Proceedings MFOI-2016

Conference on Mathematical Foundations of Informatics

Institute of Mathematics and Computer Science
July 25-30, 2016, Chisinau, Moldova

CZU 51+004(082)
M 48

INSTITUTE OF MATHEMATICS AND COMPUTER SCIENCE
5, Academiei street, Chisinau, Republic of Moldova, MD 2028
Tel: (373 22) 72-59-82, Fax: (373 22) 73-80-27,
E-mail: imam@math.md
WEB address: http://www.math.md

Editors: Prof. S.Cojocaru, Prof. C.Gaindric.

Authors are fully responsible for the content of their papers.

Descrierea CIP a Camerei Naţionale a Cărţii

Conference on Mathematical Foundations of Informatics: Institute of Mathematics and Computer Science, Jule 25-30, 2016, Chişinău, Moldova: Proceedings MFOI-2016/Inst. of Mathematics and Computer Science; ed.: S. Cojocaru, C. Gaindric. – Chişinău: Institute of Mathematics and Computer Science, 2016 (Tipogr. "Valinex" SRL). – 351 p.

Referinţe bibliogr. la sfârşitul art. şi în subsol. – Apare cu sprijinul financiar al Information Society Development Institute.

ISBN 978-9975-4237-4-8.

51+004(082)

ISBN 978-9975-4237-4-8

This issue is supported by the Information Society Development Institute

Part 1

Invited papers

Proceedings of the Conference on Mathematical Foundations of Informatics
MFOI2016, July 25-29, 2016, Chisinau, Republic of Moldova

Propositional inquisitive logic: a survey

Ivano Ciardelli

Abstract

This paper provides a concise survey of a body of recent work on propositional inquisitive logic. We review the conceptual foundations of inquisitive semantics, introduce the propositional system, discuss its relations with classical, intuitionistic, and dependence logic, and describe an important feature of inquisitive proofs.

Keywords: questions, inquisitive logic, dependency, intermediate logics, proofs-as-programs.

1 Introduction

Inquisitive semantics stems from a line of work which, going back to [12], has aimed at providing a uniform semantic foundation for the interpretation of both statements and questions. The approach was developed in an early version, based on pairs of models, in [13, 16]; it reached the present form, based on information states, in [3, 9], where the associated propositional logic was also investigated. An algebraic underpinning for the inquisitive treatment of logical operators was given in [19]. The foundations of the inquisitive approach have been motivated starting from a language-oriented perspective in [11], and starting from logic-oriented perspective in [7, 8].

The aim of this paper is to provide a short survey of the work done on propositional inquisitive logic, drawing mostly on [3, 9, 6, 8]. More precise pointers to the literature will be provided when discussing specific topics. We will start in Section 2 by showing at a general level how questions can be brought within the scope of logic by means of a simple

but fundamental shift in the way semantics is viewed. In Section 3, we instantiate this general approach in the propositional setting, introducing propositional inquisitive logic. In Sections 4, 5, and 6, we examine the connections of this logic to the propositional versions of classical logic, intuitionistic logic, and dependence logic. In Section 7 we discuss inquisitive proofs and their constructive content. In Section 8, we present an extension and a generalization of propositional inquisitive logic. Section 9 wraps up and concludes.

2 Bringing question into the logical landscape

Traditionally, logical entailment captures relations such as the one exemplified by (1): the information that Alice and Bob live in the same city, combined with the information that Alice lives in Amsterdam, yields the information that Bob lives in Amsterdam.

(1) Alice and Bob live in the same city
Alice lives in Amsterdam

Bob lives in Amsterdam

Inquisitive logic brings questions into this standard picture, broadening the notion of entailment so as to encompass patterns which we might write as in (2): the information that Alice and Bob live in the same city, combined with the information on where Alice lives, yields the information on where Bob lives.

(2) Alice and Bob live in the same city
Where Alice lives

Where Bob lives

Notice the crucial difference between the two examples: in (1) we are concerned with a relation holding between three specific pieces of information. The situation is different in (2): given the information that

Alice and Bob live in the same city, *any* given piece of information on Alice's city of residence yields some corresponding information on Bob's city of residence. We may say that what is at play in (2) are two *types* of information, which we may see as labeled by the questions *where Alice lives* and *where Bob lives*. Entailment captures the fact that, given the assumption that Alice and Bob live in the same city, information of the first type yields information of the second type.

The entrance of questions into the logical arena is made possible by a fundamental shift in the way the semantics of a sentence is construed. In classical logic, the meaning of a sentence is given by laying out in what states of affairs the sentence is *true*; however, this truth-conditional view does not seem suitable in the case of questions. In inquisitive logic, by contrast, the meaning of a sentence is given by laying out what information is needed in order for a sentence to be *supported*. Accordingly, sentences are evaluated relative to objects called *information states*, which formally encode bodies of information.

Unlike truth-conditional approach, the support approach is applicable to both statements and questions. To give concrete examples, a statement like (3-a) is supported by an information state s if the information available in s implies that Alice lives in Amsterdam; on the other hand, a question like (3-b) is supported by an information state s if the information available in s determines where Alice lives.

(3) a. Alice lives in Amsterdam.
b. Where does Alice live?

This more general semantic approach comes with a corresponding notion of entailment, understood as preservation of support: an entailment holds if the conclusion is supported whenever all the premises are. Assuming a natural connection between the truth-conditions of a statement and its support conditions—namely, that a state supports a statement iff it implies that the statement is true—this notion of entailment coincides with the truth-conditional one as far as statements are concerned. The novelty, however, lies in the fact that now, questions can also participate in entailment relations. Thus, for example, we can

indeed capture the pattern in (2) as a case of logical entailment. To see this, suppose an information state s supports the premises of (2): this means that the information available in s implies that Alice and Bob live in the same city, and also determines in which city Alice lives; clearly, then, the information available in the state determines in which city Bob lives, which means that the conclusion of (2) is supported.

The one discussed in this section is a very general approach to logic, which can be instantiated by a range of concrete systems, differing with respect to their logical language and to the relevant notion of information states. Just as for classical logic, we have inquisitive logics of different sorts: propositional, modal, first-order, etc. The remaining sections of the paper provide an overview of the results obtained in the most basic and best understood setting—the propositional one.[1,2]

3 Propositional inquisitive logic

The language of propositional inquisitive logic, InqB, is the propositional language built up from a set of atomic sentences and $\bot$ by means of conjunction, $\wedge$, implication, $\to$, and inquisitive disjunction, $\mathbin{\mathbb{V}}$.

$$\phi \ ::= \ p \mid \bot \mid \phi \wedge \phi \mid \phi \to \phi \mid \phi \mathbin{\mathbb{V}} \phi$$

Negation and classical disjunction are defined by setting $\neg\phi := \phi \to \bot$, and $\phi \vee \psi := \neg(\neg\phi \wedge \neg\psi)$. Formulas that contain no occurrence of $\mathbb{V}$ are called *classical* formulas.

In the propositional setting, an information state is construed as a set of propositional valuations. The idea here is that a set s encodes the information that the actual state of affairs corresponds to one of the valuations in s. This means that if $t \subseteq s$, then t contains at least as much information as s, and possibly more.

[1] For discussion on the semantic foundations of the inquisitive approach, on the role of questions in logic, and on the relation between truth and support, see [6, 8].

[2] The research in inquisitive modal logic and inquisitive first-order logic has also been growing rapidly in these last few years. Recent work includes [10, 4, 6, 22].

The clauses defining the relation of *support* relative to an information state are the following ones:

- $s \models p \iff w(p) = 1$ for all $w \in s$
- $s \models \bot \iff s = \emptyset$
- $s \models \phi \wedge \psi \iff s \models \phi$ and $s \models \psi$
- $s \models \phi \vee \psi \iff s \models \phi$ or $s \models \psi$
- $s \models \phi \to \psi \iff \forall t \subseteq s : t \models \phi$ implies $t \models \psi$.

A key feature of the semantics is *persistency*: if ϕ is supported by an information state s, then it is also supported by any state $t \subseteq s$ which contains at least as much information. This means that as information grows, more and more formulas become supported. In the information state $\emptyset$, which represents the state of *inconsistent* information, every formula is supported. This may be regarded as a semantic analogue of the *ex falso quodlibet* principle.

4 Relations with classical logic

In inquisitive logic, the fundamental semantic notion is that of support relative to an information state. However, the notion of *truth* relative to a particular valuation w can be recovered by setting: $w \models \phi \iff \{w\} \models \phi$. It is then easy to check that all classical formulas receive the standard truth-conditions.

For some formulas, support at a state simply amounts to truth at each world in the state. If this is the case, we say that the formula is *truth-conditional*. More formally, ϕ is truth-conditional in case for all states s: $s \models \phi \iff \forall w \in s, w \models \phi$. We regard truth-conditional formulas as corresponding to *statements*. The intuition is that there is only one way for an information state s to support a statement: the information available in s must imply that the statement is true.

As a matter of fact, large classes of formulas in InqB are truth-conditional. In particular, all classical formulas are.

Proposition 1. All classical formulas are truth-conditional.

This means that all classical formulas receive essentially the same treatment as in classical propositional logic: their semantics is fully determined by their truth-conditions, which in turn are the standard ones. This is reflected by the relation of entailment among these formulas.

Proposition 2 (Conservativity over classical logic)**.**
Entailment restricted to classical formulas coincides with entailment in classical propositional logic.

This means that the classical fragment of InqB can be identified for all intents and purposes with classical propositional logic, and our logic may be regarded as a conservative extension of classical propositional logic with an inquisitive disjunction operator.

Formulas formed by means of inquisitive disjunction are typically not truth-conditional. We take such formulas to correspond to *questions*. For instance, the formula $p \mathbin{\mathbb{V}} \neg p$, abbreviated as $?p$, corresponds to the question *whether p or not p*. An information state can support this formula in two different ways: either by implying that p is true, or by implying that p is false. Similarly, the formula $p \mathbin{\mathbb{V}} q$ can be regarded as encoding the question *whether p or q*, which can be supported either by establishing that p is true, or by establishing that q is true.[3]

Like in classical logic, formulas in inquisitive logic can be written in a very constrained normal form: namely, any formula of InqB can be written as an inquisitive disjunction of classical formulas.

Theorem 1 (Inquisitive normal form)**.**
Recursively on ϕ, we can define a set $\mathcal{R}(\phi) = \{\alpha_1, \ldots, \alpha_n\}$ of classical formulas, called the *resolutions* of ϕ, such that $\phi \equiv \alpha_1 \mathbin{\mathbb{V}} \ldots \mathbin{\mathbb{V}} \alpha_n$.

Intuitively, we can regard the resolutions of a formula as capturing the different ways in which the formula may be supported. If ϕ is a

[3] An exclusive reading of the question *whether p or q* can be formalized as well, by translating the question as $(p \wedge \neg q) \mathbin{\mathbb{V}} (q \wedge \neg p)$.

classical formula, then it can be supported in only one way, by establishing that it is true; accordingly, we have $\mathcal{R}(\phi) = \{\phi\}$. On the other hand, if ϕ stands for a question, there will be multiple ways of supporting the formula, and thus multiple resolutions; for instance, we have $\mathcal{R}(?p) = \{p, \neg p\}$, and $\mathcal{R}(p \mathbin{\mathbb{V}} q) = \{p, q\}$. Any formula in InqB can thus be construed as offering a (possibly trivial) choice among classical formulas.

We saw that entailments among classical formulas amount to entailments in classical logic. On the other hand, we saw that our language also includes formulas which can be regarded as questions. Instances of entailment which involve such formulas capture interesting logical relations that lack a counterpart in classical logic; notably, entailments involving both question assumptions and question conclusions capture relations of *logical dependency* among these questions, possibly within the context of certain statements. For instance, the following entailment captures the fact that, given the information that $r \leftrightarrow p \wedge q$, the question $?r$ is completely determined by the questions $?p$ and $?q$.

$$r \leftrightarrow p \wedge q,\ ?p,\ ?q \ \models ?r$$

Summing up, then, propositional inquisitive logic can be regarded as a conservative extension of classical propositional logic with an inquisitive disjunction: while the classical fragment of the language coincides with classical logic, by means of the operator $\mathbb{V}$ we can build formulas which express propositional questions, and capture dependencies among such questions as special cases of the relation of entailment.[4]

5 Relations with intuitionistic logic

In the previous section, we saw that InqB can be viewed as a conservative extension of classical logic if $\mathbb{V}$ is regarded as an additional, non-standard connective. In this section we will show that if, on the other hand, we regard $\mathbb{V}$ as the standard disjunction of the system,

[4]For more on the relations between inquisitive logic and classical logic, see [6, 8].

then InqB turns out to be a special kind of *intermediate* logic, i.e., a logic sitting in between intuitionistic and classical logic.

The first step in this direction is to notice that our semantics can be regarded as a case of intuitionistic Kripke semantics on a particular Kripke model, having consistent information states as its elements, the relation $\supseteq$ as accessibility relation, and the valuation function $V(p) = \{s \mid w(p) = 1 \text{ for all } w \in s\}$. Since Kripke semantics is sound for intuitionistic logic, this implies that anything that can be falsified in inquisitive logic can be falsified in intuitionistic propositional logic, IPL. On the other hand, it is easy to see that singleton information states $\{w\}$ behave just like the corresponding propositional valuation w: this ensures that anything that can be falsified in classical logic, CPL, can also be falsified in inquisitive logic. If we identify a logic with the corresponding set of validities, we can sum up our findings as follows.

Proposition 3. $\mathsf{IPL} \subseteq \mathsf{InqB} \subseteq \mathsf{CPL}$

Thus, from this perspective InqB is a logic stronger than intuitionistic logic, but weaker than classical logic. It is not, however, an *intermediate logic* in the usual sense of the term. This is because InqB is not closed under the rule of *uniform substitution*: in particular, the double negation law is valid for propositional atoms, but invalid when atoms are replaced by questions: $\neg\neg p \to p \in \mathsf{InqB}$, but $\neg\neg ?p \to ?p \notin \mathsf{InqB}$. The conceptual point here is that atoms in InqB are not intended as placeholders for arbitrary sentences, but only placeholders for arbitrary *statements*. As we saw, statements are truth-conditional, and as such they validate the double negation law, which is not generally valid. It is worth emphasizing that this is not an accident, but a deliberate architectural choice (see pp. 66-67 of [6]). This choice (i) enables InqB to retain a classical fragment, which encodes the underlying logic of statements; (ii) allows for a recursive decomposition of questions into resolutions; and (iii) makes a neat proof system possible.

Besides this classical feature of atoms, inquisitive logic differs from intuitionistic logic in that the space of information states has a special structure, which renders valid some non-intuitionistic principles. The

best known of these is the Kreisel-Putnam scheme, first studied in [14]:

$$(\mathsf{KP}) \qquad (\neg\phi \to \psi \mathbin{\mathbb{V}} \chi) \to (\neg\phi \to \psi) \mathbin{\mathbb{V}} (\neg\phi \to \chi)$$

While this principle may look mysterious at first, it can be shown (see p. 80 of [6]) to encode a fundamental relation between statements and questions: a statement only counts as resolving a question if it entails a specific resolution to the question.

As shown in [9], the classicality of atoms and the validity of the KP scheme, together with the underlying intuitionistic base, suffice to characterize inquisitive propositional logic completely. More formally, InqB can be characterized as the set of formulas obtained by extending IPL with all instances of KP and with $\neg\neg p \to p$ for all atoms p, and closing the resulting set under *modus ponens*.

Theorem 2. $\mathsf{InqB} = \mathsf{IPL} + \mathsf{KP} + \neg\neg p \to p$

In fact, besides the Kreisel-Putnam logic axiomatized by the scheme KP, there is a whole range of intermediate logics which, when extended with classical atoms, yield inquisitive logic: as shown in [9], this range consists exactly of those intermediate logics which include Maksimova's logic [15] and are included in Medvedev's logic of finite problems [17, 18]. In particular, Medvedev's logic is the largest standard intermediate logic included in InqB.

An important aspect of the relation between inquisitive logic and intuitionistic logic can be observed based on the normal form result given by Theorem 1. This result guarantees that any formula can be written as an inquisitive disjunction of classical formulas. Since classical formulas behave as in classical logic, they are logically equivalent to their own double negation. Thus, it follows that in InqB, any formula ϕ is equivalent to an inquisitive disjunction of negations $\phi^{\mathsf{DNT}} = \neg\psi_1 \mathbin{\mathbb{V}} \ldots \mathbin{\mathbb{V}} \neg\psi_n$. Now, the following theorem shows that the map $(\cdot)^{\mathsf{DNT}}$ is a translation of inquisitive logic into intuitionistic logic.

Theorem 3. $\Phi \models \psi \iff \Phi^{\mathsf{DNT}} \models_{\mathsf{IPL}} \psi^{\mathsf{DNT}}$

This result can be extended to show that the Lindenbaum-Tarski algebra for InqB is isomorphic to the sub-algebra of the Lindenbaum-Tarski algebra for IPL consisting of equivalence classes of disjunctions of negations. Thus, while classical propositional logic can be regarded as the negative fragment of intuitionistic logic, propositional inquisitive logic can be regarded as the *disjunctive-negative fragment* of intuitionistic logic—the fragment consisting of disjunctions of negations.[5]

6 Relations with dependence logic

We mentioned above that in inquisitive logic, entailments involving questions capture logical dependencies. The relation of dependency is also the focus of recent work in the framework of *dependence logic* [23]. Dependence logic and inquisitive logic are tightly connected frameworks, as discussed in detail in [7]. In the propositional setting, full translations are possible between the two [25]. In both propositional systems, formulas are interpreted relative to sets of assignments; while propositional inquisitive logic enriches classical propositional logic with questions, propositional dependence logic enriches it with formulas called *dependence atoms*, written $=(p_1, \dots, p_n, q)$, which capture the fact that the truth-value of an atomic proposition q is determined by the truth-values of other atomic propositions $p_1, \dots, p_n$. The semantics of these atoms is given by the following clause:

$$s \models =(p_1, \dots, p_n, q) \iff \forall w, w' \in s : \text{ if } w(p_i) = w'(p_i) \text{ for all } i, \text{ then } w(q) = w'(q)$$

It is easy to check that such a dependence atom can be expressed in InqB by means of the formula $?p_1 \wedge \dots \wedge ?p_n \to ?q$. This is not an accident: as shown in [7], in inquisitive logic, the fact that a question ν is fully determined by questions $\mu_1, \dots, \mu_n$ is generally captured by the implication $\mu_1 \wedge \dots \wedge \mu_n \to \nu$. More precisely, the formula $\mu_1 \wedge \dots \wedge \mu_n \to$

[5] For more on the relations between propositional inquisitive logic, intuitionistic logic, and intermediate logics, see [3] and [9].

