties of the person [2]: Class IV Mallampati, short mandible length, limited neck mobility, obesity or sleep apnea problems or existence of physical obstruction of the airway by a large mass or object;
- massive facial trauma;
- anaphylactioc shock;
- choking and suffocation: a critical problem in Europe with an increased rate [4][5].

Another problem is that the implementation of this procedure requires qualified staff. This qualification requires the pursuit of specialization courses that have a high price, so very few people are qualified to carry out the procedure of cricothyroidotomy. In Romania, for example, very few of the people working on emergency ambulances are qualified. For this reason, even if the ambulance arrives in time, the victim has a high chance of dying. Moreover, due to the fact that the ambulance personnel do not know the procedure, they refuse to try to do it because it could aggravate the victim's condition, thus being directly responsible for the complications caused.

3 Proposed solution
The solution proposed by us is a hardware device made up of a cervical collar that has the function of maintaining a fixed position of the neck. This collar is made up of a device that emits radio waves capable of making a 3D mapping into layers of the neck and a Raspberry Pi. Using radio frequencies we can determine which the components of the neck are.
These results are returned as a matrix of values by the radiofrequency device and then processed by the trained model of the neural network. Thus, the resulting image is transmitted to a Raspberry Pi device on which the trained pattern of the neural network is loaded and which accurately determines our reference object, i.e. the median cricothyroid ligament. By attaching the Raspberry Pi to the cervical collar we can do the entire process local without using any cloud service.

In this way, the device can make the incision in the neck or mark the point at which the medical staff can perform this incision.

Figure 1. The prototype device attached to the neck.

4 Theoretical aspects
4.1 Neck anatomy
The shape of the neck in humans is formed from the upper part of the vertebral column at the back and a series of cartilage that surrounds the upper part of the respiratory tract. At the top of the neck we have a thyroid
cartilage on which Adam's apple is. Following Adam's apple down there is a small hole followed by cricoids cartilage. Thus, cricothyroid membrane is located between Adam's apple and cricoids cartilage. Along with the cricothyroid membranes there are two important muscles of the respiratory tract, cricothyroid muscles used in speech to the left and right. These are muscle tensors that help in the phonation process.

Figure 2. Ligaments of the larynx [3].

4.2 Convolutional Neural Networks for object detection

For the implementation of the deep learning algorithm for medial cricothyroid ligament detection we used a convolutional neural network using the sliding window procedure of neural networks [6] like in Fig. 3. As one of the initial versions, we opted for this approach even though the sliding window strategy is considered inefficient and redundant [7]. This is one of the brute force methods by which each frame of the matrix is parsed. One important thing to note is that neural network weights have to be the same for every single frame of the matrix. In this way, we ensure that each frame is treated equally.

So, for detection, we used a window frame of approximately equal magnitude to the ligament. This was obtained from the matrix of
radiofrequency values obtained from the neck scanning process as we can see in Fig. 4, and the ligament having an approximate size of 20x20.

Figure 3. Example of sliding window over the matrix.

Figure 4. Frequencies matrix from neck scan.

So we have a data matrix representing the mapping of a victim's neck. Using the window frame, go through the entire array, from left to right,
from bottom to top. When we find the subject of interest at the current window frame position, it means that ligament detection has been successfully performed.

5 Experimental setup

For the development of the application was used Visual Code IDE with Python 3.6 with the following dependencies:

- OpenCV: used for labeling the object detected in image;
- Keras: used for neural network implementation;
- SKLearn: used for splitting train and test data;
- Numpy.

Frequency matrices used for the training set were labeled by a specialist medical practitioner on the cricothyroidotomy procedure. In this way, we ensure that network training is properly conducted to reduce the number of errors as much as possible. Thus, the layers used in our convolutional neural network are structured as can be seen in the listing presented in Fig. 5.

```python
model = Sequential()
model.add(Conv2D(16, (3, 3), input_shape=(1, 20, 20), activation='relu'))
model.add(MaxPooling2D(pool_size=(2, 2)))
model.add(Conv2D(8, (3, 3), activation='relu'))
model.add(MaxPooling2D(pool_size=(2, 2)))
model.add(Dropout(0.2))
model.add(Flatten())
model.add(Dense(128, activation='relu'))
model.add(Dense(1, activation='sigmoid'))
```

Figure 5. Result with marked incision zone.

Therefore, the network layers are made up of two convolution kernels with 16 and 8 filters of size 3x3 and ReLu activation that are convolved with the layer input. In fact, we have two layers for max pooling used for taking the maximum value in non-overlapping rectangles from the matrix and reducing dimensionality. A dropout layer used to avoid overfitting and forcing the neural network to remain flexible without memorizing some specific patterns. Using dropout layer, we flatten the output layer.
and we will use the fully connected layers used to consolidate the information into output classes for object prediction.

6 Experiments and results

Applying everything presented in previous chapters has made it possible to set the objective of the application. We also managed to get the desired object detection using two convolution layers with 16 and 8 filters and ReLu activation. The model was also trained using Google’s Colaboratory notebook environment. It offers the ability to run in cloud and store projects in Google Drive. Moreover, it allows us to use the GPU through accelerator hardware that uses the Tesla K80 GPU [8].

The CNN model used has been trained around 150 epochs using a training sample of 20 matrixes. The resulting accuracy has a value of 0.65113 that gives us an optimistic result for the future developmental possibilities of the application.

So, following the training of the model, we succeeded in successfully detecting the area of the ligament where the incision should be performed. This can also be seen in Fig. 6 which presents the image of the matrix of values of the frequencies resulting from the scanning of the neck. Thus, the model receives as input this matrix on which it succeeds to detect the ligament and to correctly mark the area in which the incision is to be made.

Figure 6. Result with marked incision zone.
7 Conclusion

This article aims to present how we can create a medical device that allows us to realize a very fine medical procedure that requires prior certification to be achieved. Thus, with the CNN model trained on its own data set and the device in the form of a neck collar that has the function of assisting the neck and scanning it, we have successfully implemented this device.

Some existing problems are related to the fact that the prediction confidence level. Currently, the level is below 0.7 which don’t allow ensuring that the incision mark is correct. Another problem is the time needed to obtain the result. For now, the trained model needs 3 seconds to output the result. Of course, all these aspects can be improved by increasing the data set and finding a better algorithm to train the model.

The application at the level that is achieved so far is functional and allows us to achieve the main objectives that we have considered in its design. The application can be improved by using an object optimization algorithm that is better optimized. A possible solution is YOLO v3 (You only look once) [9] whose detection system does not use a brute force method. The YOLO algorithm divides the image received as input into regions, predicting bounding boxes and probabilities for detecting a particular object in the region. Thus, we can reduce the time the image is being analyzed, saving time that could save lives.

References


Cricothyroidotomy using AI


Gheorghe-Catalin Crisan

Affiliation/Institution: Lucian Blaga University of Sibiu, Faculty of Science, Bulevardul Victoriei 10, Sibiu, ROMANIA
E-mail: stiinte@ulbsibiu.ro
User recognition based on keystroke dynamics

Constantin-Răzvan Cazuc, Adrian Iftene

Abstract

Keystroke dynamics is the analysis of timings and patterns in the way an individual is typing on a keyboard. It has been observed that every person’s typing rhythms and habits can be mathematically modeled in a way that allows us to uniquely identify a user. In this paper, we investigate the performance of some of the metrics proposed in current literature using a dataset collected by us, so as to lower inaccuracies as much as possible.

Keywords: keyboard data collection, behavioral biometrics, equal error rate, user identification.

1 Introduction

The term biometrics refers to metrics related to human measurable characteristics used to label and describe individuals. Some examples include the DNA, fingerprints, palmprints, and retina. The way most persons are identified in today’s society is based on tokens such as ID, driver’s license, passport. The problem with this approach is that they can be forged way more easily than modifying one’s fingerprints for example. While most of the examples described are used to identify a person, others are used in marketing, software application development and helping people with disabilities.

Eye tracking technology has been successfully used to construct devices that help people with locomotors disabilities to be able to use and navigate a computer [1]. It has also been used in webpage design to identify what the subjects are attracted to and pay more attention to. Mouse tracking, including hover times, is also actively used in the creation of online stores to make the user experience as smooth and
captivating as possible so that the potential customers don’t get frustrated when looking for certain features.

Keystroke dynamics analysis has been an evolving field in recent years, many researchers approaching the subject through the use of different anomaly-detection algorithms such as neural networks [10], SVM [11] and k-means [12]. The behavioral biometric of keystroke dynamics relies on the form and rhythm a person types using a keyboard. The rhythm of how a person presses keys on a keyboard can be used to uniquely identify an individual.

The main research question of this paper intend to answer: Could be considered the typing rhythm a way to categorize user's profile?

Over the years several methods of identification have been created and tested, many of them varying in complexity and accuracy. The main data needed to create those techniques are key seek-time and key hold-time. The seek-time represents the amount of time it takes a user to get to a different key and press it, while the hold time is the amount of time between a key being pressed and then the same key being released. Most people have unique attributes that can classify them, such as the time to find a certain letter compared to others. Those letters and patterns can wildly vary from individual to individual. Because the field of keystroke dynamics is statistical in nature, most identifying techniques output a percentage probability of being a certain person rather than an absolute value (true/false).

2 Data Collection

Data collection is a very important step to create correct models for the identification process. Keyboard layouts can differ drastically depending on the language and there are other factors such as the type (membrane or mechanical) that can influence the data patterns. For this test, we used a mechanical keyboard with a standard QWERTY layout that features longer key travel length and anti-ghosting technology (no key presses are lost in case of multiple keys being used at the same time). The method of key and timing logging was a Python script using the pinput library, which gives us access to the Windows API. Data were collected from 3 subjects, all active in the software development industry all of them being right-handed with an age between 23 and 25. All the texts were written
were in English with the mention that none of the subjects are native English speakers, but all of them have English certifications of C1 and above.

There are two methods of identification presented in this paper:

• Identification based on a predetermined string of characters (password);
• Identification based on writing free text.

For the first method, we asked each user to create a password that contains at least 8 characters and includes at least one number [4]. We opted not to include capital letters because of the use of “CapsLock” or the “Shift” key, same being applied for the use of special characters. The subjects were asked to practice their newly chosen password for a couple of minutes. If one of the users mistyped the password that information would be manually removed from the recorded data. The subjects were asked to provide the data during different days and at different hours because keystroke timings are heavily influenced by a person’s state of mind. The total amount of repetitions was 75 for each password. The subjects were also asked to write the other subject’s password 10 times in order to use this data as impostor data.

The second method needs a very large amount of text in order to be able to analyze the patterns and timings. In this data gathering session, we focused on texts with larger amounts of characters. We had each subject type a 500 characters text 5 times, in two different ways. During the first data gathering session, they had the sample text displayed on a different monitor and they had to copy it while seeing what they were writing and allowing the use of keys such as “Backspace”. A lot of timing errors appear when the subject moves his head in order to read the presented text. During the second session, the subjects have dictated a text and they had to write it without seeing what they were writing. This was done in order to see writing errors that were specific to each individual in order to build patterns (such as writing “ahve” instead of “have”).

3 Data Processing
The raw data collected was stored as a series of tuples containing the name of the key pressed, the moment in time it was pressed at and the
moment the key was released. Out of this representation, we can extract a lot of features of the keystroke dynamics including [2]:

- Key hold times;
- KeyDown-KeyDown times between two successive keys;
- KeyUp-KeyDown times between two successive keys;
- KeyUp-KeyUp times between two successive keys;
- KeyDown-KeyUp times between two successive keys.

For our test, we used the key hold times, KeyDown-KeyDown times and KeyUp-KeyDown times. The resulting features were filtered using standard deviation - see formula (1) [5].

\[
\sigma = \sqrt{\frac{1}{N} \sum_{i=1}^{N} (x_i - \mu)^2}
\]  

(1)

Data that was more than 3 times different than the standard deviation was removed from further processing. The next step was to implement 3 anomaly-detection methods (Euclidean, Manhattan and Mahalanobis distance metrics). We used python as the programming language given its ease of use and extensive mathematical library [3].

The Euclidean anomaly-detection algorithm models each timing recorded for a password as a point in an \( n \)-dimensional space, where \( n \) is the number of features calculated for each password. Given our collected data we formed between 70 to 75 timing vectors (after filters) for each password. Our testing methodology consisted of 2 steps. First, the mean vector for our training data was calculated. Second, a score is calculated based on the squared Euclidean distance between the mean vector and a test vector – see formula (2) [8].

\[
d(p, q) = \sqrt{\sum_{i=1}^{N} (p_i - q_i)^2}
\]  

(2)

The Manhattan anomaly-detection algorithm is very similar in terms of steps to the Euclidean one. The main difference is represented by the testing phase where instead of the Euclidean distance we use the
Manhattan distance. This approach is preferable in the case of high dimensional data because the Manhattan distance will try to reduce all errors equally since the gradient has a constant magnitude – see formula (3) [9].

$$d(p, q) = ||p - q||_1 = \sum_{i=1}^{N} |p_i - q_i|$$  \hspace{1cm} (3)

The Mahalanobis anomaly-detection algorithm uses a more complex distance formula that takes into account the similarities between the features of the vectors, to correct for the heterogeneity and non-isotropy observed in most real data. The Mahalanobis distance measures how many standard deviations are from a point to a mean. During the training phase not only the mean vector is calculated but also the covariance matrix of the timing vectors – see formula 4 [9].

$$d(p, q) = (p - q)^T S^{-1} (p - q)$$  \hspace{1cm} (4)

4 Results

To measure the results we had to compute the FAR (False Acceptance Rate) and FRR (False Rejection Rate) for our tests – see formulas (5a) and (5b) [6].

$$\text{FAR} = \frac{\text{Number Of False Acceptance}}{\text{Number of Identification Attempts}}$$  \hspace{1cm} (5a)

$$\text{FRR} = \frac{\text{Number Of False Rejections}}{\text{Number of Identification Attempts}}$$  \hspace{1cm} (5b)

For a secure system, both the FAR and FRR need to be as small as possible. There is still a difference to be made on whether we want to avoid the frustration of the genuine users and create a more accepting model or risk that the genuine users are locked out because of very rigorous constraints.
<table>
<thead>
<tr>
<th>Table 1. Testing Results for Euclidean</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Attempts</strong></td>
</tr>
<tr>
<td>Attempts</td>
</tr>
<tr>
<td>Error</td>
</tr>
<tr>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Table 2. Testing Results for Manhattan</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Attempts</strong></td>
</tr>
<tr>
<td>Attempts</td>
</tr>
<tr>
<td>Error</td>
</tr>
<tr>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Table 3. Testing Results for Mahalanobis</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Attempts</strong></td>
</tr>
<tr>
<td>Attempts</td>
</tr>
<tr>
<td>Error</td>
</tr>
<tr>
<td></td>
</tr>
</tbody>
</table>

Results can vary substantially based on the threshold set for validating a user and the number of features taken into consideration during the training phase of each algorithm. Those results can be compared to the work of Hempstalk *et al.* [6] and the study of Xi *et al.* [7], both showing very small FAR results compared to ours with a massive increase in the FRR results.