ν is supported at a state s in case relative to s, any way of resolving the questions $\mu_1, \ldots, \mu_n$ determines a corresponding way to resolve the question ν. What a dependence atom expresses is that the question $?q$ is determined by the questions $?p_1, \ldots, ?p_n$, hence the representation $?p_1 \wedge \cdots \wedge ?p_n \to ?q$.

Realizing that dependencies can be captured generally as implications between questions is interesting for various reasons. The first kind of reason is proof-theoretic: in inquisitive logic, all the connectives, including those involved in a dependence formula, can be handled by essentially standard inference rules. Thus, for instance, a dependency $?p \to ?q$ may be formally proved to hold by assuming the question $?p$ and trying to conclude the question $?q$. In fact, this perspective brings out the fact that the *Armstrong axioms* for functional dependency [2] used in database theory are essentially nothing but the axioms of implication in disguise—a fact that was first noted in [1].

Moreover, realizing that dependencies can be generally captured as implications between questions allows us to see that dependence atoms are a particular case of a more general pattern. Not just for atomic polar questions of the form $?p$, but for all sorts of questions $\mu_1, \ldots, \mu_n, \nu$ expressible in the system—in fact, in *any* inquisitive system—the fact that ν is determined by $\mu_1, \ldots, \mu_n$ is expressed by $\mu_1, \ldots, \mu_n \to \nu$.

Finally, realizing that dependencies can be expressed as implications among questions allows us to use inquisitive logics to investigate the logical properties of the notion of dependency. For example, consider the valid entailment $?p,\ ?p \wedge ?q \to ?r \models ?q \to ?r$. This captures the fact that given the information whether p, from a dependency of $?r$ on both $?p$ and $?q$ we can always compute a dependency of $?r$ on $?q$. If we think of a dependency as encoded by a function (cf. the notion of *dependence function* in §2 of [6]), this amounts to the fact that we can saturate one of the arguments of this function.[6]

[6] For more on the relations between inquisitive and dependence logic, see [7, 6, 26].

7 Questions in proofs

An important feature of inquisitive logic is that it shows that questions can meaningfully be manipulated in logical inferences, and that their logical behavior is in fact rather familiar. In the propositional setting, a natural deduction system for inquisitive logic is obtained by extending a system for intuitionistic logic with the following two inference rules, where α ranges over classical formulas, and ϕ, ψ over arbitrary formulas.

$$\frac{\alpha \to (\phi \mathbin{\mathbb{V}} \psi)}{(\alpha \to \phi) \mathbin{\mathbb{V}} (\alpha \to \psi)}\ (\mathsf{split}) \qquad \frac{\neg\neg\alpha}{\alpha}\ (\mathsf{dne})$$

The second of these rules captures the fact that classical formulas are truth-conditional, and thus behave exactly as in classical logic. The first—related to the Kreisel-Putnam scheme discussed above—captures the interaction among statements and questions, stipulating that if a statement resolves a question, it must do so by yielding a particular resolution to it. The completeness of this system for InqB, proved in [6], implies in particular that any valid propositional dependency can be formally proved by making inferences with propositional questions in this system. Thus, questions are interesting proof-theoretic tools: by making inferences with them, we can establish the existence of certain logical dependencies. Moreover, the following theorem, proved in [6], shows that a proof of a dependency does not just *witness* that the dependency holds, but actually encodes a method for computing it.

Theorem 4 (Constructive content of inquisitive proofs)**.**
Suppose P is a natural deduction proof having assumptions $\phi_1, \dots, \phi_n$ and conclusion ψ. Recursively on P, we can define a procedure f_P which, when given as input resolutions $\alpha_1, \dots, \alpha_n$ of the assumptions, outputs a resolution $f_P(\alpha_1, \dots, \alpha_n)$ of the conclusion with the property that $\alpha_1, \dots, \alpha_n \models f_P(\alpha_1, \dots, \alpha_n)$.

What this theorem shows is that proofs in inquisitive logic have a specific kind of constructive content: they encode methods for turning any given resolutions of the question assumptions into a resolution

of the conclusion which is determined by them. This is reminiscent of the proofs-as-programs interpretation of intuitionstic logic, and it shows once more that, while our logic coincides with classical logic on statements, encoded by classical formulas, the logic of questions has a constructive flavor to it.[7]

8 Extensions and generalization

In the last couple of years, the work on propositional inquisitive logic presented in the previous sections has been extended in several directions. First of all, it has been taken as the basis for logics that go beyond the propositional realm, such as the modal logics given in [4, 10, 6], and the first-order logics given in [6, 7]. Presenting these richer logics goes beyond the scope of the present survey. However, in this section I want to briefly discuss an extension and a generalization of InqB, both due to Vít Punčochář, that remain within the domain of propositional logic.

First, the system InqB is extended in [20] with a *weak negation* connective, denoted $\sim$, which allows us to express the fact that a certain formula fails to be supported at the evaluation state.

$$s \models \sim\phi \iff s \not\models \phi$$

Evidently, the addition of this connective results in a system in which support is no longer persistent: a formula $\sim\phi$ may be supported by a state s, yet it may fail to be supported by a stronger state $t \subseteq s$. One reason why such a system is interesting is that—while remaining within the propositional inquisitive setting—it allows for the definition of formulas $\Diamond\phi$ which express the fact that the state of evaluation can be extended consistently to support ϕ. Interestingly, this logic is axiomatized by means of a proof system which allows for two different modes of hypothetical proofs. In one mode, making the assumption ϕ corresponds to supposing that the current information state supports ϕ. In

[7] For more on the role of questions in inference and on the constructive content of inquisitive proofs, see [8, 6, 5].

the other mode, it corresponds to supposing that the current information state is extended so as to support ϕ. In this second mode, only some formulas from outside the hypothetical context can be appealed to when reasoning within the hypothetical context.

A generalization of propositional inquisitive logic is explored in [21]. This paper defines an operation G which, given a logic Λ with $\mathsf{IPL} \subseteq \Lambda \subseteq \mathsf{CPL}$, returns a corresponding logic $\mathrm{G}(\Lambda)$, called the *global variant* of Λ. Logics of the form $\mathrm{G}(\Lambda)$ are called G-logics.[8] Intuitively, $\mathrm{G}(\Lambda)$ is a logic obtained by extending the $\mathbb{V}$-free fragment of Λ with an inquisitive disjunction connective. In Section 4, we saw that InqB can be seen as arising from extending classical logic with inquisitive disjunction. And indeed, we have $\mathsf{InqB} = \mathrm{G}(\mathsf{CPL})$, which means that inquisitive logic is the greatest of all G-logics. The smallest G-logic, $\mathrm{G}(\mathsf{IPL})$, is the logic $\mathsf{IPL}{+}\mathsf{H}$ axiomatized by extending intuitionistic logic with the following scheme, where ϕ, ψ range over arbitrary formulas, and α ranges over Harrop formulas, and closing under *modus ponens*.[9]

$$(\mathsf{H}) \qquad (\alpha \to \phi \mathbb{V} \psi) \to (\alpha \to \phi) \mathbb{V} (\alpha \to \psi)$$

All other G-logics fall in between $\mathsf{IPL} + \mathsf{H}$ and InqB, and share many of the core features of inquisitive logic. All of them have the disjunction property, meaning that a disjunction $\phi \mathbb{V} \psi$ can only be valid if either ϕ or ψ is valid. None of them is closed under uniform substitution. All of them coincide with the base logic Λ in their $\mathbb{V}$-free fragment, and allow for an analogue of Theorem 1, stating that any formula is equivalent to a disjunction $\alpha_1 \mathbb{V} \ldots \mathbb{V} \alpha_n$ of classical formulas. Finally, all G-logics can be characterized axiomatically in a uniform way: $\mathrm{G}(\Lambda)$ amounts to the logic obtained by extending intuitionistic logic with the scheme H and all $\mathbb{V}$-free formulas which are valid in Λ, and closing this set under *modus ponens*.

[8] Here, the logic Λ is assumed to be closed under *modus ponens*, but not necessarily under uniform substitution.

[9] A Harrop formula is defined as a formula in which disjunction is only allowed to occur within the antecedent of an implication.

9 Conclusion

In this paper I have tried to give a bird's eye view of propositional inquisitive logic, including its conceptual underpinnings, its main mathematical features, and its relations to other logics. My hope is that this survey, together with the pointers scattered through the paper, will provide a valuable guide to the growing literature on the subject.

Acknowledgments. Financial support from the Netherlands Organization for Scientific Research (NWO) is gratefully acknowledged.

References

[1] S. Abramsky, J. Väänänen. *From IF to BI.* Synthese, 167(2): 207–230 (2009).

[2] W. Armstrong. *Dependency structures of data base relationships.* In Proceedings of the IFIP Congress (1974).

[3] I. Ciardelli. *Inquisitive semantics and intermediate logics.* MSc Thesis, University of Amsterdam (2009).

[4] I. Ciardelli. *Modalities in the realm of questions: axiomatizing inquisitive epistemic logic.* Advances in Modal Logic 10, R. Goré, B. Kooi, A. Kurucz, eds., pp. 94–113, College Publications (2014).

[5] I. Ciardelli. *Interrogative dependencies and the constructive content of inquisitive proofs.* In Logic, Language, Information and Computation. Proceedings of WoLLIC 2014, U. Kohlenbach, P. Barceló, and R. de Queiroz, eds., pp. 109–123. Lecture Notes in Computer Science (2014).

[6] I. Ciardelli. *Questions in Logic.* PhD Thesis, University of Amsterdam (2016).

[7] I. Ciardelli. *Dependency as Question Entailment.* Dependence Logic: theory and applications, S. Abramsky, J. Kontinen, J. Väänänen and H. Vollmer, eds., Springer (2016).

[8] I. Ciardelli. *Questions as Information Types.* Manuscript, under review (2016).

[9] I. Ciardelli, F. Roelofsen. *Inquisitive Logic.* Journal of Philosophical Logic, 40(1):55–94 (2011).

[10] I. Ciardelli, F. Roelofsen. *Inquisitive Dynamic Epistemic Logic.* Synthese, 192(6):1643–1687 (2015).

[11] I. Ciardelli, J. Groenendijk, and F. Roelofsen. *Inquisitive semantics: a new notion of meaning.* Language and Linguistics Compass, 7(9):459–476 (2013).

[12] J. Groenendijk. *The logic of interrogation.* In T. Matthews and D. Strolovitch eds., Semantics and Linguistic Theory, pp. 109–126 (1999).

[13] J. Groenendijk. *Inquisitive semantics: Two possibilities for disjunction.* In Peter Bosch, David Gabelaia, and Jérome Lang, eds., Seventh International Tbilisi Symposium on Language, Logic, and Computation (2009).

[14] G. Kreisel, H. Putnam. *Eine Unableitbarkeitsbeweismethode für den intuitionistischen Aussagenkalkül.* Archiv für Mathematische Logik und Grundlagenforschung, 3, pp. 74–78 (1957).

[15] L. Maksimova. *On maximal intermediate logics with the disjunction property.* Studia Logica, 45, pp. 69–75 (1986).

[16] S. Mascarenhas. *Inquisitive semantics and logic.* MSc Thesis, University of Amsterdam (2009).

[17] J. T. Medvedev. *Finite problems.* Soviet Mathematics Doklady, 3, 227–230 (1962).

[18] J. T. Medvedev. *Interpretation of logical formulas by means of finite problems.* Soviet Mathematics Doklady, 7, 857–860 (1966).

[19] F. Roelofsen. *Algebraic foundations for the semantic treatment of inquisitive content.* Synthese, 190(1):79–102 (2013).

[20] V. Punčochář. *Weak negation in inquisitive semantics.* Journal of Logic, Language and Information, 23, 47–59 (2015).

[21] V. Punčochář. *A generalization of inquisitive semantics.* Journal of Philosophical Logic, DOI:10.1007/s10992-015-9379-1 (2015).

[22] V. Punčochář. *A new semantic framework for modal logic.* Philosophical Alternatives, 23:47–59 (2014).

[23] J. Väänänen. *Dependence Logic: A New Approach to Independence Friendly Logic.* Cambridge University Press (2007).

[24] J. Väänänen. *Modal dependence logic.* In K.R. Apt and R. van Rooij, editors, New Perspectives on Games and Interaction. Amsterdam University Press (2008).

[25] F. Yang. *On extensions and variants of dependence logic: a study of intuitionistic connectives in the team semantics setting.* PhD thesis, University of Helsinki (2014).

[26] F. Yang, J. Väänänen. *Propositional logics of dependence.* Annals of Pure and Applied Logic, 167(7), 557–589 (2016).

Ivano Ciardelli

[1]ILLC, University of Amsterdam
Email: i.a.ciardelli@uva.nl

Natural Language Processing versus Logic. Pros and cons on the dispute whether logic is useful in the computational interpretation of language

Dan Cristea

Abstract

In this essay I express some personal opinions regarding the influence that logic has on modern approaches to process natural language in artificial systems. I start by presenting some successful linguistic formalisms that were originated in logic, arguing why logic is important in conveying the meaning of language expression. Then, I counterbalance the argumentation with a number of examples where logic is impuissant to mirror language usage, finally supporting a rather temperate opinion about the usefulness of logic to formalise low level linguistic processes and about the limits of language formalisation.

Keywords: natural language processing (NLP), logic theories in NLP, statistical approaches, symbolic versus statistical approaches in NLP.

1 Introduction

It seems that we, human beings, motored by the need to understand the reality among us (a condition for survival), make use of shallower or deeper cognitive processes in our efforts to assign meanings to messages we receive through language. This cognitive behaviour resembles, in some cases, logical formalisms (what is logic if not an expression of thinking?). This is just as we have in our brains a symbolic machinery

capable to help us make inferences and offer solutions to complex puzzles that the language encodes. In other cases, however, logic is of no use: language manifests a totally weird behaviour.

The need for symbolic approaches, as opposed to statistical ones, is in itself an expression of the feelings researchers have that language can be expressed as a system of rules, which would make its messages to be "computable". This paradigm is similar to the one of mathematical logic, since there too, based on a very clear notation of a number of basic ingredients and of defined ways in which they can be grouped together, truth values of sentences (i.e. syntactical constructions) can be deduced.

The debate here, is not if logical, i.e. symbolic, formalisms are useful, but to what extend can they be applied to explain a range of linguistic phenomena. Then, for the whole rest of linguistic phenomena for which logic fails to offer support, ought we instead resort to statistical or neural-based solutions? Or part of our manifestations triggered by language escapes from any formalisation, being it based on logic, statistics or neural grounds? And finally, what is the range of practical applications of language which put at their base logical solutions?

In this essay I express some personal opinions regarding the influence that logic has on modern approaches to processing natural language.

2 When do we need logic to decipher language?

Language ought to be logical, or, else, the communication based on it would be impossible (imagine that the inscription of messages would be made in a randomizing system of signs…). In most of cases, people are able to transmit their intentions correctly to the intended receivers. So, when we say *two horses* we mean something of the kind: $\exists\ S = \{x \mid \text{horse}(x)\}$ ^ $\text{card}(S) = 2$, where $\text{card}(S)$ means the cardinal of the set S, $\text{horse}(x)$ is a qualifying function asserting that x, its argument, has a semantic property that is shared in the common knowledge of the living inhabitants of this world, that this property is currently denominated in English by the word *horse*, and that the pragmatic context in which the expression is uttered clearly separates the different meanings that this English word may encode. In the same time, uttering this noun phrase, we are also aware that the listener possesses an equivalent decoding mechanism that enables her to coagulate an equivalent meaning. So, it

seems that in extremely many situations of the real life, when we use language, we implicitly resort to mathematical logic to express our messages. In fact, although we don't really do that, it is just like doing that, i.e. just like somebody above us, listening to what we are saying, quickly encodes our saying in a logical expression that could unambiguously be "read" or decoded by our partner in the conversation.

The challenge to describe language in a logical system of notation that would allow non-ambiguous representations of its lexical elements and coherent composition/decomposition has preoccupied modern linguistics for a long time already. In this section I will make a very quick survey of only some of these approaches.

2.1 Generative Lexicon and Qualia Structures

This kind of attitude towards language brings forward Generative Lexicon (GL) [6], a theory that intends to build lexical and semantic resources capable of expressing in computational terms (which is another name for logic) the rich lexical variety of the language (any language, in principle), including its capacity to combine meanings of lexical items through grammar and, to a certain extend, through pragmatics. GL tries to describe the semantic flexibility shown by words in combination with others. To account for diverse interpretations that words can display when placed in combinations with others, GL associates a hidden event in the lexicon description of nouns, adjectives and adverbs. Originated in the Aristotelian concept of *aitia* (explanation), with re-interpretations added by Moravcsik [4], further developments of GL [5, 7] introduce Qualia structures, describing roles according to which the meaning of words can be decomposed on four different coordinates:

- *Formal* (F): encoding taxonomic information about the lexical item (the *is-a* relation);
- *Constitutive* (C): encoding information on the parts and constitution of an object (*part-of* or *made-of* relation);
- *Telic* (T): encoding information on purpose and function (the *used-for* or *functions-as* relation);
- *Agentive* (A): encoding information about the origin of the object (the *created-by* relation).