5 Conclusion and Future Work

This paper presented the results of some of the approaches that can be used in order to authenticate a person by their typing rhythm. While the Manhattan distance approach provided decent results, the system is still not accurate enough to become a staple worldwide authentication method.

Even if other biometric measurement systems could prove to be more accurate, such as retina and palm scanning, those come at a high cost that not everyone can afford. The main advantage of authentication by the use of keyboard biometrics is its non-invasiveness and ease of use.
User recognition based on keystroke dynamics

References


353

Constantin-Răzvan Cazuc, Adrian Iftene

“Alexandru Ioan Cuza University” of Iași, Faculty of Computer Science
E-mail: razvan.cazuc@yahoo.com; adiftene@info.uaic.ro
Approaches in formalization of a
distributed transaction
compensation

Stefan Saramet, Florin Olariu

Abstract

With the advance of capabilities that distributed systems
currently have, the commonly used architectural pattern for
building such applications is the Microservices pattern. Considering
that each micro-service may have an associated database
(Database-per-service pattern), handling a distributed transaction
has become a significant challenge. The already-existing
mechanism of handling transactions and roll-backs has become
unusable in a micro-service environment system. This paper
presents how transactions are handled by making use of the Saga
pattern and its building approaches (orchestration and
choreography), as well as presenting two formalization method for
executing compensations in case of a transaction step failure.

Keywords: Saga pattern, choreography, orchestration, Saga
calculi, COWS

1 Introduction

Several years ago, software systems that handled business logic had only
the task of processing and storing the data received from requests into a
persistence layer, without any significant interaction with third-parties.

Transactions in this case were easy to implement, because any
operation that would cause a failure inside the transaction logic would
imply a roll-back in the entire transaction and the system would absorb
and handle that specific error accordingly.
Among with the evolution of customer’s needs, the concept of a system that only processes and/or stores data has evolved, leading to more complex systems, that can integrate with other systems, to give a more interesting user experience. Modern business systems have now the role of an entity that executes steps from real-life situations.

In terms of software architecture, in order to tackle the challenge of building a complex system that will have to communicate with other parties as part of a business process, the Microservices architectural pattern has become more and more used, in order to make a system flexible and modular enough to support future changes. This pattern implies that a complex system should be designed as collections of small, task-specific services that can be aggregated and triggered whenever a business transaction would take place.

In contrast to the Monolith architectural pattern, in which all the logic for a business process would be contained inside the same application, this approach enhances the idea of code reuse and easier maintainability for changes to come. On the other hand, this idea of splitting the functionality of a gigantic system into smaller services has important downsides, especially when trying to execute a transaction on this kind of architecture.

In the scenario in which every small service has its own persistence option (Database-per-service pattern), a transaction that needs to make use of several of these services would become difficult and complex to be designed; the existing approach of using local ACID transaction has become unusable, because the entities implied in a business transaction are spanned across multiple systems and a service may not know anything about the other parties implied in a transaction.

This is how distributed transactions have emerged, with the two-phase commit technique (2PC) as the most used when running such kind of transactions. Since a transaction may last longer than expected (long-running transaction), this kind of situation will induce other problems; the mechanics of running a distributed transaction are nevertheless reduced to the idea of resource locking for guaranteeing the ACID properties of a transaction. By holding on to resources when waiting for all the logic of a transaction to be executed successfully, a system may become unresponsive and error-prone.
2 Saga pattern

Since in a microservice architecture the entities that take part into a distributed transaction may not be in the same database and adding the condition that those entities may have one to many relationships, a transaction of this kind may become a serious coding challenge.

One way of tackling this scenario would be to make use of the Saga pattern. The idea of Saga, as how it was described in [11], states that a distributed transaction which alters the state of multiple service databases can be designed as a set of local, atomic transactions that can be completed by each microservice. Events of completion will be sent to the next service, to continue the execution of the bigger transaction flow. This ensures that the resource locking will be kept to a minimum and locally to each participant in the transaction.

There are two techniques that can be used to implement a Saga: (1) the orchestration technique, in which the idea of a coordinator is assumed, that will communicate with each party implied in the transaction and coordinate the execution of the transaction and (2), the choreography approach, in which the services communicate directly between them, by passing messages, executing steps of that transaction in an order defined beforehand.

The paper is organized as follows: in the next section, we define more formally the idea of compensation as a method of undoing the changes in Sagas, as well as methods of formally describing the compensations in a Saga flow in each of the techniques presented above. Section 4 presents a distributed transaction example that will support the notation for each formalization approach (compensation in the orchestration approach – Section 5, and for choreography approach – Section 6). The last section of this paper (Section 7) formulates the conclusions of this research.

3 Compensation

The concept of compensation is widely used in programming languages among with the concept of reparation and it refers to a computer program that is written to undo the actions of another one, recovering a system to a previous state before the process that failed started running. This will ensure that the system will not move to an inconsistent state and data will remain unmodified in case of an error.
Speaking of distributed business transactions, compensation is a sequence of actions that will undo the changes done by the steps already done within the data persistence layer. In this case, a compensation does not imply the fact that the changes already made before the failure occurred in one step of the transaction will be deleted; data will be moved to the previous consistent state by executing other actions that will revert the changes already made, so that the system will not be affected when trying to execute other transactions. Note that a compensation may not be run in the same order as how the initial transaction steps were executed.

In Sagas as well, the notion of compensation is attached to each local step of a transaction execution; for each local business logic that will be executed against the database of each service, a compensation must be present so that, in case of a failure, the local changes will be undone, and, after that, events will be triggered to cancel the bigger transaction.

Studies were conducted to formalize the notion of a compensation, especially for the need of clear semantics and the possibility of applying verification techniques. In the next sections, we will review two techniques of formalizing the compensations in Sagas, one for each technique with which a Saga can be constructed: one for the centralized coordination mechanism – or the orchestration approach and the other for the decentralized collaboration mechanism – or the choreography approach.

4 Example description
To present in a more facile way the notations for each approach mentioned above, we will consider as an example a mortgage application transaction, for the ease of understanding. A customer wants to get a mortgage to buy a house from the bank; so, he submits a request through the internet banking application and then uploads the required documents that will confirm his income.

The bank systems create a process into an external system that will track and trace the mortgage application and stores the uploaded documents into a storage service; once the checks for the documents have passed, another system generates a mortgage offer that is sent to the customer to be reviewed and signed digitally. Once the customer agrees the mortgage offer and the payment conditions, he signs digitally, and the
entire process moves to the last verification steps within the bank. If the checks pass, the mortgage is created, and the customer can use it to buy his desired property.

If a failure occurs within any of the above-mentioned processes, each completed step needs to be compensated. So, if the document verification fails (the customer is not suitable for obtaining a mortgage), the system needs to delete the process from the track-and-trace system and remove the documents from the storage system. If the customer does not agree with the mortgage offer or the terms, the system needs to withdraw the current offer and update and trigger the manual handling of that process. If the mortgage is not created successfully (an internal error from the bank), the documents must be deleted, the process and the offer removed.

To consolidate the notations and the abbreviations described informally above, we will make use of the following schema:

- **Transaction** – the mortgage application transaction;
- **Request** – the process of requesting a mortgage from the client;
- **Upload** – process of uploading the required mortgage documents from the client’s internet banking app and storing into the bank’s document vault;
- **Update** – update of the process step in the bank’s track-and-trace system;
- **Quotation** – the process of generating a mortgage offer with the terms and conditions associated;
- **Review** – the process of signing the offer from the customer;
- **Eligibility** – verification of the possibility and the amount of money to be given to a customer;
- **Champagne** – successful start of a house mortgage;
- **Deny** – rejection of the mortgage request based on condition verified by the bank;
- **Unload** – removal of the documents stored within the bank’s document vault;
- **Withdraw** – the withdrawal of the mortgage offer;
- **Cancellation** – rejection of the signing of the offer by the customer;
- **End** – end of the business process;
- **Operator** – manual processing of the transaction.
For every process described, there is a compensation attached to it as when that specific step fails, the system can compensate and move back to a consistent state.

The tuples can be viewed in this form:

• Eligibility – Deny;
• Upload – Unload;
• Update – End;
• Quotation – Withdraw;
• Review – Cancellation;
• Champagne – End.

5 Compensation in orchestration approach

Orchestration approach, as how it was described in the sections above, assumes the idea of a coordinator (orchestrator), whose job is to supervise the execution of the saga by communicating directly with the other services implied in that transaction. By knowing beforehand, the order of the calls to each service, the coordinator passes message to continue the execution of the distributed transaction. If an error occurs, the coordinator knows each compensation associated to a service and can stop the Saga execution and trigger the compensations for the steps completed.

This approach has the advantage that the services do not need to necessarily know one about each other to be composed as part of a Saga.

In terms of formalization, Saga is a flow composition language based on the idea presented in [1], in which a long-lived transaction can be designed as a series of steps that either succeed totally or fail and will need to be compensated.

A process in Saga language, by definition, is defined by the following grammar:

\[
P ::= 0 \mid A \mid A \div B \mid P; P \mid P|P|\{[[P]]\}
\]

Processes can be the empty activity 0, an activity A (also called a step in [1]) without a specified compensation, or an activity A ÷ B specifying B as compensation for A. Processes can be composed in sequence (P; P) or in parallel (P|P). Sagas can be nested, thus a saga \{[[P]]\} is a process too.

Sagas can be defined as following:
A saga may be an enclosed process \{[P]\}, a Saga S with the associated compensation P (which can have multiple compensations); a saga with exception handling try S with P in which, if S fails, P is executed; or a Saga with alternative forwarding try S or P in which, if S fails, but successfully aborts, P is executed and if P succeeds, the transaction can continue its execution.

Semantically speaking, the idea of this language introduces the idea of contexts, concept that moves away from the idea of code failure. The rule \( \Gamma \vdash \langle P, \beta \rangle \xrightarrow{\alpha} \langle \mathbf{[]}, \beta' \rangle \) states that, under context \( \Gamma \), transaction P terminates if the actual flow of control is \( \alpha \).

A transaction may terminate in three different ways: \( \mathbf{[]}, \mathbf{\Box}, \mathbf{\star} \), where \( \mathbf{[]} \) signifies that a transaction terminated successfully, \( \mathbf{\Box} \) signifies that the transaction has been correctly aborted; i.e. a failure occurred but the transaction has successfully compensated, while \( \mathbf{\star} \) signifies an abnormally terminated transaction; i.e. one which failed and whose compensation has failed as well.

If a step is executed successfully \( (A \div B) \), B will be added as a prefix to the currently accumulated compensation:

\[
A \mapsto \mathbf{[]}, \Gamma \vdash \langle A \div B, \beta \rangle \xrightarrow{A} \langle \mathbf{[]}, B ; \beta \rangle.
\]

Speaking of the compensations for a Saga, the rule below describes the mechanics of executing a compensation:

\[
\begin{align*}
\Gamma \vdash \langle \beta, 0 \rangle & \xrightarrow{\alpha} \langle \mathbf{[]}, 0 \rangle \quad & \Gamma \vdash \langle \beta, 0 \rangle & \xrightarrow{\alpha} \langle \mathbf{\Box}, 0 \rangle \\
A \mapsto \mathbf{\Box}, \Gamma \vdash \langle A \div B, \beta \rangle & \xrightarrow{\alpha} \langle \mathbf{\Box}, 0 \rangle \quad & A \mapsto \mathbf{\Box}, \Gamma \vdash \langle A \div B, \beta \rangle & \xrightarrow{\alpha} \langle \mathbf{\Box}, 0 \rangle
\end{align*}
\]

The rule on the left state that, if the activity A fails, but the compensation B succeeds, the transaction is successfully aborted; the second rule refers to the abnormal termination in which the activity A fails, and the compensation B fails as well, leaving the transaction to an unknown state. An interesting aspect of Saga compensation mechanism is
that the parent of a successfully aborted saga is unaware that one of its child sagas have aborted.

Moving back to the mortgage application transaction, by using the notions defined above, the transaction can be formally described as following:

\[
\text{Transaction} \equiv \begin{cases} 
\text{(Request);} \\
(\text{Eligibility + Deny};) \\
(\text{Upload + Unload};) \\
(\text{Update + End};) \\
(\text{Quotation + Withdraw};) \\
(\text{Review + Cancellation};) \\
(\text{Champage + End}) 
\end{cases} \text{ with Operator}
\]

6 Compensation in choreography approach

Choreography approach in a distributed transaction does not assume the idea of a supervisor; the services that need to be composed to form the higher-level transaction communicate directly through messages. So, in this case, each service knows exactly the state of the other participants in the transaction. Once a step of the transaction is completed, a message is sent to the next service to continue the process; each service in this case knows beforehand how to move forward once its local update has finished successfully.

The language selected for the formalization in this case would be COWS. This language is specifically targeted to provide a way of formalizing web services. In COWS, an endpoint that receives or sends an invocation request has two defining elements: a partner (location) and an operation. The basic syntax of this language does not define the concept of failure, transaction, scope or success; rather than on SAGAs, in which processes were defined, in this language the computational entity is called service and not process.

\[
g ::= 0 \mid p \cdot o? \bar{w}.s \mid g + g \\
S ::= \text{kill}(k) \mid u \cdot u'!e \bar{g} \mid g \mid s \mid s \mid \{[s]\} \mid [d]s \mid *s
\]
The syntax of COWS is given below with \( s \) ranging over services; \( k \) ranging over killer labels; \( w \) ranging over variables and values (\( \bar{w} \) over tuples); \( e \) ranging over expressions (\( \bar{e} \) over tuples); \( d \) ranging over killer labels, names and variables; \( p \) ranging over names used to refer to partners; \( o \) ranging over names used to refer to operations; and \( u, u' \) ranging over names and variables.

An input-guarded choice (\( g \)) is either the inert service 0; the receive activity \( p \cdot o? \bar{w}.s \), receiving \( \bar{w} \) on endpoint \( p \cdot o \) and continuing as \( s \); and the choice of two input-guarded choices, denoted by \( g + g \).