Qualia are formally represented as typed feature structures. For instance, the one in Fig. 1 can account for combinations such as: *large car* (F =

vehicle), *broken car* (C = motor), *speedy car* (T = drive), *Italian car* (A = made in Italy). Also, adding arguments to roles, Qualia structures can deal with metonymy, as in: *the car from behind honked*: T = drive (human, vehicle). Such lexicon representations actually encode in a logical form part of an ontology of lexicalised concepts, out of which "understanding" can be computed.

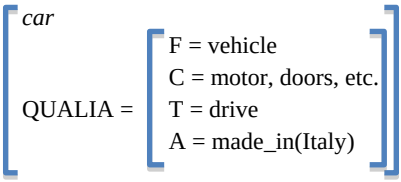


Figure 1. A Qualia structure for the lexical concept car.

Qualia structures are recognised to have various shortcomings. Pustejosky and Jezek [7], for instance, recognise the rather limited "ability [of the formalism] to take on an indefinite variety of possible senses depending on the other words they combine with". He gives the example of the verb *like* and wonders whether it has two different meanings in "He likes my sister" and "He likes vanilla ice cream", and if so, how is this difference to be represented in decompositional terms?

2.2 Discourse coherence

But logic is needed at upper levels of language interpretation as well. One example is the need to consolidate meanings of sequences of utterances in discourse. Compare, for instance:

Maria dropped the egg from her hand. (1.1)
She cleaned the floor. (1.2)

with:

Maria dropped the feather from her hand. (2.1)
She cleaned the floor. (2.2)

While sequence (1) is perfectly coherent, sequence (2) apparently has no meaning, although the utterances of each sequence convey unambiguous meanings. It is clear that an inferential chain of deductions, triggered by common sense (or ontological) knowledge, link the two utterances in (1), as opposed to (2), where the connection is much harder to establish. The

meaning in (1) is built out of a reasoning sequence showing a temporal occurrence of events that could be schematized as follows:

(1.1) => drops(AG:Maria, OB:egg) => falls(REC:egg) => touches(AG:egg, OB:*X*)
(3)
Qualia.Formal(egg) = container; Qualia.Constitutive(egg) = {eggshell(fragile),liquid}
(4)
(3), (4) => breaks(REC:eggshell) => leaks(REC:liquid, ON:*X*) => perceives(AG:Maria, OB:dirty(*X*))
(5)
(1.2) => cleans(AG:Maria, OB:floor) => perceives(AG:Maria, OB:dirty(floor))
(6)

The equality of (5) and (6), when *X* equals floor, closes the inference chain, proving the high degree of coherence of the sequence (1). A longer inference chain (if any), implying volitional searches in a space of possibilities fuelled by imagination, as opposed to the first case, in which the inferences are common-sense, natural, spontaneous, could, in the mind of an intrigued reader, possibly link the utterances in (2), thus showing a much lower degree of coherence.

2.3 Textual Entailment

In the fight to decipher the meaning expressed in language, two contrary phenomena have to be faced: variability and ambiguity. Variability of language means that the same meaning can be verbalized in different surface forms. Ambiguity means that one surface form can be interpreted as having different meanings. A number of NLP applications that deal with the variability of language trying to reduce the distance between form and meaning are: Information Retrieval (IE), Textual Entailment (TE) and Question Answering.

In TE, it is said that **text *t* entails hypothesis *h* ($t \Rightarrow h$) if humans reading *t* will infer that *h* is most likely true**. So, textual entailment is a directional relation between two texts. In practical applications of TE

(including competitions[1]) *t* could be complemented with external knowledge in order for *h* to be entailed, but *h* cannot be entailed only by the knowledge itself (for instance, by searching on the web).

Here is an example of a true entailment (from RTE data):

t: *…a shootout at the Guadalajara airport in May, 1993, that killed Cardinal Juan Jesus Posadas Ocampo and six others.*
h: *Cardinal Juan Jesus Posadas Ocampo died in 1993.* (7)

and of a false one (same source):

t: *Regan attended a ceremony in Washington to commemorate the landings in Normandy.*
h: *Washington is located in Normandy.* (8)

One of the methods used to measure the similarity between *t* and *h* does syntactic matching or transformations at the syntactic level. To see the complexity of such a tentative, I will examine in some detail an example. Suppose *t* is:

Philanthropic Golding Inc. came into existence in January 2004. (9.1)
One year after its foundation the company declared bankruptcy. (9.2)

and *h*:

Philanthropic Golding Inc. bankrupted in January 2005. (10)

One way to check the validity of such an entailment, is to launch a pipeline of processes, at the end of which the sentences of both *t* and *h* are expressed in a symbolic form that allows close comparison. Applied to (9.1) the pipeline produces the following successive results (simplified)[2]:

[1] For instance, the EU FP-6 Funded PASCAL Network of Excellence 2004-7: Recognizing Textual Entailment (RTE) Challenges.

[2] Here we use an XML coding, but a representation that uses RTF tuples or another notation convension can also be employed.

Step1: tokenisation (not shown), part-of-speech tagging (not shown), chunking noun phrases and clashing multi-word expressions.
<NP id="n1">*Philanthropic Golding Inc.*</NP>
<MWE id="m1">*came into existence*</MWE>
Step 2: recognition of entity mentions, of time expressions and resolution of anaphora.
<COREF-LIST id="ent1" TYPE = "ENTITY" REF-LIST="n1" />[3]
<TIMEX3 tid="t1" type="DATE" value="2004-01">*January 2004*</TIMEX3>
Step 3: functional dependency parsing; here we show a Universal Dependency (UD) coding [3].

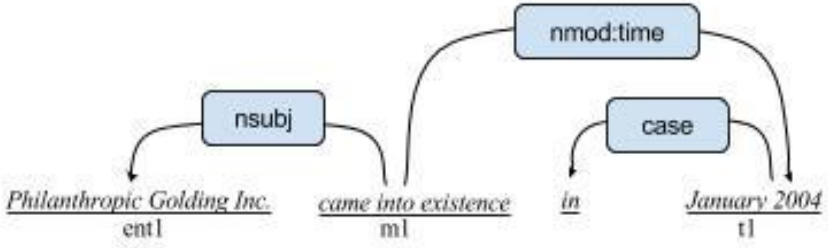


Figure 2. UD parsing of sentence (9.1).

Step 4: time analysis, in which EVENT and TLINK elements, formalizing the events and their temporal relations, are generated[4].
<EVENT eid="ev1" VB="m1" AG="ent1"/>
<TLINK eventID="ev1" relatedToTime="t1"
relType="BEGINS"/>
Step 5: generation of equivalent structures (transformations) and the application of the closure tool, which computes the transitive closure of temporal relations; the transformation here concerns the equivalence of

[3] A COREF-LIST with only one member signals the first mention of an ENTITY or EVENT, according to TYPE.
[4] To simplify notations, MAKEINSTANCE and SIGNAL elements are ignored and EVENT elements are complemented with roles.

the expressions: *X comes into existence* and *UNKNOWN founds X*. This rule triggers the element:
<EVENT eid=”ev2” VB=”found” AG=”UNKNOWN” OB=”ent1”/>
and its correspondent time link:
<TLINK eventID="ev2" relatedToTime="t1"
relType="BEGINS"/>
And now the pipeline applied to (9.2):
Step 1:
<NP id=”n3”><NP id=”n2”>*its*</NP>*foundation*</NP>
<NP id=”n4”>*the company*</NP>
<NP id=”n5”>*bankruptcy*</NP>
Step 2:
<TIMEX3 tid=”t2” type=”DURATION” value=”P1Y”>*one year*</TIMEX3>
<COREF-LIST id=”ent1” TYPE=”ENTITY” REF-LIST=”n1 n2 n4” />
<COREF-LIST id=”eve1” TYPE=”EVENT” REF-LIST=”ev1 ev2 n3” />
<COREF-LIST id=”eve2” TYPE=”EVENT” REF-LIST=”n5” />
Step 3:

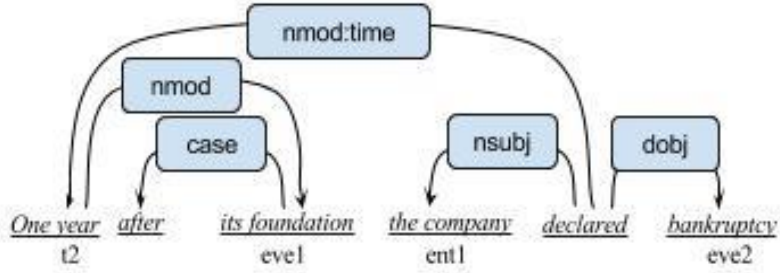


Figure 3. UD parsing of sentence (9.2).

Step 4:
<EVENT eid=”e3” POS=”VERB” CLASS=”REPORTING” AG=”ent1” OB=”eve2”>*declared*</EVENT>
<EVENT eid=”e4” POS=”NOUN” CLASS=”OCCURRENCE” AG=”ent1”> *bankruptcy*</EVENT>
<TLINK eventID="e3" relatedToTime="t2" relType="AFTER"/>

<SLINK eventID="e3" subordinatedEvent="e4"
relType="FACTIVE"/>

Step 5: the closure tool produces:
<TIMEX3 tid="t4" type="DATE" value="2005-01-xx" />
<TLINK eventID="e4" relatedToTime="t4"
relType="DURING"/>
(11)

A similar processing pipeline applied to (10) should yield:
<EVENT eid="e5" POS="VERB" AG="ent1">*bankrupted*</
EVENT>
<TIMEX3 tid="t5" type="DATE" value="2005-01-xx">*January 2005*</TIMEX3>
<TLINK eventID="e5" relatedToTime="t5"
relType="DURING"/> (12)

And the equivalence of (11) and (12) proves the entailment.

2.4 Other NLP formalisms rooted on logic

Prolog, the programming language of logic, has inspired much work on NLP. In syntax, this means to express a grammar as a set of statements in a logic formalism (e.g. Horn clauses), and to use a theorem prover (e.g. resolution) in order to parse or generate sentences. Recently used in information extraction, SHERLOCK [8] is a system able to learn Horn clauses in a large-scale, domain independent manner from Web texts. The learned rules can then be used to fuel a first-order reasoning system, as HOLMES, described by Schoenmackers *et al.* [9], which infers answers from tuples.

3 And when logic is of no use?

Languages have specific ways to express linguistic phenomena. Some of them seem to escape any logical explanations.

3.1 Double negation

In propositional logic the double negation is equivalent to an affirmation. However, applied to language, this rule does not always hold. In

connection to this phenomenon, Falaus [2] inventories two main types of languages. In Double Negation languages, among which standard varieties of Germanic and Scandinavian, two negative elements cancel each other out resulting in a positive reading, as in (13) below:

Paul didn't see nobody. = Paul saw somebody. (13)

However, in Negative Concord languages, among which Romanian and Italian, multiple occurrences of negation are interpreted as one semantic negation, as in (14):

Paul n-a văzut pe nimeni. = Paul didn't see anybody. (14)

The sentence can be paraphrased as "It is not the case that there is an individual *x*, such that Paul saw *x*."

3.2 Linear position

Romanian is known to be ambivalent with respect to the position of quality adjectives around the nouns they modify: they may occur pre-nominally as well as post-nominally. Cornilescu [1] notices that certain associations of noun+adjective versus adjective+noun makes a difference of interpretation, as here:

femeia singură => the woman alone
singura femeie => the only woman (15)

I believe that the following examples display similar behaviour. Suppose somebody has two cars, one bought some time ago and one recently bought, and the one recently bought belongs to an old brand while the one bough in the past belongs to a newer brand. Then:

maşina lui cea veche refers to *his old brand car*, while
vechea lui maşină refers to *the car owned by him for a long time* (16)

However, in the following associations the sense does not change:

domnişoara frumoasă, frumoasa domnişoară => the beautiful young lady
cartea interesantă, interesanta carte => the interesting book (17)

Also the positional ambivalence does not apply to any adjective. Certain modifiers make sense only when situated in the pre-position with respect to the modified noun. For example, *biet* (*poor*, *pitiful*) is not accepted unless it precedes the noun: *biet om* (*poor man*), but not: *om biet*. Vulchanova [10] explains this for Balkan languages: these associations seem to contradict the usual intersection-based composition. In general, if *X* is an adjective and *Y* – a noun, then *X Y* (or *Y X*) means the set of objects *Y* that have the property *X*, or the intersection between the set of objects

having the property *X* and the set of objects *Y*. As such, *poor men* should be taken as the intersection between the set of things which are poor and the set of men, but *bieţii oameni* means something different than the subset of the set of men which are poor, it means a subset of the set of men which are in a pitiful/miserable state.

3.3 Contexts and the mist of pragmatics

It is a truism that the context determines the meaning of words, and the previous section showed some examples. By "context" here I mean both textual (i.e. positional) and not textual (for instance, temporal).

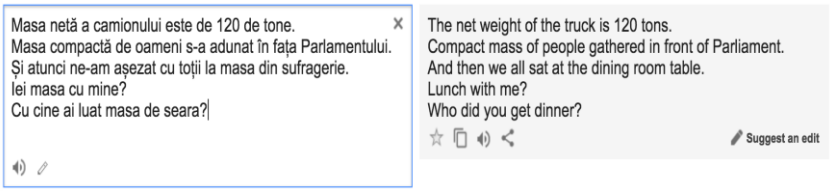


Figure 4. Google Translate solutions for difference occurrences of the Romanian word "masa".

The left side of Fig. 4 shows different contexts of occurrence of the Romanian word "masa". As with any other poli-semantic words, its sense is fixed by the context. Google Translate applies statistics to disambiguate and, as can be seen, remarkably well. For the time being, I cannot imagine a workable logical solution to this issue, and if this would ever be achieved, what would be the cost of the supportive lexico-semantic resources?

But it is also clear that words induce different reactions in humans, depending of their culture, the moment of the utterance and any other pragmatic conjuncture. A notorious example is the wood language. I wonder how would a logical approach detect the humorous effect that is conveyed by phrases such as the following, and the reader can find more examples in [11]:

"Obiectivele majore, de însemnătate cu adevărat istorică, pe care Partidul Comunist Român le-a încredinţat frontului culturii româneşti în perioada

făuririi societăţii socialiste multilateral dezvoltate angajează – deschis şi plenar…"[5] (approximately: *The major objectives, of truly historic significance, which the Romanian Communist Party has entrusted to the frontline of the Romanian culture in the making of the multilaterally developed socialist society – openly and fully…*) (18)

"Pus astfel în lumină, ancorat în sinergia faptelor, recursul la universalitate nu eludează meandrele concretului."[6] (approximately: *Thus put in light, anchored in the synergy of its facts, the appeal to universality does not circumvent the meanders of the concrete.*) (19)

"Să luptăm pentru propăţirea neamului şi aducerea României pe cele mai înalte culmi de civilizaţie multilateral dezvoltată." (approximately: *Let's fight to thrive our stirps and bring Romania on the highest peaks of multilaterally developed civilization.*) (20)

3.4 Style in literature

Humans perceive co-occurrence of words as producing very suggestive images. Confronted with the extraordinary diversity of suggestion that words can convey, logic seems to me faint, forceless, impuissant. How could poetical expressions, such as the following:

"constelaşia ochilor mei" (*the constellation of my eyes*), "atingi cu auzul" (approximately: *your hearing touches*), "nisipuri de fiară" (*beast sands*) – Nichita Stănescu, *Autoportret în timp de veghe* (Auto portrait during watch time) (21)

be encoded in logical constructions? Or how could emotions incurred in sentences like the following one be seized in logical expressions?:

---

[5] Florian Georgescu (1982) Muzeul de istorie — factor dinamic in aplicarea politicii P.C.R. de educare patriotică a maselor, de formare a conştiinţei socialiste, în Sesiunea ştiinţifică de comunicări: Permanenţă, unitate şi progress în istoria poporului Român. Partidul Comunist Român la a 60-a aniversare. Muzeul Naţional, VI. www.mnir.ro

[6] attributed to Ion Iliescu

"It's enough for me to be sure that you and I exist at this moment." – Garcia Marquez: *One Hundred Years of Solitude* (22)

I agree that an effort to formalise in logical terms a metonymic sense of an expression, as in this magnificent sequence of simple words:

"lipindu-se de răcoarea tocului uşii" (approximately: *sticking to the chil of the door frame*) — Garcia Marquez: *One Hundred Years of Solitude* (23)

in which a touched object is replaced with a sensation that the agent borrows from that object is a challenging task. Same happens when looking for logical equivalent of metaphorical language, as here:

"Te mângâi cu degetele muiate în amintiri." (*I caress you with my fingers dipped in memories.*). (24)

4 And the solution is? (instead of conclusions)

The last examples I have given address the philosophical question of whether it is worth looking for a formalisation of language able to encode all its extreme diversity of expressing power. Supposing logic proves to be successful in representing some language aspects, there should be a limit where the ambition to express natural language in logical form has to stop because it reaches an insurmountable limit.

Some people, including me, think that since humans use language all the time, but only rarely make explicit use of logic in their lives, the domain of NLP should not necessarily be dependent on logical formalisms. Their lack of confidence in logic as a universal machinery for processing language comes from observations of the inability of logic to support exhaustively the processing infrastructures of language. In their opinions, there are aspects of language production and understanding for which something else than logic should be used in order to explain and reproduce on the machine these human performances. On the other hand, these people agree that logic is necessary for acquiring certain types of representations. What they don't believe is that each sentence, or each

sequence of sentences, should be transformed into a theorem that necessitates a proof.