A service is either a kill activity \( \text{kill}(k) \), representing a termination request of \( k \)'s scope; an invoke activity \( u \cdot u'!\bar{e} \), sending \( \bar{e} \) through endpoint \( u \cdot u' \), where \( u \) stands for the partner and \( u' \) stands for the operation; an input-guarded choice; a parallel composition of services \( s \mid s \); a protection \( \{s\} \), protecting \( s \) from being externally killed; a delimitation \( [d] \), binding \( d \) inside \( s \); or a replication of a service \( *s \).

The semantics of COWS is given in terms of a structural congruence and a labelled transition relation. The labelled transition relation \( \alpha \rightarrow \) is the least relation over services induced by several rules of which we only present a selection here. The label \( \alpha \) is generated by the following grammar:

\[
\alpha \ ::= \ uparrow k | (p \cdot 0) \triangleleft \bar{v} | (p \cdot o) \triangleright \bar{w} | p \cdot o) \sigma | \bar{w} \bar{v} | \uparrow
\]

\( \uparrow k \) denotes the execution of a request for terminating a term \( k \); \( (p \cdot 0) \triangleleft \bar{v} \) and \( (p \cdot o) \triangleright \bar{w} \) denote an invoke and a receive action on endpoint \( p \cdot o \) respectively; \( (p \cdot o) \sigma | \bar{w} \bar{v} \) denotes communication over \( p \cdot o \) — receiving parameters \( \bar{w} \) with corresponding values \( \bar{v} \) with substitutions \( \sigma \) (still to take place); and \( \uparrow \) denotes forced termination.

The constructs for handling scopes, exception handling and compensations are defined in terms of the basic syntax introduced above:
By applying the notions described above onto the mortgage application, the transaction can be specified as follows:

\[
\langle\text{catch}(\phi)\{s\}\rangle_k = \langle p_\phi \cdot o_{\text{fault}} ? \rangle \cdot \langle k' \rangle \cdot \langle s \rangle_k'
\]

\[
\langle\text{undo}(i)\rangle_k = \langle p_i \cdot o_{\text{comp}} ! \rangle \cdot \langle x_{\text{done}} \cdot o_{\text{done}} ! \rangle
\]

\[
\langle\text{throw}(\phi)\rangle_k = \langle p_\phi \cdot o_{\text{fault}} ! \rangle \cdot \langle x_{\text{done}} \cdot o_{\text{done}} ! \rangle \cdot \text{kill}(k)
\]
Approaches in formalization of a distributed transaction compensation

References


Stefan Saramet, Florin Olariu


Stefan Saramet\textsuperscript{1}, Florin Olariu\textsuperscript{2}

\textsuperscript{1}Alexandru Ioan Cuza University of Iasi, Faculty of Computer Science Iasi
E-mail: saramet.stefan@info.uaic.ro

\textsuperscript{2}Alexandru Ioan Cuza University of Iasi, Faculty of Computer Science Iasi
E-mail: florin@florinolariu.ro
Comparison between modern
JavaScript frameworks
performance

Ionuț Iacob, Florin Olariu

Abstract

Comparison between most popular JavaScript frameworks: Angular, React, Vue. Metrics used: DOM manipulation, loading time, project size and Lighthouse score. 3 identical applications developed for each framework with the same functionality. Tests ran on all applications and included the results in the paper. Angular offers consistent performance and more tools out of the box. React offers best performance and smaller size. Vue offers very good performance and is more efficient in DOM manipulation.

Keywords: computer science, JavaScript frameworks, frontend development, performance comparison.

1 Purpose

This paper aims to compare the most popular frameworks in the modern JavaScript ecosystem available at the time of writing this paper. These frameworks allowed for much more complex applications to be developed on the frontend part.

The comparison will take into account 4 metrics and offer a conclusion in order to choose the most suitable tool / framework based on the needs of the project.
2 Setup

JavaScript has various frameworks, but the most popular choices are the frameworks that are in the scope of this paper: Angular 1, React 2 and VueJS 3.

In order to provide an accurate comparison and results an identical application has been created in all the above mentioned frameworks, with the same web components structure, data size, routing and styling. The only differences are the framework specific syntax and structure needed in order to achieve the same results (e.g. generated empty arrays of count size for iteration to a specific index in Angular). The 3 framework host applications can be found in a public GitHub repository 4.

The experiment has been conducted by taking into account the following 3 metrics for the comparison:

- **DOM\(^5\) Manipulation**: takes into account how much time is needed by the framework to perform various operations that alter the DOM and update it in real time. This metric was chosen based on the relevancy shown in another comparison article attached in bibliography;
- **Loading Time**: measures the time needed for the application since the web application starts loading until the FMP\(^6\). The loading time has been chosen due to the performance oriented web development of the current frontend ecosystem;
- **Project size**: measures the production-grade application size for each framework (metric is relevant because the applications have identical structure, functionality and styling). Project size was analyzed due to the increase in size of modern web applications;
- **Lighthouse score**: overall performance test tool provided by Google, gives an insight into more detailed areas (e.g. SEO).

---

1 [https://angular.io/](https://angular.io/)
2 [https://reactjs.org/](https://reactjs.org/)
3 [https://vuejs.org/](https://vuejs.org/)
4 [https://github.com/ionut17/jsframeworks-benchmarks-hosts](https://github.com/ionut17/jsframeworks-benchmarks-hosts)
5 Document Object Model
6 First Meaningful Paint
• In each of the metrics above a lower time / size is better.
  Furthermore, the host application is available to be tested in a live version at the following address: http://dizertatie.ionutrobert.com/.

3 DOM Manipulation

The first metric of the comparison is: DOM Manipulation. This is tested in the Grid page of each of the applications. The grid page contains a table with a configurable number of rows (starting with 1,000) and 10 columns.

We will compare the loading time for the starting 1000 rows, creating new elements (1,000 or 10,000), deleting elements (1,000 or 10,000) and deleting all elements. In the last row a median of all the times is presented in order to draw a conclusion for this metric.

Table 1. DOM Manipulation Comparison

<table>
<thead>
<tr>
<th>Name</th>
<th>Angular</th>
<th>React</th>
<th>Vue</th>
</tr>
</thead>
<tbody>
<tr>
<td>First load (1000)</td>
<td>602ms</td>
<td>477ms</td>
<td>481ms</td>
</tr>
<tr>
<td>Create rows (1000)</td>
<td>2494ms</td>
<td>2153ms</td>
<td>1960ms</td>
</tr>
<tr>
<td>Delete rows (1000)</td>
<td>414ms</td>
<td>395ms</td>
<td>290ms</td>
</tr>
<tr>
<td>Create rows (10000)</td>
<td>24.5s</td>
<td>23.1s</td>
<td>23.1s</td>
</tr>
<tr>
<td>Delete rows (10000)</td>
<td>3.2s</td>
<td>3.4s</td>
<td>3.5s</td>
</tr>
<tr>
<td>Delete all rows</td>
<td>122ms</td>
<td>87ms</td>
<td>98ms</td>
</tr>
<tr>
<td>Median Time</td>
<td>5.2s</td>
<td>4.9s</td>
<td><strong>4.8s</strong></td>
</tr>
</tbody>
</table>

The times have been calculated using Google Chrome (version 74.0.3729) and the developer tools provided. Each test has been run 5 times and the average values have been selected.