It has be clear by now that I am not talking here about reasoning, as the one needed to make volitional connections, to find explicit links, without which understanding would be impossible, as the ones exemplified in sections 2.2 and 2.3, where, clearly, logic is on the first plan, but about the primary processes that enable the use of language as a communication channel, therefore that allow cognition based on language.

However, can be said that the whole domain of formal linguistics is inspired by logical formalisms. Indeed, why representing NL sentences as trees? Because, doing this, we intrinsically incorporate decisions about their ambiguities, this way preparing the path towards semantic representations and reasoning. A logical system applied to language means to interpret signs of the language, i.e. words, in their surface variation, as dictated by morphology, then their sequences, as dictated by syntax, and their meanings, as dictated by semantics, with the goal of arriving to an overall formal and unambiguous representation. Above the sentence boundaries, the sentential representations would be combined in discourse trees (sometimes graphs), on which rhetorical deductions could be made and inter-sentential links, as those implied by anaphorae, would be fulfilled, or extra-textual connections, as those evoked by named entities in the cultural background of the receiver of the message, would be activated. If sufficiently sensitive antennae oriented towards the external world would also be available, pragmatic contexts could make subtle revisions to these representations.

But is this enough? Suppose the day D has come when all computational theories now evolving in the field of language with the aim to decipher and represent language **in symbolic form** would reach a successful finalization, and a sufficiently rich collection of accompanying resources, necessary to support with data these formalizations, would be acquired. Are we done with the interpretation of language in that day? Can we install this tremendous computational machinery on a high performance computer or on an whatever network of cloud interconnected devices and say: from now on, whatever text we read (we, the humans), this Goliath super-computer can also read and it will get similar reactions (of course, inventoried in a very rich annotation language)?

First, let's notice that many successful NLP applications already exist, that are so dumb in a real "interpretation" of the language that would horripilate a "bad" classical linguist. Among them: machine translation. The Google Translate machine, exemplified in Fig. 4, does not "understand" a iota, in the classical sense. Put to represent the meaning of those sentences, it will be incapable. And yet, it deals so well with those sentences: the result is equivalent to that obtained by a human being graduated in Romanian-English translation.

In these approaches, of a purely statistical nature, the performance to translate any source language text into any other target language can be obtained by "compiling" a very large collection of parallel documents and a very large collection of target language documents, out of which huge tables of figures, called language models, are extracted. This computer performance in fact copies the human ability to learn a language by practising it, instead of using grammar books and drill exercises that formally incorporate the language competence.

Then, going a little bit further, we may think that a similar statistical apparatus could be used in a dialog system, that would support a human-machine dialogue, resembling intelligence (as in a Turing's test). Again, a machine, statistically trained to answer questions, could arrive to a similar level of performance as a humanly incorporated call-centre operator. Also, close to this, many models of now-a-days chat-bots make use of purely statistical solutions.

However, we should not exaggerate with congratulations and compliments. Until now, statistical solutions proved to behave well in applications where of interest is the conveying of meaning more than the stylistic expression, the clear message more than the poetical language that adorns the message with subliminal adds, in general there where language refinements that imply more than pure transmission of information, those that touch the domain of literature and art, are not involved. But, although still far away from any acceptable solutions, challenged to approach this side of language, my conviction is that still statistical methods (and neural) have more chances than rule-based ones.

A vivid area of research in NLP is called Sentiment Analysis. The type of applications belonging to this domain addresses the interests that big commercial companies have to interpret opinions from their clients

involving their products, in order to improve them or to estimate future trends. Big data methods put to work on text files are being employed successfully here. Still, a sentiment means much more than the mere and overtly expression of a taste or inclination. For instance, actual systems would perhaps categorise as positive a sentence like *I love you.* and as neutral one that says *I see and I smell everything around us differently since I met you.*

I am confident that machines will arrive to interpret texts, in the sense of extracting the information contained in them (if I would not believe in this success, why bothering to remain in the field?...). Moreover, more and more complex applications that put the interpretation of language at their very base will be on the market. However, I am rather reserved on the usefulness of pure logical approaches in practical NLP settings, one important reason for this being the difficulty of fuelling these systems with the amount of resources that are needed to support the complex reasoning processes. On the contrary, mixed approaches, that maculate the purity of logical approaches with statistics and/or neural models have a much greater chance of being successful. But my optimism dilutes significantly if the border of semantic content interpretation is overpassed, and we will dig our feet on touching more subtle aspects of language, those that involve emotion sourced in language and interpretation of artistic style, therefore those that address the genuine and inspired juxtaposition of words.

References

[1] A. Cornilescu. *The linearization of the Romanian adjectives and the structure of the Romanian DP*, in Torck D. and Wetzles L. (eds), Romance Languages and Linguistic Theory, 2006", Amsterdam: John Benjamins, (2009), pp. 45-68.

[2] A. Falaus. *Romanian N-words as Negative Quantifiers*, Working Papers in Linguistics, Proceedings of the 31st Annual Penn Linguistics Colloquium, University of Pennsylvania, Volume 14, Issue 1 (2008).

[3] M.-C. de Marneffe, C. D. Manning. *Stanford typed dependencies manual*. (2008). http://nlp.stanford.edu/software/dependencies_manual.pdf

[4] J. M. Moravcsik. *Aitia as Generative Factor in Aristotle's Philosophy*, Dialogue, 14:622-36 (1975).

[5] J. Pustejovsky. *The Generative Lexicon*, Computational Linguistics, 17:409-441 (1991).

[6] J. Pustejovsky. *The Generative Lexicon*. Cambridge, MA: MIT Press (1995).

[7] J. Pustejovsky, E. Jezek. *Integrating Generative Lexicon and Lexical Semantic Resources*, LREC Tutorials, May 23 (2016). http://lrec2016.lrec-conf.org/media/filer_public/2016/05/10/tutorialmaterial_pustejovsky.pdf

[8] S. Schoenmackers, J. Davis, O. Etzioni, D. Weld. *Learning First-Order Horn Clauses from Web Text*, Proceedings of EMNLP (2010).

[9] S. Schoenmackers, O. Etzioni, and D. Weld. *Scaling Textual Inference to the Web*. In Procs. of EMNLP (2008).

[10] M.-D. Vulchanova. *Modification in the Balkan nominal expression: An account of the (A)NA – AN(A) order contract*. In Martine Coene and Yves D'hulst (eds): From NP to DP: The syntax and semantics of noun phrases, John Benjamin Publishing Company, Amsterdam/Philadelphia (2003), pp. 91-118.

[11] R. Zafiu. *Limba de lemn*. În Dilema veche, Nr. 314 / 18-24 februarie (2010).

Dan Cristea

"Alexandru Ioan Cuza" University of Iași, Faculty of Computer Science and Romanian Academy, Institute for Computer Science,
E-mail: dcristea@info.uaic.ro

Proceedings of the Conference on Mathematical Foundations of Informatics
MFOI2016, July 25-29, 2016, Chisinau, Republic of Moldova

An Indian Logic of Property and Location

Eberhard Guhe

Abstract

Late Nyāya and Navya-Nyāya are renowned for their affinity to Western formal logic. In their attempt to develop a kind of "logic of property and location" (Matilal) Navya-Naiyāyikas came up with interesting new ideas concerning the interaction between logic and ontology. Some logical inquiries about properties in Navya-Nyāya are related to similar problems in set theory, although the Navya-Nyāya concept of property should not be confounded with the Western concept of set. In the present exposition of Navya-Nyāya logic we outline a property-theoretic framework for a formal reconstruction and demonstrate its utility by referring to some pertinent examples, namely the Navya-Naiyāyikas' operations applied to properties and relations, their discovery of theorems related to these operations and their account of the reference of number words.

Keywords: Indian logic, Navya-Nyāya, property theories, non-well-foundedness, Bealer.

1 Epistemological and ontological presuppositions

According to B. K. Matilal the logic of Navya-Nyāya is a kind of property-location-logic (cf. [16], p. 112). Maheśa Chandra defines the concept of property (*dharma*) in the following way: *dhriyate tiṣṭhati vartate yaḥ sa dharmaḥ. . . . yatra yo vartate sa tasya dharmaḥ.* ([4], p. 8, 9f = [10], p. 60) – "What is fixed [somewhere], depends [on something or] is resident [somewhere], that is a property. . . . What resides somewhere that is the property of that something."

Navya-Naiyāyikas regard a property and its locus (*adhikaraṇa*) as the ultimate constituents of a cognitive event (*jñāna*). "A *jñāna* is a particular just as a color spot or tone is a particular. It can very well be viewed as an *event* in the sense that a particular tone or sound can be viewed as a physical event. (...) Furthermore, a *jñāna* is a momentary event, being in this respect also like a tone or sound." ([15], p. 7) In the same way as expressions refer to something cognitions are always directed to an object. "Being directed to an object" (*viṣayatā*) is a special relation in Nyāya which obtains between a cognition and its content: *tatra viṣayā ghaṭapaṭādayo jñānecchādau viṣayatāsaṃbandhena vartante. viṣayitāsaṃbandhena ca jñānecchādayo ghaṭapaṭādau viṣaye tiṣṭhantīti.* ([4], p. 12, 16f = [10], p. 72) – "In that case objects, [such as] a pot, a cloth etc., occur via objecthood relation in a cognition, a wish etc. And by the relation 'being directed to an object' a cognition, a wish etc. depend on an object, [such as] a pot, a cloth etc." There seems to be a functional equivalence here between this objecthood relation and the reference relation in Western logic. The former links an object to a cognition, whereas the latter links an object to an expression which designates it.

Navya-Naiyāyikas do not only consider ordinary physical objects (such as a pot or a cloth), when they analyze the content of cognitions into properties and loci. The logically more interesting elementary constituents of a cognition include objects designated by means of nouns ending in abstract suffixes like *-tva* or *-tā*, which can be translated by means of English abstract suffixes like "-ness" or "-hood". In some cases circumlocutions by means of the word "being" may also be feasible as translations. So, the Sanskrit words *ghaṭatva* and *kṛtatva*, which derive from *ghaṭa* ("pot") and *kṛta* ("created"), can be translated by "potness" and "being created", respectively.

Some of these abstract nouns denote universals (*sāmānya*), one of the ontological categories which the Navya-Naiyāyikas inherited from the school of the Vaiśeṣikas. Numbers were also denoted by means of such abstract nouns, but classified as qualities. Thus, "twoness" (*dvitva*) was supposed to refer to the number "two" as a quality of two

things conceived of as a dyad.

Of course, not every abstract noun of this type can be said to be a name of some kind of real entity. Matilal refers to discussions in Navya-Nyāya concerning unlocatable "properties", such as "being the son of a barren woman", "being a golden mountain" etc.: "An unlocatable property is a suspect in Navya-nyāya. It is regarded as a ficticious property which cannot be located in our universe of loci." ([17], p. 147)

It might be tempting to regard what is commonly called an "imposed property" (*upādhi*) in Navya-Nyāya as fictitious as well. "The meanings of a large number of general terms in our language are construed as *upādhis*, i.e. 'nominal' properties, not objective universals. Kaṇāda and Praśatapāda did not really address this issue. It was, above all, Udayana who tried to provide a set of criteria for the exclusion of invalid or counterfeit universals, such as 'cookness' (*pācakatva*) or 'being an inhabitant of Ayodhyā' (*ayodhyāvāsitva*); ..." ([12], p. 119) Some nominal properties are like the latter example compound, whereas universals should be elementary. Udayana notes six other invalidating factors (*jātibādhaka*). (Cf. [12], p. 132)

Although such imposed properties do not belong to the elementary constituents of the empirical world, it would be a mistake to regard them as mere fabrications of the mind: "A failed universal as an *upādhi* would be a cognized property that like universals is 'repeatable,' i.e. can occur in more than one instance, but that falls to one or another of six 'blockers of natural-kind status', *jāti-bādhaka*. Such would be then a 'surplus property,' a property surplus to the system of ontological analysis. In other words, this would be a 'condition' in a very abstract and non-concomittal sense, a 'something extra' that is not just mind generated, that is a real property of something, but a property whose taxonomical character we have not determined. Some 'surplus properties' do seem to be mainly due to verbal excess, to saying things non-perspicaciously. But others do not seem so, and all *upādhi*-s are grounded in some fashion or other in the way the world is. Otherwise, they would not become objects, cognitive objects, that is, indicated by

our perceptions and conversations about the world." ([19], p. 25)

Finally, there is an ontological category of negative properties called "absence" or "non-being" (*abhāva*) in Navya-Nyāya, which is logically particularly interesting, as we will see below: Navya-Naiyāyikas distinguish two types of absence, namely "mutual absence" (*anyonyābhāva*) and "relational absence" (*saṃsargābhāva*). A mutual absence, which is also called "difference" (*bheda*), can be illustrated by the difference from a cloth (or: the mutual absence of a cloth) residing in a pot. Relational absence is the opposite of presence due to some kind of relation. Any locus where there is no pot, e.g., is characterized by a relational absence of pot.

2 Towards a formal reconstruction of the logic of Navya-Nyāya

2.1 G. Bealer's calculus T1 as a basic framework

The present formal reconstruction of the logic of Navya-Nyāya is an elaboration of B. K. Matilal's interpretation. Some decades ago Matilal and Bocheński already talked about intensional tendencies in the logic of Navya-Nyāya (cf. [15], p. 67 and p. 74, [16], p. 169, [5], p. 513 and p. 517). This impression is owing to the fact that logical inquiries in Navya-Nyāya are mostly concerned with the above-mentioned property names ending in abstract suffixes like *-tva* or *-tā*.

Matilal's idea was to formalize such property names by using Quine's notation for intensional contexts: If one writes Px for "x is a pot", then $x[Px]$ (which can be read as "being an x such that x is P") is an analytical expression for "potness". The function of the variable x in front of this term is to bind the free occurrence of x in Px. The square brackets around Px indicate an intensional context. If one substitutes an expression within the bracketed part by another one which is extensionally equivalent, one might change the reference of the property term. Such restrictions concerning the substitutability of extensionally equivalent expressions generally distinguish intensional

from extensional logical systems.

The property theory designed in [2] can be used to elaborate Matilal's formal analysis of Navya-Nyāya logic. Bealer never thought about such an application of his theory. He wants to explicate his realist notion of properties, relations and concepts, which is supposed to open up new vistas in analytical philosophy, especially in the realm of semantics, philosophical logic, philosophy of mind and philosophy of mathematics. For that purpose Bealer designed three calculi called "T1", "T2" and "T2'". T1 is especially suited to the treatment of modal matters, whereas T2 serves to check epistemic arguments. T2' is a synthesis of T1 and T2. T1 can also be used as a basis for a formal analysis of the logic of properties in Navya-Nyāya, as we will see below.

The language of T1 consists of the following primitive symbols (cf. [2], p. 43):

i) Logical operators: $\wedge$, $\neg$, $\exists$

ii) Predicate letters: F_1^1, F_2^1, ..., F_p^q

iii) Variables: x, y, z, ...

iv) Brackets: (,), [,]

Simultaneous inductive definition of terms and formulas:

1.) All variables are terms.

2.) If $t_1, \ldots, t_j$ are terms, then $F_i^j t_1 \ldots t_j$ is a formula.

3.) If A and B are formulas and v_k a variable, then $(A \wedge B)$, $\neg A$ and $\exists v_k A$ are formulas.

4.) If A is a formula and $v_1, \ldots, v_m$ $(0 \leq m)$ are distinct variables, then $[A]_{v_1 \ldots v_m}$ is a term.

The predicate letter F_1^2 is singled out as a distinguished logical predicate. Formulas of the form $F_1^2 t_1 t_2$ are to be written in the form

$t_1 = t_2$. Moreover, the symbols $\forall$, $\rightarrow$, $\vee$, $\leftrightarrow$, $\Box$ and $\Diamond$, which can be defined in terms of $\exists$, $\neg$, $\wedge$, [and], are included in the language of T1. (The definition of the modal operator $\Box$ by means of the square brackets will be explained below.)

Remarks:

- In 3.) A is an arbitrary formula, in which the variable v_k need not occur. Similarly, in 4.) the variables $v_1, \ldots, v_m$ are not required to be components of A. If they do occur in A, they are bound by the index variables. Generally speaking, an occurrence of a variable v_i is bound (free) if and only if it lies (does not lie) within a formula of the form $\exists v_i A$ or a term of the form $[A]_{v_1 \ldots v_i \ldots v_m}$.
- Occasionally, $a, b, c, \ldots$ will be used as constant symbols in the present formal representation of Navya-Nyāya expressions, although they are not part of the elementary symbols of T1.

Instead of going into the details of the semantics for the language of T1, it will suffice to explain the meaning of the "exotic" expressions:

A term of the form $[A]_{v_1 \ldots v_m}$ denotes ...

a) a proposition, if $m = 0$ ("that A").

b) a property, if $m = 1$ ("being a v_1 of which A is true").

c) an m-ary relation, if $m \geq 2$ ("the relation which holds between $v_1, \ldots, v_m$ *iff* A applies to them").

Remarks:

- In contrast to possible-worlds approaches Bealer treats intensional entities as individuals and not as functions. Therefore his property logic has much in common with the hypostasized understanding of properties in Navya-Nyāya. "The new semantic

method does not appeal to possible worlds, even as a heuristic. The heuristic used is simply that of properties, relations, and propositions, taken at face value." ([2], p. 42f)

- An expression of the form $\Box A$ is adopted as a convenient abbreviation of expressions such as $N[A]$, where N is a one-place predicate expressing "...is necessary". The semantic model structure for T1 (cf. [2], p. 49f) contains a condition which ensures that there is only one necessary truth (cf. [2], p. 52f). Since $[x = x]$ is a trivial necessary truth for any proposition x, $[A]$ can be identified with it if A is necessarily true. Therefore it is possible to define the modal operator $\Box$ simply by means of the square brackets: $\Box A :\leftrightarrow [A] = [[A] = [A]]$ (A is necessarily true *iff* the proposition "that A" is identical to a trivial necessary truth.)