Based on the values above, the major differences appear when working with large datasets (> 10,000 rows). The difference between frameworks are scalable with the datasets and Vue performs the best out of the 3 frameworks.

4 Loading Time

The loading time metric measures the time needed for the application since the web application starts loading until the FMP.

The measurements are made in each of the 3 pages of the application: Home, Grid and Forms. The Forms page contains a large number of inputs.
which slows down the rendering in order to have more accurate results for this metric.

Table 2. Loading Time Comparison

<table>
<thead>
<tr>
<th>Page</th>
<th>Angular</th>
<th>React</th>
<th>Vue</th>
</tr>
</thead>
<tbody>
<tr>
<td>Home</td>
<td>590ms</td>
<td>514ms</td>
<td>564ms</td>
</tr>
<tr>
<td>Grid</td>
<td>602ms</td>
<td>477ms</td>
<td>481ms</td>
</tr>
<tr>
<td>Forms</td>
<td>601ms</td>
<td>523ms</td>
<td>659ms</td>
</tr>
<tr>
<td>Median Time</td>
<td>597ms</td>
<td><strong>504ms</strong></td>
<td>568ms</td>
</tr>
</tbody>
</table>

The measurements were made by loading the page 5 times and making an average under the same conditions. The home page time contains loading all the page assets. The Grid, Forms pages times are calculated on page switching.

The time differences above are small but are representative for each framework capability, because these differences scale with the complexity of the application.

5 Project Size

Each framework (Angular, React, Vue) have different implementations and project sizes. The projects were created from scratch using the CLI\(^7\) provided by each framework and generating the recommended starting boilerplate. Based on the starting boilerplate the same features, structure & styling was added to each web application (with some minor framework related limitations).

To further optimize the size and provide a comparison for real use case scenarios the production build for each framework was used in order to get the processed (e.g. minified, tree shaking\(^8\)) final files.

Table 3. Application Size Comparison

<table>
<thead>
<tr>
<th>Mode</th>
<th>Angular</th>
<th>React</th>
<th>Vue</th>
</tr>
</thead>
<tbody>
<tr>
<td>Development</td>
<td>413 KB</td>
<td>668 KB</td>
<td>431 KB</td>
</tr>
<tr>
<td>Production</td>
<td>387 KB</td>
<td><strong>63 KB</strong>(^*)</td>
<td>154 KB(^*)</td>
</tr>
</tbody>
</table>

\(^7\) Command Line Interface  
\(^8\) Dropping unused code
Comparison between modern JavaScript frameworks performance

The comparison is made in Windows 10 64bit using the file explorer details feature and taking the size property into account. Further we will consider only the Production size as being relevant. The Development size is only to show the optimizations capabilities.

For Vue / React, in the production mode, the .map files were not included in the size count because they are used only for debugging purposes.

6 Lighthouse Score

The Lighthouse tool (provided by Google) which assesses web applications were used in order to get a rating for the boilerplate of the host applications for each framework.

Table 4. Lighthouse Report

<table>
<thead>
<tr>
<th>Metric</th>
<th>Angular</th>
<th>React</th>
<th>Vue</th>
</tr>
</thead>
<tbody>
<tr>
<td>Performance</td>
<td>97</td>
<td>99</td>
<td>99</td>
</tr>
<tr>
<td>Accessibility</td>
<td>100</td>
<td>100</td>
<td>92</td>
</tr>
<tr>
<td>Best Practices</td>
<td>86</td>
<td>86</td>
<td>86</td>
</tr>
<tr>
<td>SEO</td>
<td>91</td>
<td>91</td>
<td>91</td>
</tr>
<tr>
<td>Average</td>
<td>93.5</td>
<td>94</td>
<td>92</td>
</tr>
</tbody>
</table>

7 Conclusion

Considering the results in the 4 sections presented we can draw the conclusion that each framework has perks and downsides. Overall, React had the most consistent behavior and offered the best results.

Table 5. Conclusion Comparison

<table>
<thead>
<tr>
<th>Name</th>
<th>Angular</th>
<th>React</th>
<th>Vue</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dom Manipulation</td>
<td>5.2s</td>
<td>4.9s</td>
<td>4.8s</td>
</tr>
<tr>
<td>Loading Time</td>
<td>597ms</td>
<td>504ms</td>
<td>568ms</td>
</tr>
<tr>
<td>Project Size</td>
<td>387 KB</td>
<td>63 KB</td>
<td>154 KB</td>
</tr>
<tr>
<td>Lighthouse Score</td>
<td>93.5</td>
<td>94</td>
<td>92</td>
</tr>
<tr>
<td>Best scores</td>
<td>0</td>
<td>3</td>
<td>1</td>
</tr>
</tbody>
</table>

The comparison table counts for each framework the number of times a framework got the best score for a specific metric.
Angular provides many functionalities out of the box and has comparable performance to the other versions. The only downside would be the larger size and heavier DOM.

React is suitable for performance-oriented apps with a fast development requirement.

Vue is suitable for DOM operation intensive apps, where pages are very dynamic.

In conclusion, each framework it’s most suitable in different scenarios and can be chosen based on performance tests and project requirements.

References


Ionuț Iacob¹, Florin Olariu²

¹Affiliation/Institution: Faculty of Computer Science, Alexandru Ioan Cuza University, Iași
E-mail: ionut.iacob17@gmail.com

²Affiliation/Institution: Faculty of Computer Science, Alexandru Ioan Cuza University, Iași
E-mail: olariu@gmail.com
Opportunities for Phone Account
Recharging from card using the possibilities offered by Optical Character Recognition

Cristian Cosneanu, Inga Titchiev

Abstract
This paper aims to use the possibilities offered by Optical Character Recognition in order to create an iOS application to recharge the phone account from recharge cards in real-time scanner form by ensuring achievement of the high accurate rate in recognition.

Keywords: optical character recognition, image processing, image filtering, Apple Xcode.

1 Introduction
Image processing and analysis includes all the techniques and methods of acquisition, storage, display, modification and operation of visual information contain images. In particular the characters recognition from scanned images of documents is a well known problem that has received much attention in the field of image processing, pattern recognition and artificial intelligence. Unfortunately classical methods in pattern recognition, due to the size, shape and style do not suffice for the recognition.

Optical Character Recognition (OCR) [3] is a process of converting alphabets in images to computer readable coded text.

Today, technology behind OCR has evolved hugely since its inception and many possibilities exist including font recognition [1], image to
document conversion, in OCR online applications as well as software packages.

Character recognition is a hard problem but by focusing on just the text in an image, we were able to obtain much higher recognition accuracy. Optical Character recognition [2, 4] also refers to the process of translating the handwritten or printed text into a format that is understood by the machines for the purpose of editing, searching and indexing. Therefore an iOS application that would be able to read the number on the recharge card (Figure 1) using the phone's camera, then to make a call to the selected operator with the previously read number will be done.

![Recharge card](image)

Figure 1. Recharge card

2 Implementation

Application development on the iOS platform can only be developed on MAC OS with the IDE provided by Apple Xcode. This application was created in a few steps:

- Reading data from the phone’s camera;
- Processing the received image to find the position of the text;
- Cut and filter the founded image;
- Reading the characters on the founded image using Tesseract OCR;
- Creating the read code and making the call.
Opportunities for Phone Account Recharging...

• Operator selector.

2.1 Reading data from your phone's camera
Apple gives us the ability to read the camera data through the AVCaptureSession object. This object works directly with the camera and provides us with a CIImage object, which is an image that will eventually be processed.

2.2 Processing the received image to find the position of the text
Tesseract OCR is a library [5] that awaits an image input and returns a string of characters.

Tesseract is a great general purpose OCR tool that, while trained to recognize text in documents, is also capable of working on a large variety of problems. Like many other models, it requires that images be pre-cropped to contain only text – which means that it works extremely well.

The lower image gives the sooner answer. Therefore focusing on just the text containing the code for recharge in an image, will allow us to obtain much higher recognition accuracy.

This is why to find the position of the scraped number appeared. Finding the position of a text in an image is done using the CITextFeature object, which is a simple quadrilateral \((x, y, length, width)\).

2.3 Cut and filter the founded image
Taking the coordinates of the text from the previous step, we can cut off the text portion of (Figure 2):

![Figure 2. Image after cutting](image)

Tesseract works best when the image is clean, and the colors are well defined. This image cropping is not the only possible optimization to shorten the reading time. Another optimization is image filtering. And in particular the Threshold filter can do that. This filter reduces the image in 2 colors (black and white) and highlights the edges, making it perfect for optimizing reading time.
2.4 Reading the characters on the founded image using Tesseract OCR

As mentioned above, Tesseract is waiting for a picture at the entrance. So we take the image from (Figure 3) and give to tesseract and he returns a string of characters "0651137158416425".

During these processes, Tesseract uses:

1. algorithms for detecting text lines from a skewed page;
2. algorithms for detecting proportional and non proportional words (a proportional word is a word where all the letters are the same width);
3. algorithms for chopping joined characters and for associating broken characters;
4. linguistic analysis to identify the most likely word formed by a cluster of characters;
5. two character classifiers: a static classifier, and an adaptive classifier which employs training data, and which is better at distinguishing between upper and lower case letters.

2.5 Creating the read code and making the call

Once we have the string, the number is dialed according to the selected phone operator, and a call is made using the UIApplication.sharedApplication().method.OpenURL.

2.6 Operator selector

Every phone operator that has recharging cards, do it by recognizing a call to a certain number, with a certain pattern.

In the example above (Figure 1), the pattern is

*100* 16 digits number #
There is no good way of reading this *100* code automatically, every operator has its own, and some operators even implement gift codes, so that you can recharge another phone number. Gift codes usually go by the pattern:

*100* phoneNumber * 16 digits number #

So was implemented a module that allows the user to manually specify the operator only once, all the operators added by user are saved in a local database called *Realm*. The user has the power to add new operators, edit existing ones or clear all the previously added ones.

![Operator editor](image)

Figure 4. Operator editor

3 Conclusion

In this paper the iOS application to recharge the phone account from recharge cards in real-time scanner form by ensuring achievement of the high accurate rate in recognition by using the possibilities offered by Optical Character Recognition was done. Creating applications of such
complexity will always be accompanied by problems. Most often because of the misunderstanding of the used technologies. The good part is that with problem solving, it comes technology understanding, which is probably the most important part of an IT employee, the experience.

References


Cristian Cosneanu$^{1,2}$, Titchiev Inga$^{1,3}$

$^1$Tiraspol State University, Chisinau, Republic of Moldova
$^2$Endava, Chisinau, Republic of Moldova
E-mail: cosneanuc@gmail.com

$^3$Vladimir Andrunachievici Institute of Mathematics and Computer Science, Chisinau, Republic of Moldova
E-mail: inga.titchiev@math.md
SmartID - a PrivateSKY based platform for medical usage in respect with GDPR

Mircea Hulea, Mihai Mart, Andrei Panu, Ștefan Romanescu, Ștefana Toader, Corneliu Munteanu, Andrei Scutelnicu

Abstract

This paper describes a suggested approach to the ever increasing necessity in the healthcare system of a solution able to generate standard forms from scanning the identity card. For example, in the case of medical examination of people applying for driving licenses, the developed application will enable the generation of both the individual record and the internal documents (the internal medical records of the medical units) for the persons undergoing the medical testing procedure. This system brings advantages in two directions. On the one hand, it reduces the time and effort of staff at the triage of the medical unit, who will no longer have to fill in the personal data of the applicants and the system is not prone for errors. On the other hand, it reduces the time spent by specialists to complete the patient records. With the signature pad and the PrivateSky module integrated into the application architecture, it is ensured that the healthcare facility are complying with the provisions of the GDPR agreement.

Keywords: OCR, GDPR, Medical System, PrivateSKY.

1 Introduction

Currently in Romania, as well as in countries all over the inside or outside of the European Union, there are different companies which can develop products and/or services for other countries from the UE. Accounting these new regulations in place, all the companies are looking for solutions
to implement the European Regulation no.679/27.04.2016 [1], concerning the confidentiality of all the data collected from customers, as well as how they can manage and share it (this regulation is known in mass-media as GDPR [1] - General Data Protection Regulation).

Of particular concern are the drastic sanctions enforced in case of breaching the legal rights regarding data management. This risk may be reduced or altogether avoided, by training all the staff involved in managing personal data on GDPR [1] regulations or through using a system (like SmartID) that offers an environment in respect with data protection.

A business company registered at the Trade Registry Office is processing different customer’s private data from its establishment and until the end of its activity. The same applies to any professional from any business field, regardless of any organizational aspect or if it is or not a legal entity (i.e. medical unit, bailiff, notary, lawyer office etc.).

Personal character data represents any information about identified or unidentified individual (“marked person”); an identified individual can be represented, directly or indirectly, especially taking into consideration an identification element, such as the name, number, local position or different specific elements as physical, physiological, genetic, economic, cultural or social identity.

In case an unidentified individual is described in this information in such a way that can be identified by making subsequent researches, this kind of information has the value of personal data. Examples: name, address, phone number, email, salary, IP address, ID, image, etc., but also special categories of data having personal character as:

- ethnic or racial origin;
- public views, religious trusts or other nature;
- health state or sexual life, genetics data or biometrics;

Personal data character with reference to criminal convictions which need special protection must be permitted only with specific guarantees.

Main problem of the medical units when it takes to implement tasks related to personal data retrieval in respect to GDPR, it’s approached by the proposed application. The agreement for data managing of the patients it is implemented by creating special connection between program and signotec Pad for the digital signature. The purpose of this connection is to
obtain and manage the patient signature so it can be adjusted in different PDF documents where the patient agreement on managing the personal information can be found. This way we can implement and at the same time respect every aspect regarding the European GDPR agreement about managing people information. The PDF document created subsequent to the patient agreement contains different personal data (name, surname, date, and signature) and it can be viewed or deleted anytime it's needed. The problem of holding online all the patients’ information has been solved by using PrivateSky, safe and complying to all the agreements on personal data protection (GDPR).

Since the implementation of the GDPR agreement with the help of the signotec Pad and by integration of the PrivateSky system, all the problems concerning the GDPR Agreements in the healthcare system have been solved.