- The term $[A]_{v_1 \ldots v_m}$ can be regarded as a counterpart of the class term $\{\langle v_1, \ldots, v_m \rangle | A\}$.

Bealer shows that T1 can be axiomatized in such a way that one gets a sound and complete calculus (cf. [2], p. 58f):

A1: Truth-functional tautologies

A2: $\forall v_i A(v_i) \rightarrow A(t)$, where t is free for v_i in A, i.e. no free occurrence of v_i in A lies within the scope of a quantifier or a sequence of index variables in a term $[\ldots]_{v_1 \ldots v_m}$ which would bind a variable occurring in t.

A3: $\forall v_i (A \rightarrow B) \rightarrow (A \rightarrow \forall v_i B)$, where v_i is not free in A.

A4: $v_i = v_i$

A5: $v_i = v_j \rightarrow (A(v_i, v_i) \leftrightarrow A(v_i, v_j))$, where $A(v_i, v_j)$ is a formula that arises from $A(v_i, v_i)$ by replacing some (but not necessarily all) free occurrences of v_i by v_j, and v_j is free for the occurrences of v_i that it replaces.

A6: $[A]_{u_1 \ldots u_p} \neq [B]_{v_1 \ldots v_q}$, where $p \neq q$.

A7: $[A(u_1, \ldots, u_p)]_{u_1 \ldots u_p} = [A(v_1, \ldots, v_p)]_{v_1 \ldots v_p}$, where these two terms are alphabetic variants.

A8: $[A]_{u_1 \ldots u_p} = [B]_{u_1 \ldots u_p} \leftrightarrow \Box \forall u_1 \ldots \forall u_p (A \leftrightarrow B)$

A9: $\Box A \rightarrow A$

A10: $\Box(A \rightarrow B) \rightarrow (\Box A \rightarrow \Box B)$

A11: $\Diamond A \rightarrow \Box \Diamond A$

R1: If $\vdash A$ and $\vdash (A \rightarrow B)$, then $\vdash B$.

R2: If $\vdash A$, then $\vdash \forall v_i A$.

R3: If $\vdash A$, then $\vdash \Box A$.

A1 – A5 along with R1 and R2 constitute an axiomatization of first-order predicate logic including identity. A6 – A8 determine how to deal with the intensional abstracts in T1. It is important to note that A8 furnishes a criterion for the identification of intensional abstracts. In this sense it has the same function as the axiom of extensionality in set theory. A9 – A11 and R3 are the modal part of the axiomatic system S5 of propositional modal logic. A11 is the S5-axiom **E** (cf. [13], p. 58), which is misquoted by Bealer: "$\Box A \supset \Box \Diamond A$" ([2], p. 59)

2.2 Extensions of T1 which function as alternatives to set theories.

2.2.1 The naive property abstraction in Navya-Nyāya

We need also some methods of formalization to express the following comprehension principle: *tattvavat tad eva*. – "Anything which possesses the property 'being that' is that." (Cf. [14], p. 36)

In order to see how this rule works one might replace the Sanskrit word *tat* ("that"), which has the same function as a schematic variable here, by words like *ghaṭa* ("pot"). *ghaṭatvavān ghaṭa eva* means:

"Anything which possesses the property 'potness' is a pot." Thus, the *tattvavat tad eva*-rule can be regarded as a kind of counterpart of the naive class abstraction in set theory:

$a \in \{x|A(x)\} \leftrightarrow A(x)$, where a is free for x in A and vice versa.

This equivalence can be transformed into a formal version of the naive property abstraction rule in Navya-Nyāya by replacing $\{x|A(x)\}$ by the corresponding property term in T1. In order to express that something possesses or is a locus of a class-like property we can use Bealer's Δ-relation, which functions as a counterpart of the $\in$-relation in set theory (cf. [2], p. 96). Thus, if we understand the *tattvavat tad eva*-rule in the sense of "a possesses (or is a locus of) the property 'being an x such that A is true of x' iff A is true of a", we can formalize it in the following way:

$(*)$ $a \,\Delta\, [A(x)]_x \leftrightarrow A(a)$, where a is free for x in A and vice versa.

Remarks:

- The present interpretation of the *tattvavat tad eva*-rule as an equivalence is confirmed by Matilal, who characterizes the specific style of Navya-Nyāya texts in the following way: "Simple predicate formulations, such as 'x is F' are noted, but only to be rephrased as 'x has *F-ness*' (where '*F-ness*' stands for the property derived from 'F')." ([16], p. 115)
- Apart from property abstraction relational abstraction also plays an important role in Navya-Nyāya logic (cf. [14], p. 44f and [16], p. 170f). Only dyadic relations with different relata are considered by Maheśa Chandra: *saṃbandhaḥ saṃnikarṣaḥ sa ca vibhinnayor vastunor viśeṣaṇaviśeṣyabhāvaprayojakaḥ.* ([4], p. 9, 13 = [10], p. 63) – "A relation is a connection and it establishes the state of being qualificand and qualifier of two different objects." It is important to note that Maheśa Chandra conceives of the pair

of relata as an ordered pair. The following remark about "indirect relations" (*paraṃparāsaṃbandha*) is certainly applicable to Maheśa Chandra's concept of relation in general: *... yasya paraṃparāsaṃbandhasyārambho yasmiṃś ca paryavasānaṃ tat tena paraṃparāsaṃbandhena tasmiṃs tiṣṭhati.* ([4], p. 10, 12 = [10], p. 66) – "..., from which an indirect relation starts (i.e. which is the initial point of the relation) and in which [the relation has its] final end (i.e. which is the terminal point of the relation) that depends on that on account of that indirect relation."

In order to formalize relational abstracts one can use the above-mentioned T1-terms of the form $[A]_{v_1 \ldots v_m}$, which are similar to Matilal's square bracket notation with prefixed binding variables (cf. [16], p. 170, where he suggests the following semi-formalization of *samaniyatatva*, i.e. "equi-locatability": $xy[x$ is *samaniyata* with $y]$).

The idea that a relational abstract $[xRy]_{xy}$ applies to an ordered pair $< x, y >$ can be expressed as $< x, y > \Delta\, [xRy]_{xy}$. According to Bealer we may understand $< x, y >$ as a property term if we replace the class term $\{\{x\}, \{x, y\}\}$, which corresponds to $< x, y >$ (according to the Wiener-Kuratowski definition), by applying the following context definition:

$\ldots \{x, y\} \ldots$ *iff*$_{\text{df}}$ $\exists z(\forall w(w\, \Delta\, z \leftrightarrow (w = x \vee w = y)) \wedge \ldots z \ldots)$, where z is a new variable not occurring in $\ldots$ (cf. [2], p. 83)

Since the Navya-Naiyāyikas had no Wiener-Kuratowski definition, they had to follow a different strategy to express that a pair of individuals is an instance of a relational abstract: When they want to express, e.g., that the ground is a locus of a certain pot, they use formulations like "The ground possesses locushood described by the pot". One of the purposes of this kind of circumbendibus is to specify the order of the members of the relation "being a locus of": The first member is the "possessor" of the relational abstract "locushood", which corresponds to the relation "being a locus of", and the second member is the "describer"

(*nirūpaka*) of the relational abstract.

- The use of the Δ-relation will be confined to cases where the second member of the relation is a class-like property term. Instances of location in a general sense ("x is a locus of y") will be rendered as xLy. Thus, one can formalize "The ground is a locus of the pot", e.g., as gLp. If a non-class-like property is located somewhere via the L-relation, then the locus should not be class-like either.

2.2.2 A property-theoretic variant of Zermelo-Russell's antinomy and its Sanskrit equivalent

Now, let us replace the word *tat* ("that") in the *tattvavat tad eva*-rule by *asvavṛttitva* ("being not resident in itself"). This property can easily be formalized. If we admit $x \Delta x$ as a formal equivalent of "x resides in itself", "being not resident in itself" can be expressed by $[\neg x \Delta x]_x$. Let r be an abbreviation of this property.

(Navya-)Naiyāyikas were aware that defining properties randomly can result in logical fallacies. Udayana, e.g., noticed that a contradiction in terms derives from the assumption of a universal for ultimate particularities and he included this defect in his list of criteria for identifying counterfeit universals (*jātibādhaka*). Nevertheless, there is no textual evidence that Navya-Nyāya logicians were aware of the possibility to use r to derive a variant of Zermelo-Russell's antinomy from the *tattvavat tad eva*-rule (cf. [7], p. 22, [8], p. 109 and [9], p. 144f):

(a) If r is resident in itself (i.e. if it is *svavṛtti*), then the property "being not resident in itself" (*asvavṛttitva*) resides in r. Therefore (according to the *tattvavat tad eva*-rule) r is not resident in itself (i.e. it is *asvavṛtti*). (Contradiction!) This is the formal counterpart of the argument:

$$r \Delta r \Rightarrow \underbrace{r \Delta [\neg x \Delta x]_x}_{\text{can be substituted for } a \Delta [A(x)]_x \text{ in } (*)} \Rightarrow \neg r \Delta r$$

(b) If r is not resident in itself (i.e. if it is *asvavṛtti*), then (according to the *tattvavat tad eva*-rule) the property "being not resident in itself" (*asvavṛttitva*) resides in r. Therefore r is resident in itself (i.e. it is *svavṛtti*). (Contradiction!) This is the formal counterpart of the argument:

$$\underbrace{\neg r \,\Delta\, r}_{\text{can be substituted for } A(a) \text{ in } (*)} \Rightarrow r \,\Delta\, [\neg x \,\Delta\, x]_x \Rightarrow r \,\Delta\, r$$

(a) and (b) together yield the following variant of Zermelo-Russell's antinomy:

$$r \,\Delta\, r \leftrightarrow \neg r \,\Delta\, r$$

2.2.3 An ST_2-style extension of T1 ("T1+") as an appropriate framework for a formal reconstruction of Navya-Nyāya logic

In order to modify $(*)$ in such a way that its paradoxical consequence disappears we can try to imitate the strategies which were pursued by the founders of set theories in order to safeguard the naive class abstraction rule against Zermelo-Russell's antinomy.

Certain restrictions in standard systems of set theory would, however, interfere with ontological commitments in Navya-Nyāya. In ZF (Zermelo-Fraenkel set theory), e.g., sets are the only objects in the domain of models of this system. However, since Navya-Naiyāyikas also talk about non-class-like objects, we need a system which is similar to set theories with urelements.

Moreover, some logical arguments in Navya-Nyāya involve universal properties such as nameability, which can be regarded as the analogue of a proper class in set theory. Talking about proper classes like, e.g., $\{x \,|\, x = x\}$ ("the universal class") is admissible in NBG (Neumann-Bernays-Gödel set theory), but not in ZF. Therefore a property adap-

tation of NBG with urelements is preferable as a system which may serve to model logical inquiries concerning properties in Navya-Nyāya.

Mendelson incorporates urelements into the framework of NBG (cf. [18], p. 297f). He uses lower-case Latin letters (x, y, z) as restricted variables for sets, capital Latin letters (X, Y, Z) as restricted variables for classes (i.e. for sets and proper classes) and lower-case boldface Latin letters $(\mathbf{x}, \mathbf{y}, \mathbf{z})$ as variables for classes and urelements alike (cf. [18], p. 297). In the present property adaptation of set-theory the same kinds of variables are used for set-like properties, class-like properties (i.e. set-like and properly class-like properties) and urelements, respectively.

A property version of the NBG comprehension axiom seems to be still too restrictive, because it does not include impredicative instantiations, which a Navya-Naiyāyika might not want to rule out (cf. the example given below). Since impredicative comprehension is admissible in QM (Quine-Morse set theory, also known as "Morse-Kelley set theory"), but not in NBG, the modification of $(*)$ should be patterned after the QM comprehension axiom. By means of the symbolization key ...

$P_s\mathbf{x}$: "$\mathbf{x}$ is a set-like property"
$U\mathbf{x}$: "$\mathbf{x}$ is an urelement"

... it can be expressed in the following way:

(C) $\forall\mathbf{x}(P_s\mathbf{x} \vee U\mathbf{x} \rightarrow (\mathbf{x}\Delta[A(\mathbf{y})]_{\mathbf{y}} \leftrightarrow A(\mathbf{x})))$, where $\mathbf{x}$ is free for $\mathbf{y}$ in A and vice versa.

Since (C) is impredicative, one can use it to formalize substitution instances of the *tattvavat tad eva*-rule, such as: "$\mathbf{x}$ is a locus of the property 'being a locus of some property which is equi-locatable with nameability' (*abhidheyatvasamaniyatakiṃciddharmādhikaraṇatva*) iff $\mathbf{x}$ is a locus of some property which is equi-locatable with nameability." The symbolization key ...

$N\mathbf{x}$: "$\mathbf{x}$ is nameable"
$\mathbf{x} \doteqdot \mathbf{y}$: "$\mathbf{x}$ is equi-locatable with $\mathbf{y}$", i.e. $\forall\mathbf{z}(\mathbf{z}L\mathbf{x} \leftrightarrow \mathbf{z}L\mathbf{y})$

...yields the following instantiation of (C):

$\forall\mathbf{x}(P_s\mathbf{x} \vee U\mathbf{x} \rightarrow (\mathbf{x}\Delta[\exists\mathbf{z}(\mathbf{x}\Delta\mathbf{z} \wedge \mathbf{z} \doteqdot [N\mathbf{y}]_\mathbf{y})]_\mathbf{x} \leftrightarrow \exists\mathbf{z}(\mathbf{x}\Delta\mathbf{z} \wedge \mathbf{z} \doteqdot [N\mathbf{y}]_\mathbf{y}))$

There is still another constraint in standard systems of set theory which should not be reproduced in a formal reconstruction of Navya-Nyāya logic: It is commonly assumed that proper classes can never be elements of classes, i.e. (even finite) collections of proper classes do not exist.

In Navya-Nyāya, however, it is possible to apply the *-tva*-abstraction technique repeatedly, so that one might create an expression like *abhidheyatvatva* ("nameabilityness"), which denotes a property of nameability. The analogue of such a property in set theory would be the singleton of the universal class, something which does not exist according to standard systems of set theory. One might call it a "hyper-class" ([6], p. 142).

An appropriate set-theoretic system on which we can model a formal reconstruction of Navya-Nyāya logic should endorse the existence of hyper-classes. In [6] (cf. p. 142f) the authors design such a system by combining the set theories of QM and ZF. The resulting system ST_2 can serve as a set-theoretic prototype of the Navya-Nyāya logic of property and location if we additionally take into account urelements. ST_2 with urelements consists of the following axioms:

(a) A sethood axiom: Every member of a set is a set or urelement.

(b) All the axioms of QM with urelements (with due regard to the above-mentioned notational convention for variables).

(c) The axioms of ZF with all variables replaced by upper case variables.

This is a two-tier set theory with sets and urelements in the bottom tier and classes in the upper tier. Due to (c) the existence of

hyper-classes is warranted in ST_2. Proper classes can be elements in this system, but they should still be distinguishable from sets. This is achieved by adding (a), which excludes that proper classes can be elements of sets.

For the purpose of a formal reconstruction of Navya-Nyāya logic we can make do with the property counterparts of only a few ST_2 axioms. All we need are the property-theoretic counterparts of ...

- (a), i.e. $\forall x \forall \mathbf{y}(\mathbf{y} \Delta x \rightarrow (P_s\mathbf{y} \vee U\mathbf{y}))$
- the QM-comprehension axiom, i.e. (C)
- a special axiom for urelements (cf. [18], p. 298), namely $\forall \mathbf{x}(U\mathbf{x} \rightarrow \forall \mathbf{y}(\mathbf{y} \not\Delta \mathbf{x}))$
- a ZF-style impredicative comprehension axiom with upper case variables, i.e.: $\exists Y(X \Delta Y \leftrightarrow A)$, where Y is distinct from X and is not free in A and A has the form $X \Delta Z \wedge B$. Alternatively, we can formulate it as $X \Delta [X \Delta Y \wedge A]_X \leftrightarrow X \Delta Y \wedge A$ (cf. [2], p. 99 and p. 265).

Instead of an extensionality principle there is A8 from T1 as a criterion for the identity of properties. All the other axioms in QM and ZF serve some kind of mathematical purpose which is irrelevant to the logic of Navya-Nyāya. There is, however, one exception, namely the axiom of regularity, as we will see below. The extension of T1 which includes the above-mentioned axioms of a property adaptation of ST_2 with urelements (excluding the axiom of regularity) will be called "T1+" from now on.

It might be tempting to choose NFU (Quine's New Foundation with urelements) instead of ST_2 as a model for a formal reconstruction of Navya-Nyāya logic, because NFU has only two axioms and one type of variables. The NFU axiom of extensionality is obsolete in the present context. So, we might just reformulate the NFU comprehension axiom:

$\exists y \forall x(x \in y \leftrightarrow Ax)$, where y is not free in A and A is stratified,

i.e. it is possible to index the variables in A such that $\in$ occurs only between variables with consecutive indices. ([6], p. 161f, [21], p. 210f)

Since $x = x$ is stratified, one can prove in NFU that $V \in V$. Similarly, one can prove the self-residence of a universal property by means of the following property adaptation of the NFU comprehension axiom:

$\forall x(x\Delta[A(y)]_y \leftrightarrow A(x))$, where $A(x)$ is stratified and x is free for y in A and vice versa.

Some Navya-Naiyāyikas regard the self-residence (*ātmāśraya*) of a property as a kind of absurdity. If we want to model their intuitions about properties, an NFU-style extension of T1 is not an appropriate framework. It is, however, congenial with respect to Navya-Naiyāyikas who affirm the self-residence of certain properties, as we will see below.