2 SmartID system

The automation application of the proposed process it’s been written in Java and it uses MySQL as database. The patients ID card is scanned using the ID scanner, this hardware component can deliver the app the image of ID card which is written by the OCR [9] (optical character recognition) algorithm (Tesseract). An OCR is a system that transforms text images, such as scanned documents, into text characters [7], [10]. Also known as text recognition, OCR makes it possible to edit and reuse text contained in scanned images. OCR uses an artificial intelligence form, known as model recognition, to individually identify the characters of a text on a page, including punctuation marks, spaces, and end of line [8]. Tesseract [11] is one of the most popular Java libraries to implement an OCR solution. The latest version of a library, adds an OCR algorithm based on LSTM (long and long-term memory neural network) and models for many other languages and scripts in a total of 116 languages. The extracted information from the ID card image will be encrypted using the API generated by the PrivateSky (Figure 1). The encryption of the users’ data from the application (medical staff, administration, and other structures) is performed using in-house crypting.
The SmartID system is divided in several modules. The first module represents the scanning of the ID card of the patient, activity stored at the clinic registry. Without this scanning module in introducing the data of the patient in the database of the clinic, the rest of the modules cannot open. After this stage, the user from the registry sends the data to another medical department (as the beginning of a medical examination).

The software will open the Log-in window (Figure 2) where we can type the username and the password to access it. Each user has defined attributes (Figure 2) and by this each user will open a different window of the program, specific for each department. There can be three types of users: admin, registry operator and doctor.
Software has integrated the following features:

- Logging and the registry of the users;
- Users have different attributes depending on occupied position in the medical unit;
- Different functions for the registry unit users;
- Different features for every medical expertise domain;
- Features which integrates GDPR and PrivateSky agreements;
- Features for the development of medical files.

PrivateSky has developed the concept of Cloud Safe Box (CSB)\[2\][3] as an improvement to the wallet concept used by crypt-coins through private data storage facilities as well as sharing capabilities for distributed file systems\[6\]. Functioning as a simple database can not be the role played by the blockchain in storing private data, identity data, or data with a high level of secrecy in general.

One of the most important aspects of distributed systems is sharing information. For systems with executable choreographies \[4\], there is an issue of efficient file transfer between organizations, people, etc. This information has a high level of confidentiality. This file transfer must be
done under the control of executable choreographies used as a signaling and control mechanism but ideally should provide security and control mechanisms superior to file systems and classical transfer methods [5].

In the development of the SmartID application, integrating the PrivateSky system, it was called an API that generates a CSB with the encrypted data of the personal information of the patient. The CSB generates a string which contains all the encrypted information of the personal data that are used in a medical examination. These steps are detailed below:

- the first implemented step, consists of a http query to http://localhost:90/beginCSB/ to get the transaction key to use for the next steps. This key is required in the communication between the program and the PrivateSky server to distinguish between simultaneous communications between the client and the server;
- after successfully obtaining the queue, the second step, where one or more files are loaded from the client interface. This is done by a http query at http://localhost:90/attachfile/ followed by the key obtained in the previous step plus the file name obtained and the files uploaded by the user of the client;
- in the third step, a backup is made to avoid certain errors that may occur during client-server communications. This is also done through an http call: http://localhost:90/addBackup/ to which the key obtained at the first step is added and the address at which the backup will be available;
- in the last step, to obtain the required token for accessing the files by the patient is done all the print an http call at http://localhost:90/buildCSB/ to which the same key obtained at the first step is added. In this query, the url and channel data were sent and transmitted. If all four queries have been successfully completed, a private token will be returned by the PrivateSky server to be saved and handed to the patient to access the documents loaded by medical staff. The answer to the four queries that is displayed in the console can be viewed in Figure 3.

<table>
<thead>
<tr>
<th>Status BeginCsb: 200</th>
<th>AnswerBegin:</th>
</tr>
</thead>
<tbody>
<tr>
<td>384</td>
<td>384</td>
</tr>
</tbody>
</table>

384
Having the GDPR regulation [1], with this API, the personal information that are used in this developed software are encrypted and accessible only to the user whose personal data are used. As can be seen in Figure 3, at AnswerBuildCSB step, it is generated a string that contains encrypted personal data of a specific user (a patient that was consulted by a medic).

### 3 SmartID use cases

The software, developed for businesses processing medical examination files for the driver license, can also be identified with the use of other specializations like: internal medicine, ORL, neurology, orthopedics-traumatology, ophthalmology and psychiatry. These specializations were also introduced in the software database by creating tables for each one. In these tables the data of the medical examinations are stocked for each specialization. The doctor who uses the software has the possibility to see all the history stored in medical files for the patient from every specialization. A medical examination may be done only if the patient is scheduled by the registry department for that specialization.

Each specialization represents a module for this project. Every module contains specific information for its own medical specialization.

Taking into consideration the registry module, the user will have the possibility of scanning a new ID card, or if it has already an old record from the clinic, it can be selected from database.

Also this module, after scanning the ID card, we open a window which will help the registry user to check if the introduced data is correct and if it’s not, he has the possibility to modify if that’s the case. After this
step, the user responsible with the registry will introduce the patient an
examination for a medical specialization.

All of the medical specialization users can complete the necessary
data for the examination, can generate different medical documents, and
can upload documents generated in PrivateSky so those can be available
online to patients, can print the access code generated by PrivateSky. A
use-case diagram with all the functionality of the software is represented
by the Figure 4.
As an example of using the application it can be talked about the next use case:

- the patient presents to the medical unit, where he/she presents the identity card (ID card);
- at the medical unit at the registry, the ID card is scanned, processed with the OCR algorithms developed within SmartID application and all personal data are uploaded to a database;
- after the data are uploaded to the database, the registry staff makes the assignment to a medical examination;
- in the specialized medical exam, the doctor will open the application for the consultation and will fill the fields specific to the clinical examination;
- after finalizing the consultation, the doctor will print an examination paper, which contains the CSB string;
- with this string, the patient can check on a platform, which is also written on the clinical data sheet, which of his/hers personal data are used by that medical unit.

4 Conclusions

Our developed software with the integration of the PrivateSky platform, developed in collaboration with the team from UAIC, offer a full package software, asked and needed by medical facilities. This assures the complete confidentiality of the entire patient’s personal data as regulation in GDPR. The compliance of this agreement is assured by using a signature pad module in the software and also the PrivateSky servers which guarantee the safety of all the documents, creating available medical examination online for patients with the GDPR agreement respected. Without these functionalities, medical units can guarantee the protection of personal data, more with difficulties and unstable.

In conclusion, the software package implemented with the PrivateSky API, offer to medical units a large scale of functionalities and help users by easing the methodology of applying the GDPR agreement.
References


PDF Transformer 2.0 PRO - two applications in a single working solution with PDF documents http://www.comunicatedepresa.ro/star-storage1/pdf-transformer-2-0-pro-doua-aplicatii-intr-o-singura-solutie-de-lucru-cu-documente-pdf/

SmartID - a PrivateSKY based platform for medical usage in respect with GDPR

Mircea Hulea\textsuperscript{1}, Mihai Mart\textsuperscript{1}, Andrei Panu\textsuperscript{2}, Ştefan Romanescu\textsuperscript{2}, Ştefana Toader\textsuperscript{2}, Corneliu Munteanu\textsuperscript{2}, Andrei Scutelnicu\textsuperscript{2}

\textsuperscript{1}3DTravel Soft-Art SRL
E-mail: \{mhulea, mihai.mart\}@gmail.com

\textsuperscript{2}Faculty of Computer Science, “Alexandru Ioan Cuza” University of Iaşi
E-mail: \{andrei.panu, stefan.romanescu, stefana.toader, andreis, cmunteanu\}@info.uaic.ro