2.2.4 Well-foundedness vs. non-well-foundedness.

2.2.4.1 The position of Varadarāja and Maheśa Chandra: Some late Naiyāyikas and Navya-Naiyāyikas such as, e.g., Varadarāja and Maheśa Chandra would call for a well-foundedness condition on properties, similar to the axiom of regularity (or "foundation") in set theory. In standard systems of set theory this axiom excludes the existence of sets which are elements of themselves or – generally speaking – the existence of (potentially looping) infinite sequences $(a_n)_{n\in\mathbb{N}}$ such that $a_{i+1} \in a_i$ for all $i \in \mathbb{N}$:

$\forall A(\exists B(B \in A) \rightarrow \exists B(B \in A \wedge \neg\exists C(C \in A \wedge C \in B)))$ ("Every non-empty set A contains an element B which is disjoint from A.")

Theorem (in ZF):
$\neg\exists(a_n)_{n\in\mathbb{N}}\forall i \in \mathbb{N}(a_{i+1} \in a_i)$

PROOF:
If an infinite series $(a_n)_{n\in\mathbb{N}}$ such that $a_{i+1} \in a_i$ for all $i \in \mathbb{N}$ did exist, then there would be a set $A = \{a_1, a_2, a_3, \dots\}$ and $\forall a_i(a_i \in A \to a_{i+1} \in a_i \cap A)$, i.e. no element of A would be disjoint from A. But according to the axiom of regularity there is no such set A. ∎

A property analogue of the axiom of foundation is contained in the early Nyāya work Tārkikarakṣā (cf. [22]) by Varadarāja:

> *ātmāśrayas tathānyonyasaṃśrayaś cakrakāśrayaḥ/*
> *anavasthety amī tarkāḥ svarūpāsiddhihetavaḥ//* ([22], p. 234f)

> "Self-dependence, mutual dependence, circularity,
> the regressus in infinitum, these inferential blockers are the causes of the [probans's] being essentially unestablished."

In this verse Varadarāja expresses his misgivings about looping chains of dependence relations involving one member (*ātmāśraya* – "self-dependence"), two members (*anyonyasaṃśraya* – "mutual dependence") or more than two members (*cakrakāśraya* – "circularity", also: "arguing in a circular way"). Moreover, he refers to the regressus in infinitum (*anavasthā* – literally: "ungroundedness") as a further type of "inferential blocker" (*tarka*). In contrast to the so-called "hypothetical reasoning", which is also named *tarka*, an inferential blocker is regarded as an "unfavourable *tarka*" (*pratikūlatarka* – cf. [19], p. 94). An inference is blocked if the presence of the probans in the inferential subject entails an impossible circular chain of qualification or causation or a regressus in infinitum. In such a case the probans is said to be "essentially unestablished" (*svarūpāsiddha*), i.e. its presence in the inferential subject is in doubt (cf. [16], p. 51).

While Varadarāja is talking here about loops and infinitely descending chains in a more general sense, an opponent in the anonymous

Navya-Nyāya treatise Upādhidarpaṇa (cf. [23]) cites this verse (cf. [23], fol. 2a, 10) to support his view that in particular the existence of looping or infinitely descending chains of residence relations should be excluded.

Similarly, Maheśa Chandra argues that the residence relation is irreflexive and asymmetric. So, there can be no loops involving one or two members: *saṃbandho yadyapy ubhayaniṣṭho yathā kuṇḍabadarayoḥ saṃbandhaḥ kuṇḍe badare cāsti tathāpi kenacit saṃbandhena kaścid eva kutracid eva tiṣṭhati. yathā saṃyogena saṃbandhena kuṇḍa eva badaraṃ tiṣṭhati na tu badare kuṇḍam. evaṃ bhūtala eva ghaṭo vartate na tu ghaṭe bhūtalam iti. atra kāraṇam etat. saṃbandhasyaikaṃ pratiyogi. aparaṃ cānuyogi bhavati. yasya saṃbandhasya yat pratiyogi bhavati tena saṃbandhena tad eva tiṣṭhati. yac ca yasya saṃbandhasyānuyogi bhavati tena saṃbandhena tatra pratiyogi tiṣṭhati. yathā kuṇḍabadarayoḥ saṃyoge badaraṃ pratiyogi kuṇḍaṃ cānuyogīti kuṇḍe badaraṃ vartate. dharmadharmiṇoḥ saṃbandhasya dharmaḥ pratiyogī dharmi cānuyogi bhavati. ata eva dharma eva dharmiṇi vartate na tu dharmi dharme.* ([4], p. 12, 19f = [10], p. 72) – "Although a relation is situated in both [things], such as the relation between pot and dried ginger in a pot and in dried ginger, something nevertheless depends on something via a certain relation. Dried ginger, e.g., is in the pot on account of the relation contact, but the pot is not in dried ginger. In the same way the pot occurs on the ground, but not the ground on the pot. Here is this the reason: One is the adjunct of the relation and the other is the subjunct. That is the adjunct of that relation which depends [on something] via that relation. And that is the subjunct of that relation on which the adjunct depends via that relation. In the case of the contact of the pot and the dried ginger, e.g., the dried ginger is in the pot, because the dried ginger is the adjunct and the pot is the subjunct. In the case of property and property bearer the property is the adjunct of the relation and the property bearer is the subjunct. Therefore the property occurs on the property bearer, but not the property bearer on the property."

The existence of self-resident properties is explicitly denied in the

following passage: *ata eva pratiyogyanuyoginor abhede 'pi ghaṭe ghaṭo nāstīti saṃsargābhāvapratītiḥ.* ([4], p. 17, 16f = [10], p. 86) – "Therefore, when there is no difference between adjunct and subjunct, there is the cognition of the relational absence 'A pot is not in a (i.e. in the same) pot'." This statement should be understood in a more general sense, since the word "pot" (*ghaṭa*) is used in Navya-Nyāya as a kind of dummy singular term and in some contexts it can have the function of a variable (cf. [15], p. 23).

In order to exclude the existence of infinitely descending or looping chains of dependence relations for properties (in accordance with Varadarāja's and Maheśa Chandra's misgivings about such phenomena) one can postulate the following property-theoretic version of the QM-axiom of regularity (cf. [18], p. 302):

(R) $\forall X(\exists \mathbf{y}(\mathbf{y}\Delta X) \rightarrow \exists \mathbf{y}(\mathbf{y}\Delta X \wedge \forall \mathbf{z}(\mathbf{z}\Delta X \rightarrow \neg \mathbf{z}\Delta \mathbf{y})))$

Theorem:
$\neg\exists(a_n)_{n\in\mathbb{N}}\forall i \in \mathbb{N}(a_{i+1}\Delta a_i)$

Proof:
Assume indirectly: There is a sequence $(a_n)_{n\in\mathbb{N}}$ such that $\forall i \in \mathbb{N}(a_{i+1}\Delta a_i)$. Let X be a property whose loci are the members of $(a_n)_{n\in\mathbb{N}}$, i.e. $X = [\exists n \in \mathbb{N}(x = a_n)]_x$. (For the definiton of natural numbers as properties cf. [2], p. 121 and section 3.3 below.) Then $\forall a_i(a_i\Delta X \rightarrow a_{i+1}\Delta a_i \wedge a_{i+1}\Delta X)$. So, $\forall a_i(a_i\Delta X \rightarrow \exists \mathbf{z}(\mathbf{z}\Delta a_i \wedge \mathbf{z}\Delta X))$ – in contradiction to (R). ■

2.2.4.2 The position of the UD: The Upādhidarpaṇa (cf. [23]) is an anonymous early Navya-Nyāya treatise. It probably predates the great Navya-Nyāya philosopher Gaṅgeśa and might have been composed not long after 1325 AD (cf. [3], p. 67). The only extant manuscript is preserved in the Bhandarkar Oriental Research Institute and there is as yet no published edition of the text.

The author of the UD does not approve of the above-mentioned

restrictions concerning the relation of location (cf. [11]). He affirms the existence of non-well-founded properties "by assenting to the occurrence of something in itself" (*svasmin svavṛttitvābhyupagamena* – [23], fol. 4a, 4f).

The axiom of regularity, which captures the idea of well-foundedness in standard systems of set theory, is relatively independent, i.e. it is possible to construct models for the other axioms of these systems in which it fails. So, the assumption of such an axiom is optional from a logical point of view. Some set theorists just leave it out: "It therefore seems prudent (. . .), not to assume the axiom of foundation. In practice that is no great concession, however, since we shall focus exclusively on grounded collections (i.e. sets) in everything that we do from now on. Readers who believe there are ungrounded collections as well will thus find nothing here with which they can reasonably disagree: the most they are entitled to is a mounting sense of frustration that I am silent about them." ([20], p. 53)

By contrast, set theorists like Peter Aczel replace the axiom of regularity by an anti-foundation axiom, which "expresses, in a particular way, that every possible non-well-founded set exists." ([1], p. xviii) It is especially designed to provide solutions to certain equations which cannot be solved in standard systems of set theory. Thus, in Aczel's non-well-founded set theory the reflexive set Ω is the solution of the equation $x = \{x\}$.

It would surely be hazardous to incorporate a similar anti-foundation axiom referring to properties into a formal reconstruction of the logical framework of the UD. In Navya-Nyāya there is no solution to the equation $x = x\text{-}tva$. However, in order to model non-well-founded intuitions about properties, such as in the UD, one might want to adopt the idea of a stratified comprehension embodied in the NFU comprehension axiom. Stratified comprehension can be used to prove the self-residence of the property which according to the UD defines a so-called "associate condition" (*upādhi*). Although both are referred to as *upādhi*s in Sanskrit, the concept of "associate condition" has to be distinguished from the concept of "imposed property", which was introduced in section 1.

The function of an associate condition is to refute or undercut putative inferences. The stock example is the assumed inference of smoke (= the probandum) from fire (= the probans). In this case (which is just the reversal of the correct inference of fire from smoke) wet fuel serves as an associate condition, because it was supposed to be a necessary precondition for the production of smoke. With regard to a locus like molten metal, where the associate condition is missing, smoke cannot be inferred from fire. It is important to note here that (i) an associate condition has to be absent somewhere in order to function as an undercutter.

According to the UD an associate condition can also have the function of a corrector of an assumed inference in the sense that in an inferential subject exhibiting the associate condition together with the probans the probandum can be secured. Concerning the aforementioned example we can say that smoke is inferable from fire wherever the latter occurs in combination with wet fuel. So, the author of the UD also calls the associate condition an "enabling [condition]" (*prayojaka*) promoting an inferential knowledge. What is important here, is that (ii) the associate condition must be present somewhere in order to function as a corrector.

On account of (i) and (ii) the author of the UD defines an *upādhi* as something which is absent somewhere and also present somewhere. This defining characteristic ("being present somewhere and absent somewhere") applies to urelements and properties alike. However, if an urelement x (such as wet fuel) functions as an associate condition with respect to an assumed inference, we can always replace x by the equi-locatable property *x-vattva* (such as the property "being a locus of wet fuel"), which may equally well serve as an associate condition with respect to the same inference. So, it would be sufficient to take into account only class-like properties as associate conditions and then one can use the Δ-relation in order to state the defining characteristic of an associate condition as a property $u =_{\mathrm{df}} [\exists y(y\Delta x) \wedge \exists y(y \not\Delta x)]_x$. It resides in locatable properties which are not universal (excluding, e.g., the property "nameability"). Hence, $\exists y(y\Delta u) \wedge \exists y(y \not\Delta u)$. Since

this formula is stratified, the author of the UD can reasonably claim that $u\Delta u$.

3 Applications of T1+ to the analysis of Navya-Nyāya logic

The presentation of the logical framework for interpreting Navya-Nyāya logic in section 2 is basically the same as in [9], [10], [11], where, however, the systems ST_2 and NFU had not yet been taken into account. The utility of our methods of formalization will now be demonstrated by referring to some pertinent examples, namely the Navya-Naiyāyikas' operations applied to properties and relations (3.1), their discovery of theorems related to these operations (3.2) and their account of the reference of number words (3.3).

3.1 Operations on properties and relations

Navya-Nyāya logicians introduced several operations on properties and relations (cf. [10], 19f). Some of them are used by Bealer in order to explain how the denotation of a complex term $[A]_\alpha$ can be determined from the denotation(s) of the relevant syntactically simpler term(s) (cf. [2], p. 46f).

3.1.1 Negation of a property

The two types of "absence" in Navya-Nyāya have already been introduced in section 1. Following Bealer, who names property operators after their corresponding propositional operators, we can regard both types of absence as negations of properties: "..., what is the most obvious relation between $[Fx]_x$ and $[\neg Fx]_x$? As before, the second is the negation of the first." ([2], p. 47)

3.1.1.1 Mutual absence: A simple first-order representation of a statement of mutual absence, i.e. of a difference, such as the difference

from a cloth (or: the mutual absence of a cloth) residing in a pot, might look like this:

$\neg\exists x(Px \wedge Cx)$, where "$x$ is a pot" and "x is a cloth" are rendered by Px and Cx respectively.

Since in Navya-Nyāya a mutual absence is regarded as a denial of an identity, one might additionally capture this idea in the following equivalent formalization:

$\neg\exists x\exists y(Px \wedge Cy \wedge x = y)$

By means of Bealer's property terms one can also formalize a mere mutual absence (instead of a statement of such an absence). The following property term is a formal representation of the "difference from anything which is F":

$[\neg F\mathbf{x}]_{\mathbf{x}}$

One can regard this as a shorthand version of the following term, which captures also the idea that identity to the absentee is denied to any locus of such an absence:

(†) $[\neg\exists\mathbf{y}(F\mathbf{y} \wedge \mathbf{x} = \mathbf{y})]_{\mathbf{x}}$

3.1.1.2 Relational absence: The "relational absence of F (i.e. of anything which is F)" can be regarded as a property which characterizes something as being devoid of (or: no locus of) anything which is F and this can be rendered by means of the term:

(‡) $[\neg\exists\mathbf{y}(F\mathbf{y} \wedge \mathbf{x}L\mathbf{y})]_{\mathbf{x}}$

Since the present formalization of relational absences has basically the same syntactic structure as a mutual absence, namely $[\neg\phi(\mathbf{x})]_{\mathbf{x}}$,

where $\phi(\mathbf{x}) :\leftrightarrow \exists\mathbf{y}(F\mathbf{y} \wedge \mathbf{x}L\mathbf{y})$, one can also regard a relational absence as a negation, i.e. as the negation of the property "being a locus of an F" ($[\exists\mathbf{y}(F\mathbf{y} \wedge \mathbf{x}L\mathbf{y})]_{\mathbf{x}}$). Possessing a relational absence of F means to be different from a locus of an F. So, a relational absence turns out to be a special case of a mutual absence. Both can be regarded as negations.

Navya-Naiyāyikas see the essential difference between the two types of absence in the relation by which the absentee, the so-called "counterpositive" (*pratiyogin*), fails to reside in the locus of the absence. In the case of mutual absence this relation is identity. In the case of relational absence it is some kind of occurrence relation (such as contact, inherence etc.). This distinctive feature is duly mirrored in the present formalizations (†) and (‡), because they differ only with respect to the relations (L and $=$).

3.1.2 Sheffer stroke applied to properties

In Mathurānātha Tarkavāgīśa's Vyāptipañcakarahasyam (quoted in [14], p. 64f) this operation is named "conjoint absence" (*ubhayābhāva*). Maheśa Chandra characterizes it as "an absence due to prefixing 'being both'": *. . . paṭaghaṭobhayatvarūpeṇa vobhayatvapuraskāreṇābhāvo . . .* ([4], p. 14, 5f = [10], p. 76) – ". . . or an absence due to prefixing 'being both' in the form of 'being both, [i.e.] cloth and pot' . . . "

Following Bealer, who names property operators after their corresponding propositional operators, one can regard the Sheffer stroke applied to properties as the negation of the conjunction of properties. An example of such a property is the absence of both, cloth and pot, in a house where there is a cloth, but no pot. *evaṃ gṛhe kevalasya paṭasya sattve 'pi ghaṭasyābhāvena paṭaghaṭobhayasyāpy abhāvo 'sty eva. ekābhāvenobhayābhāvasyāvaśyaṃbhāvitvād . . .* ([4], p. 14, 8f = [10], p. 76) – "So, when there is only the cloth in the house, there is absence of both, the cloth and the pot <collectively>, because of the absence of the pot, because the absence of both <collectively> is necessary on account of the absence of one."

Since . . .

$$\neg\exists\mathbf{y}(P\mathbf{y}\wedge hL\mathbf{y})\wedge\exists\mathbf{z}(C\mathbf{z}\wedge hL\mathbf{z})\rightarrow\underbrace{\neg(\exists\mathbf{y}(P\mathbf{y}\wedge hL\mathbf{y})\wedge\exists\mathbf{z}(C\mathbf{z}\wedge hL\mathbf{z}))}_{\exists\mathbf{y}(P\mathbf{y}\wedge hL\mathbf{y})\uparrow\exists\mathbf{z}(C\mathbf{z}\wedge hL\mathbf{z})}$$

(where $P\mathbf{x}$ is to be read as "$\mathbf{x}$ is a pot", $C\mathbf{x}$ as "$\mathbf{x}$ is a cloth", $\mathbf{x}L\mathbf{y}$ as "$\mathbf{x}$ is a locus of $\mathbf{y}$" and h as "the house")

..., it is appropriate to say that the house possesses the negation of the conjunction of the properties $[\exists\mathbf{y}(P\mathbf{y}\wedge\mathbf{x}L\mathbf{y})]_{\mathbf{x}}$ and $[\exists\mathbf{z}(C\mathbf{z}\wedge\mathbf{x}L\mathbf{z})]_{\mathbf{x}}$, i.e.:

$$hL[\exists\mathbf{y}(P\mathbf{y}\wedge\mathbf{x}L\mathbf{y})\uparrow\exists\mathbf{z}(C\mathbf{z}\wedge\mathbf{x}L\mathbf{z})]_{\mathbf{x}}$$

3.2 Identities concerning iterated absences

[9] (cf. p. 147f) contains a proof of the following identity concerning iterated absences, which is endorsed by Mathurānātha (cf. [14], p. 71 and [16], p. 152f):

(Id) The relational absence (*saṃsargābhāva*) of the difference (*bheda*) from a pot is identical to potness.

According to 3.1.1.1 the difference from a pot can be represented as ...

$[\neg P\mathbf{x}]_{\mathbf{x}}$, where $P\mathbf{x}$ translates into "$\mathbf{x}$ is a pot".

In order to obtain a formal representation of the absence of the difference from a pot one might specify $F\mathbf{y}$ in (‡) (cf. 3.1.1.2) as ...

$\mathbf{y} = [\neg P\mathbf{x}]_{\mathbf{x}}$.

Then the absence of the difference from a pot (*ghaṭabhedābhāva*) can be expressed as ...

$[\neg\exists\mathbf{y}(\mathbf{y} = [\neg P\mathbf{x}]_{\mathbf{x}} \wedge \mathbf{x}\Delta\mathbf{y})]_{\mathbf{x}}$ ("being no locus of anything which is identical to the difference from a pot").

Now (Id) can be rendered as a T1+ proposition and one can prove it in T1+:

THEOREM (Id):
$[\neg\exists\mathbf{y}(\mathbf{y} = [\neg P\mathbf{x}]_{\mathbf{x}} \wedge \mathbf{x}\Delta\mathbf{y})]_{\mathbf{x}} = [P\mathbf{x}]_{\mathbf{x}}$

The proof contains an application of the following instantiation of (C):

$$\forall\mathbf{x}(P_s\mathbf{x} \vee U\mathbf{x} \rightarrow (\mathbf{x}\Delta[\neg P\mathbf{x}]_{\mathbf{x}} \leftrightarrow \neg P\mathbf{x}))$$

It is plausible to assume that both members of the equivalence $\mathbf{x}\Delta[\neg P\mathbf{x}]_{\mathbf{x}} \leftrightarrow \neg P\mathbf{x}$ in this formula are true of every $\mathbf{x}$ which fulfills the condition $\neg(P_s\mathbf{x} \vee U\mathbf{x})$, i.e. $\forall\mathbf{x}(\neg(P_s\mathbf{x} \vee U\mathbf{x}) \rightarrow \neg P\mathbf{x} \wedge \mathbf{x}\Delta[\neg P\mathbf{x}]_{\mathbf{x}})$. ("Individuals which are neither set-like properties nor urelements are different from pots and possess the property to be different from pots.") Hence, the equivalence $\mathbf{x}\Delta[\neg P\mathbf{x}]_{\mathbf{x}} \leftrightarrow \neg P\mathbf{x}$ can be applied unconditionally in this case.

PROOF of (Id):

(A1)	$\neg\neg P\mathbf{x} \leftrightarrow P\mathbf{x}$
(C)	$\neg\mathbf{x}\Delta[\neg P\mathbf{x}]_{\mathbf{x}} \leftrightarrow P\mathbf{x}$
(1st-order logic)	$\neg\exists\mathbf{y}(\mathbf{y} = [\neg P\mathbf{x}]_{\mathbf{x}} \wedge \mathbf{x}\Delta\mathbf{y}) \leftrightarrow P\mathbf{x}$
(R2, R3)	$\Box\forall\mathbf{x}(\neg\exists\mathbf{y}(\mathbf{y} = [\neg P\mathbf{x}]_{\mathbf{x}} \wedge \mathbf{x}\Delta\mathbf{y}) \leftrightarrow P\mathbf{x})$
(A8, R1)	$[\neg\exists\mathbf{y}(\mathbf{y} = [\neg P\mathbf{x}]_{\mathbf{x}} \wedge \mathbf{x}\Delta\mathbf{y})]_{\mathbf{x}} = [P\mathbf{x}]_{\mathbf{x}}$ ■

Maheśa Chandra states two other identities concerning iterated absences, namely the following reduction rules, which are referred to as (Id′) and (Id″) below: *tathāhi dvitīyābhāvaḥ (ghaṭābhāvābhāvaḥ) pratiyogi(ghaṭa)svarūpas tṛtīyābhāvaḥ (ghaṭābhāvābhāvābhāvaḥ) prathamābhāva(ghaṭābhāva)svarūpa iti prathamābhāvasya (ghaṭābhāvasya) ghaṭa*

iva dvitīyābhāvo 'pi (ghaṭābhāvābhāvo 'pi) pratiyogī. ([4], p. 15, 27f = [10], p. 81) – "So, the second absence (the absence of the absence of pot) is essentially identical to the counterpositive (pot). The third absence (the absence of the absence of the absence of pot) is essentially identical to the first absence (the absence of pot). So, the second absence (the absence of the absence of pot) is like 'pot' of the first absence (the absence of pot) a counterpositive (author's note: The "second absence" *ghaṭābhāvābhāva* is the counterpositive of the "third absence" *ghaṭābhāvābhāvābhāva*.)."

(Id′) The relational absence of the relational absence of a pot is identical to "pot".

(Id″) The relational absence of the relational absence of the relational absence of a pot is identical to the relational absence of a pot.

In order to explicate the right side of (Id′) in an appropriate way one might substitute "pot" (*ghaṭa*) by "being a locus of a pot" (*ghaṭavattva*), since this is common practice in Navya-Nyāya (cf. [16], p. 115). After all, the property "being a locus of a pot" is equi-locatable with every pot. Even though the Navya-Naiyāyikas do regard expressions like *ghaṭa* and *ghaṭavattva* as interchangeable, this is not unproblematic, because a pot possesses potness, whereas the property "being a locus of a pot" does not.

Theorem (Id′):
$[\neg\exists\mathbf{z}(\mathbf{z} = [\neg\exists\mathbf{y}(P\mathbf{y} \wedge \mathbf{x}L\mathbf{y})]_{\mathbf{x}} \wedge \mathbf{x}\Delta\mathbf{z})]_{\mathbf{x}} = [\exists\mathbf{y}(P\mathbf{y} \wedge \mathbf{x}L\mathbf{y})]_{\mathbf{x}}$

The proof contains an application of the following instantiation of (C):

$$\forall\mathbf{x}(P_s\mathbf{x} \vee U\mathbf{x} \rightarrow (\mathbf{x}\Delta[\neg\exists\mathbf{y}(P\mathbf{y} \wedge \mathbf{x}L\mathbf{y})]_{\mathbf{x}} \leftrightarrow \neg\exists\mathbf{y}(P\mathbf{y} \wedge \mathbf{x}L\mathbf{y})))$$

It is plausible to assume that both members of the equivalence

$\mathbf{x}\Delta[\neg\exists\mathbf{y}(P\mathbf{y}\wedge\mathbf{x}L\mathbf{y})]_{\mathbf{x}}\leftrightarrow\neg\exists\mathbf{y}(P\mathbf{y}\wedge\mathbf{x}L\mathbf{y})$ in this formula are true of every $\mathbf{x}$ which fulfills the condition $\neg(P_s\mathbf{x}\vee U\mathbf{x})$, i.e. $\forall\mathbf{x}(\neg(P_s\mathbf{x}\vee U\mathbf{x})\rightarrow\neg\exists\mathbf{y}(P\mathbf{y}\wedge\mathbf{x}L\mathbf{y})\wedge\mathbf{x}\Delta[\neg\exists\mathbf{y}(P\mathbf{y}\wedge\mathbf{x}L\mathbf{y})]_{\mathbf{x}})$. ("Individuals which are neither set-like properties nor urelements are different from loci of pots and possess the property to be different from loci of pots.") Hence, the equivalence $\mathbf{x}\Delta[\neg\exists\mathbf{y}(P\mathbf{y}\wedge\mathbf{x}L\mathbf{y})]_{\mathbf{x}}\leftrightarrow\neg\exists\mathbf{y}(P\mathbf{y}\wedge\mathbf{x}L\mathbf{y})$ can be applied unconditionally in this case.

Proof of (Id′):

(A1)	$\neg\neg\exists\mathbf{y}(P\mathbf{x}\wedge\mathbf{x}L\mathbf{y})\leftrightarrow\exists\mathbf{y}(P\mathbf{y}\wedge\mathbf{x}L\mathbf{y})$
(C)	$\neg\mathbf{x}\Delta[\neg\exists\mathbf{y}(P\mathbf{y}\wedge\mathbf{x}L\mathbf{y})]_{\mathbf{x}}\leftrightarrow\exists\mathbf{y}(P\mathbf{y}\wedge\mathbf{x}L\mathbf{y})$
(1st-order logic)	$\neg\exists\mathbf{z}(\mathbf{z}=[\neg\exists\mathbf{y}(P\mathbf{y}\wedge\mathbf{x}L\mathbf{y})]_{\mathbf{x}}\wedge\mathbf{x}\Delta\mathbf{z})\leftrightarrow\exists\mathbf{y}(P\mathbf{y}\wedge\mathbf{x}L\mathbf{y})$
(R2, R3)	$\Box\forall\mathbf{x}(\neg\exists\mathbf{z}(\mathbf{z}=[\neg\exists\mathbf{y}(P\mathbf{y}\wedge\mathbf{x}L\mathbf{y})]_{\mathbf{x}}\wedge\mathbf{x}\Delta\mathbf{z})\leftrightarrow\exists\mathbf{y}(P\mathbf{y}\wedge\mathbf{x}L\mathbf{y}))$
(A8, R1)	$[\neg\exists\mathbf{z}(\mathbf{z}=[\neg\exists\mathbf{y}(P\mathbf{y}\wedge\mathbf{x}L\mathbf{y})]_{\mathbf{x}}\wedge\mathbf{x}\Delta\mathbf{z})]_{\mathbf{x}}=[\exists\mathbf{y}(P\mathbf{y}\wedge\mathbf{x}L\mathbf{y})]_{\mathbf{x}}$

■

In order to prove (Id″) and any other reduction rule which states the identity of an uneven number of such relational absences to a single relational absence, it suffices to prove:

(Id*) The relational absence of the property "being a locus of a pot" is identical to the relational absence of a pot.

By adding one relational absence on both sides of (Id′) we can infer from (Id′) that the relational absence of the relational absence of the relational absence of a pot is identical to the relational absence of <u>pot</u> (where the underlined "pot" is supposed to be explicated in the sense of "the property 'being a locus of a pot'"). On account of (Id*) the relational absence of "pot", i.e. of the property "being a locus of a

pot", is identical to the relational absence of a pot, and this proves (Id″).

THEOREM (Id*):
$[\neg\exists\mathbf{z}(\mathbf{z} = [\exists\mathbf{y}(P\mathbf{y} \wedge \mathbf{x}L\mathbf{y})]_{\mathbf{x}} \wedge \mathbf{x}\Delta\mathbf{z})]_{\mathbf{x}} = [\neg\exists\mathbf{y}(P\mathbf{y} \wedge \mathbf{x}L\mathbf{y})]_{\mathbf{x}}$

The proof contains an application of the following instantiation of (C):

$$\forall\mathbf{x}(P_s\mathbf{x} \vee U\mathbf{x} \rightarrow (\mathbf{x}\Delta[\exists\mathbf{y}(P\mathbf{y} \wedge \mathbf{x}L\mathbf{y})]_{\mathbf{x}} \leftrightarrow \exists\mathbf{y}(P\mathbf{y} \wedge \mathbf{x}L\mathbf{y}))).$$

It is plausible to assume that neither of the members of the equivalence $\mathbf{x}\Delta[\exists\mathbf{y}(P\mathbf{y} \wedge \mathbf{x}L\mathbf{y})]_{\mathbf{x}} \leftrightarrow \exists\mathbf{y}(P\mathbf{y} \wedge \mathbf{x}L\mathbf{y})$ in this formula is true of any $\mathbf{x}$ which fulfills the condition $\neg(P_s\mathbf{x} \vee U\mathbf{x})$, i.e. $\forall\mathbf{x}(\neg(P_s\mathbf{x} \vee U\mathbf{x}) \rightarrow \neg\exists\mathbf{y}(P\mathbf{y} \wedge \mathbf{x}L\mathbf{y}) \wedge \neg\mathbf{x}\Delta[\exists\mathbf{y}(P\mathbf{y} \wedge \mathbf{x}L\mathbf{y})]_{\mathbf{x}})$. ("Individuals which are neither set-like properties nor urelements are different from loci of pots and do not possess the property to be loci of pots.") Hence, the equivalence $\mathbf{x}\Delta[\exists\mathbf{y}(P\mathbf{y} \wedge \mathbf{x}L\mathbf{y})]_{\mathbf{x}} \leftrightarrow \exists\mathbf{y}(P\mathbf{y} \wedge \mathbf{x}L\mathbf{y})$ can be applied unconditionally in this case.

PROOF of (Id*):

(A1)	$\neg\exists\mathbf{y}(P\mathbf{x} \wedge \mathbf{x}L\mathbf{y}) \leftrightarrow \neg\exists\mathbf{y}(P\mathbf{y} \wedge \mathbf{x}L\mathbf{y})$
(C)	$\neg\mathbf{x}\Delta[\exists\mathbf{y}(P\mathbf{y} \wedge \mathbf{x}L\mathbf{y})]_{\mathbf{x}} \leftrightarrow \neg\exists\mathbf{y}(P\mathbf{y} \wedge \mathbf{x}L\mathbf{y})$
(1st-order logic)	$\neg\exists\mathbf{z}(\mathbf{z} = [\exists\mathbf{y}(P\mathbf{y} \wedge \mathbf{x}L\mathbf{y})]_{\mathbf{x}} \wedge \mathbf{x}\Delta\mathbf{z}) \leftrightarrow \neg\exists\mathbf{y}(P\mathbf{y} \wedge \mathbf{x}L\mathbf{y})$
(R2, R3)	$\Box\forall\mathbf{x}(\neg\exists\mathbf{z}(\mathbf{z} = [\exists\mathbf{y}(P\mathbf{y} \wedge \mathbf{x}L\mathbf{y})]_{\mathbf{x}} \wedge \mathbf{x}\Delta\mathbf{z}) \leftrightarrow \neg\exists\mathbf{y}(P\mathbf{y} \wedge \mathbf{x}L\mathbf{y}))$
(A8, R1)	$[\neg\exists\mathbf{z}(\mathbf{z} = [\exists\mathbf{y}(P\mathbf{y} \wedge \mathbf{x}L\mathbf{y})]_{\mathbf{x}} \wedge \mathbf{x}\Delta\mathbf{z})]_{\mathbf{x}} = [\neg\exists\mathbf{y}(P\mathbf{y} \wedge \mathbf{x}L\mathbf{y})]_{\mathbf{x}}$

■

3.3 A quasi-Fregean account of the reference of number words

According to Maheśa Chandra words for natural numbers refer to properties. So, "two" refers to twoness. There are two kinds of twoness: *ayaṃ na dvau kiṃtu dvitvavān iti pratīter dvau dvitvavān iti padayor arthaviśeṣāvadhāraṇāya navīnaiḥ kaścit paryāptināmakaḥ saṃbandhaḥ svīkriyate. paryāptiḥ paryavasānaṃ sākalyena saṃbandho 'rthād yasya yāvanta āśrayāḥ santi tasya tāvatsv evāśrayeṣu militeṣv eva saṃbandhaḥ. paryāptisaṃbandhena dvitvasaṃkhyā militayor eva dvayor vartate na tv ekaikasmin. evaṃ tritvasaṃkhyā militeṣv eva triṣu vartate na tv ekaikasmin dvayor vā. ata eva dvitvādayaḥ saṃkhyā vyāsajyavṛttaya (vyāsajya sarvam evādhāram adhikṛtya vartante) ity ucyante. evaṃ ca dviśabdasya paryāptisaṃbandhena dvitvādhāratāpratīter ekasya ca paryāptisaṃbandhena dvitvādhāratvābhāvād ayaṃ na dvāv iti pratītir bhavati. samavāyasaṃbandhena punar dvitvasaṃkhyā dvayor ekaikasminn api tiṣṭhatīti samavāyasaṃbandhena dvitvāśraya ity artham abhipretya dvitvavān iti prayogaḥ. tataś cāyaṃ na dvau kiṃtu dvitvavān iti vākyasyāyaṃ na paryāptisaṃbandhena dvitvavān kiṃtu samavāyasaṃbandhena dvitvavān ity arthaḥ.* ([4], p. 12, 3f = [10], p. 71) – "A certain relation called *paryāpti* is assumed by Navya-Naiyāyikas for the purpose of specifying the different meanings of two expressions, [namely] 'two' [and] 'It possesses twoness' [as part] of the cognition 'This is not two, but it possesses twoness'. *paryāpti* is completion, a relation on account of thoroughness. The relation occurs in as many substrates of n collectively as there are on account of n's meaning. By the *paryāpti* relation the number 'twoness' resides in 2 [things] collectively, not in each one. In the same way the number 'threeness' resides in 3 [things] collectively, not in each one or in two. Therefore the numbers "twoness" etc. are said to have a joint occurrence. (They reside jointly with respect to all as a substrate.) And so, since for the word 'two' there is the cognition of substratumness of twoness via *paryāpti* relation and because of one [thing's] absence of substratumness of twoness via *paryāpti* relation, there is the cognition 'This is not two'. <translator's note: Substratumness of twoness is absent from

a single thing via *paryāpti* relation.> But since the number 'twoness' depends on each of two [things] via inherence relation, there is the usage 'It has twoness' with the intended meaning 'It is the substrate of twoness via inherence relation'. And therefore the sentence 'This is not two, but it possesses twoness' has the meaning 'This does not possess twoness via *paryāpti* relation, but it possesses twoness via inherence relation.'"

The twoness which inheres in each of two things when someone regards them as a dyad corresponds to the old Vaiśeṣika understanding of numbers. According to Ingalls "The 'two-ness that inheres in each member of pairs' corresponds to the Western 'class of two members'." ([14], p. 77) Although this comparison appears to be a little bit flawed, there is some truth in Ingalls's claim that the twoness which resides by *paryāpti* in each dyad can be compared to Frege's understanding of numbers: "This theory that numbers subsist by *paryāpti* in effect points out what Frege first pointed out in Europe in the nineteenth century. ... The 'two-ness that is related by *paryāpti* to the pairs and not to the members of the pairs' corresponds to the Western 'number two, the class of all classes of two members'." ([14], p. 77)

Since Maheśa Chandra regards the referent of "two" not as a class but as a property, it would be more appropriate to compare the twoness residing by *paryāpti* in each dyad to Bealer's neo-Fregean analysis of natural numbers. Bealer interprets "12", e.g., as "a property whose instances are all and only properties having 12 instances" ([2], p. 124). If we regard "being an apostle" as instantiated by twelve individuals, then this property is one of the instances of "12".

Bealer specifies his concept of natural number (including the number 0, which Maheśa Chandra does not refer to) by means of the following three definitions (cf. [2], p. 121):

(1) $0 =_{\mathrm{df}} [\neg\exists\mathbf{u}(\mathbf{u}\Delta\mathbf{y})]_{\mathbf{y}}$ ($0 =_{\mathrm{df}}$ the property of being a property with no instances.)

(2) $\mathbf{x}' =_{\mathrm{df}} [\exists\mathbf{u}(\mathbf{u}\Delta\mathbf{x} \wedge \exists\mathbf{v}(\neg(\mathbf{v}\Delta\mathbf{u}) \wedge \mathbf{y} \div [\mathbf{w}\Delta\mathbf{u} \vee \mathbf{w} = \mathbf{v}]^{\mathbf{uv}}_{\mathbf{w}}))]^{\mathbf{x}}_{\mathbf{y}}$ (the successor of $\mathbf{x} =_{\mathrm{df}}$ the property of being a property with one more

instance than the instances of **x**.)

(3) $NN\mathbf{x}\ \mathit{iff}_{\text{df}}\ \forall\mathbf{z}((0\Delta\mathbf{z}\wedge\forall\mathbf{y}(\mathbf{y}\Delta\mathbf{z}\rightarrow\mathbf{y}'\Delta\mathbf{z}))\rightarrow\mathbf{x}\Delta\mathbf{z})$ (**x** is a natural number iff_{df} **x** is an instance of every property **z** such that 0 is an instance of **z** and the successor $\mathbf{y}'$ of every instance **y** of **z** is also an instance of **z**.)

Remarks:

- In (2) upper index variables on the right side of a property term signify the free variables in the wff within square brackets which are not bound by any lower index variable of the property term. Moreover, $\mathbf{y}\doteqdot\mathbf{z}\ \mathit{iff}_{\text{df}}\ \forall\mathbf{w}(\mathbf{w}\Delta\mathbf{y}\leftrightarrow\mathbf{w}\Delta\mathbf{z})$.

- It should be noted that Bealer defines a number n as a property whose instances are properties having n instances. So, (2) can be paraphrased in the following way: $\mathbf{x}'$ is a property such that each of its instances is a property **y** which has as instances (i) the same instances as a property **u** which is an instance of **x** and (ii) an instance **v** which is not an instance of **u**.

- The twoness which resides by *paryāpti* "resides in 2 [things] collectively": *... militayor eva dvayor vartate ...* ([4], p. 12, 6f = [10], p. 71) One may wonder what is to be understood by "collectively" (*milita*) here. If we want to render Maheśa Chandra's idea precise, we might do it in the same way as Bealer, who introduces properties as instances of the number n such that each of them has n instances. Alternatively, one might conceive of the entites which are instantiated by n individuals as classes of n elements. However, there is no reference to classes in the ontological framework of Navya-Nyāya.

- Although Bealer's use of the term "property" largely coincides with what Maheśa Chandra understands by a *dharma*, definition (1) seems to be an exception. Unlike the properties as instances of the property corresponding to 0 according to (1) a *dharma* should always have instances. But this is a minor problem, which does

not affect the present formal reconstruction of Maheśa Chandra's ideas, since he does not include the number 0 in his analysis of natural numbers.

References

[1] P. Aczel, *Non-Well-Founded Sets*. [CSLI Lecture Notes 14]. Stanford, CA: Stanford University, Center for the Study of Language and Information. 1988.

[2] G. Bealer, *Quality and Concept*. Oxford: Clarendon Press. 1982.

[3] D. Bhattacharya, *History of Navya-Nyāya in Mithilā*. [Mithila Institute Series 2]. Darbhanga: Mithila Institute of Post-Graduate Studies and Research in Sanskrit Learning. 1958.

[4] *Brief Notes on the Modern Nyāya System of Philosophy and its Technical Terms*. By Mahāmahopādhyay Maheśa Chandra Nyāyaratna. Calcutta: Hare Press. 1891.

[5] J. M. Bocheński, *Formale Logik*. [Orbis Academicus III, 2]. Freiburg/Munich: Karl Alber. 1978.

[6] A. Fraenkel, Y. Bar-Hillel, A. Levy, *Foundations of Set Theory*. [Studies in Logic and Foundations of Mathematics 67]. Amsterdam/London/New York/Oxford/Paris/Shannon/Tokyo: North Holland. 1973.

[7] E. Guhe, *Die Lehre von der zusätzlichen Bestimmung (upādhi) im Upādhidarpaṇa*. Unpublished Dissertation. Vienna. 1999.

[8] E. Guhe, Intensionale Aspekte der indischen Logik. Berliner Indologische Studien, vol. 13/14 (2000). pp. 105-116

[9] E. Guhe, George Bealer's Property Theories and their Relevance to the Study of Navya-Nyāya Logic, in: *Logic, Navya-Nyāya & Applications*. Ed. M. K. Chakraborti et al. [Studies in Logic, vol. 15]. pp. 139-153. London: College Publications. 2008.

[10] E. Guhe, *Maheśa Chandra Nyāyaratna's "Brief Notes on the Modern Nyāya System of Philosophy and its Technical Terms"*. Shanghai: Fudan University Press. 2014.

[11] E. Guhe, The Problem of Foundation in Early Nyāya and in Navya-Nyāya. History and Philosophy of Logic, vol. 36/2 (2015). pp. 97-113.

[12] W. Halbfass, *On Being and What There Is*. Classical Vaiśeṣika and the History of Indian Ontology. Albany: State University of New York Press. 1992.

[13] G. E. Hughes and M. J. Cresswell, *A New Introduction to Modal Logic*. London/New York: Routledge. 1996.

[14] D. H. H. Ingalls, *Materials for the Study of Navya-Nyāya Logic*. [Harvard Oriental Series 40]. Cambridge, Massachusetts: Harvard University Press. 1951.

[15] B. K. Matilal, *The Navya-Nyāya Doctrine of Negation*. [Harvard Oriental Series 46]. Cambridge, Massachusetts: Harvard University Press. 1968.

[16] B. K. Matilal, *Logic, Language and Reality*. Delhi: Motilal Banarsidass. 1990.

[17] B. K. Matilal, *The Character of Logic in India*. Albany: State University of New York Press. 1998.

[18] E. Mendelson, *Introduction to Mathematical Logic*. London: Chapman & Hall. 1997.

[19] S. Phillips, *Gaṅgeśa on the Upādhi, the "Inferential Undercutting Condition"*. Introduction, Translation and Explanation (with N. S. Ramanuja Tatacharya). Delhi: Indian Council of Philosophical Research. 2002.

[20] M. Potter, *Set Theory and its Philosophy*. Oxford: Oxford University Press. 2004.

[21] W. v. O. Quine, *Mengenlehre und ihreLogik*. Transl. by Anneliese Oberschelp of *Set Theory and its Logic*. Braunschweig: Friedr. Vieweg + Sohn. 1973.

[22] *tārkikarakṣā. śrīmadācāryavaradarājaviracitā. tatkṛtasārasaṅgrahābhidhavyākhyāsahitā. mahopādhyāyakolācaśrīmallināthasūriviracitayā niṣkaṇṭakākhyayā vyākhyayā jñānapūrṇanirmitayā laghudīpikākhyayā ṭīkayā samanvitā.* Varanasi 1903: Pandit Reprint.

[23] *Upādhidarpaṇa.* BORI-Ms. No. 6. 1898-99.

Eberhard Guhe[1]

[1]Fudan University, Shanghai
Email: eberhard@guhe.de

Insertion Modeling and Its Applications

Alexander Letichevsky, Oleksandr Letychevskyi,
Vladimir Peschanenko

Abstract

The paper relates to the theoretical and practical aspects of insertion modeling. Insertion modeling is a theory of agents and environments interaction where an environment is considered as agent with a special insertion function. The main notions of insertion modeling are presented. Insertion Modeling System is described as a tool for development of deferent kinds of insertion machines. The research and industrial applications of Insertion Modeling System are presented.

Keywords: process algebra, insertion modeling, formal models, verification.

1 Introduction

Insertion modeling is an approach for research of distributed multi-agent systems and for development of tools for verification of its models. The first papers about insertion modeling were published about 20 years ago[1-2]. A model of the agents and environments interaction which helps the insertion function notion was presented in these papers.

The main sources of insertion modeling are in a model of interacting control and operating automata, which were found by V.M. Glushkov[3,4] for the computers description. An algebraic abstraction of this model has been studied in the theory of discrete transformers and has provided some important results on the problem of equivalence of programs, their equivalent transformation and optimization. Macroconveyor models of parallel computing [5] are even closer to the model of interaction between agents and environments. In these models processes corresponding to

parallel processors can be regarded as agents interacting in distributed environment data structures. In recent years the insertion simulation becomes a tool for development applications of verification of systems requirements and specifications of distributed interacting systems [6-10].

Another source of insertion modeling is a general theory of interacting information processes, which was created in previous century and is the basis for modern research in this area. It includes CCS (Calculus of Communicated Processes) [11-12] and π-calculus of R. Milner [13], CSP (Communicated Sequential Processes) of T. Hoar [14], ACP (Algebra of Communicated Processes) [15] and many other different branches of these basic theories. A quite complete review of the classical theory of processes is represented in the handbook on algebra processes [16], which was published in 2001.

The second section is defined by the algebra of behaviors and the bisimulation equivalence of transition systems. The third section introduces the concepts of environment and agents features. The fourth section is devoted to the Insertion modeling system. The fifth section deals with the application of insertion modeling, and finally discusses the possibilities for further development and possible new applications.

2 Behavior algebras

2.1 Transition System

A common approach for describing the dynamics of systems in modern computer science is the notion of transition system, which is defined by sets of states and transitions. Usually this notion is enriched by the additional structures, the most important of which are the transition labelling (labelled transition system introduced by Park [18] to describe the behavior of automata on infinite words). The basic notion in the insertion modeling is an attribute transition system[10], which is defined as follows:

$$< S, A, U, T, \varphi > \qquad (1)$$

where S is a set of states, A is a set of actions, which are used for marking the transition, U is a set of labeled attributes, which are used for marking the states, T is transition relation: $T \subseteq S \times A \times S \cup S \times S$, which consists of labeled transitions $s \xrightarrow{a} s'$ and not labeled transitions $s \to s'$.

Function $\varphi: S \to U$ is a function of labeling states. U could be defined as a set $U = D^R$ of mapping of a set R of attributes in a set of data D (a range of values of attributes) or as a $U = (D_\xi^{R_\xi})_{\xi \in \Xi}$, where Ξ is a set of data types. A formula of some logic language *L(R)* is used for symbolic modeling as attributes labels $U \subseteq L(R)$, where R is a set of attributes or a set of attributes with types $R = (R_\xi)_\xi$. It could be interpreted by first order language, which could be expanded by some temporal logic modality. States labeling is considered as some equivalence for symbolic case.

Transition system can also be configured by highlighting some specific sets of states from the set of states *S*. Among them there are the most important set of initial states S_0, a set of termination states S_Δ and a set of non-defined states $S_\perp$. The last one is used in the theory to determine the relationship of approximation and to build infinite systems in form of finite limits.

As in the theory of automata states the transition systems are considered as some equivalence. In the branch of different equivalences which are considered in the [19] the most important are the trace and bisimulation equivalence (strongest and weakest respectively).

For simplicity, we consider only the system with no hidden transitions. History of operation of attribute transition system is defined as a finite or infinite sequence $s_1 \xrightarrow{a_1} s_2 \xrightarrow{a_2} \ldots$ of transitions, and a trace corresponding to this history is defined as a sequence

$$\varphi(s_1) \xrightarrow{a_1} \varphi(s_2) \xrightarrow{a_2} \ldots$$

The trace is called maximal if it can't be continued. Let *L(s)* be the set of all maximal traces which are started in a state *s*. The states *s* and *s'* are called trace equivalent, if *L(s)=L(s').*

Bisimulation equivalence is weaker than trace and defined by thinner manner. A binary relation *R* on the set of states of the system (1) is called a relation of bisimulation, if for every pair (*s*, *s'*) of its states the following rules are true:

$$1)\,(s,s') \in R \Rightarrow \varphi(s) = \varphi(s')$$

$$2)\,(s,s') \in R \wedge s \xrightarrow{a} t \Rightarrow \exists t'((t,t') \in R \wedge s' \xrightarrow{a} t')$$

$2)(s,s') \in R \wedge s' \xrightarrow{a} t' \Rightarrow \exists t((t,t') \in R \wedge s \xrightarrow{a} t)$

The state s and s' system (1) are called bisimulation equivalent if a bisimulation relation R exists, such as $(s,s') \in R$.

Equivalence of systems is usually defined in terms of their equivalence of states. For example, for the initial systems two systems are declared to be equivalent, if the initial state of each of them is equivalent to the initial state of another. The difference between the trace and bisimulation equivalence occurs only in the case of non-deterministic systems. A labeled system is called deterministic if

$$s \xrightarrow{a} s' \wedge s \xrightarrow{a} s'' \Rightarrow (s',s'') \in R$$

Two deterministic systems are bisimulation equivalent if and only if they are trace equivalent.

2.2 Behavior algebra

In contrast to the trace equivalence for which the invariant of equivalence (a set of traces) is given together with the definition, the invariant of bisimulation equivalence is not so obvious. In insertion modeling as invariants (generally infinite) expressions or system of equations in algebra behavior are used. A behavior algebra is arranged simply. It is a two-sorted algebra <*U, A*>, the first component *U* is a set of behaviors, and the second *A* is a set of actions. The signature of the behavior algebra consists of two operations, one relation and three constants. The first operation *a.u* is called *prefixing*. Its arguments are action *a* and behavior *u*. The result is a new behavior. The second operation is the operation of a non-deterministic choice of *u+v*. This is a binary operation defined in the set of behaviors. It is commutative, associative and idempotent. The behavior algebras constants are the successful termination Δ, undefined behavior $\perp$ and the deadlock behavior 0, which is a neutral element of non-deterministic choice. On the set of behaviors a binary relation of approximation $\subseteq$ is defined, which is a relation of a partial order with the smallest element $\perp$. Prefixing and non-deterministic choice operation are monotonous and continuous with respect to this relation. The main role is played by a full behavior algebra *F(A)*, which contains all limits of directed sets and, therefore, a theorem on the minimal fixed point is applied. The exact structure algebra *F(A)* (for any, including infinite number of actions) is presented in [17].

In the full algebra of behavior each element has the following representation:

$$u = \sum_{i \in I} a_i.u_i + \varepsilon_u \ ,$$

which is uniquely defined (up to a commutativity and associativity), if all $a_i.u_i$ are different.

With each state *s* of transition system a behavior *beh(s)=u_s* of system *S* is associated as the lowest component of the system of equations

$$u_s = \sum_{s \xrightarrow{a} t} a.u_t + \varepsilon_s$$

where $\varepsilon_s = 0, \Delta, \perp, \Delta + \perp$ depends on the conditions $s \notin S_\Delta \cup S_\perp$, $s \in S_\Delta \setminus S_\perp, s \in S_\perp \setminus S_\Delta, s \in S_\Delta \cup S_\perp$ respectively. The main theorem, which characterizes a bisimulation equivalence claims that *two states are bisimulation equivalent if and only if they have equal behavior*. Other approaches to the characterization of a bisimulation equivalence can be found in [20].

3 Agents and Environments

Agent is a transition system, which defines a state up to bisimulation equivalence.

Environment is an agent that has an insertion function. In additional environments there is <*E*,*C*,*A*,Ins>, where *E* is a set of states of an environment, *C* is a set of actions which could be inserted into an environment, $Ins: E \times F(A) \to E$ is an insertion function. Since the states transition systems are considered as bisimulation equivalence, they can be identified with the behavior and talk about continuity of an insertion function. The main requirement for the environment is a continuity of an insertion function. This assumption implies a number of useful effects. For example, the fact that an insertion function can be set with the help of systems of rewriting rules as the minimal fixed point of the system of functional equations. A result *Ins(e,u)* of agents insertion, which is in a state *u*, is defined as $e[u]$. Assuming $e[u,v] = (e[u])[v]$ we get the opportunity to talk about the combination of agents that are inserted in an environment and to consider the state of an environment of the form

$e[u_1, u_2, ...]$. Taking into account that an environment is an agent, it can be inserted in a top level environment, considering the multi-level environments like $e[e_1[u_{11}, u_{12}, ...]_{E_1}, e_2[u_{21}, u_{22}, ...]_{E_2}, ...]$, where definition $e[u_1, u_2, ...]_E$ clearly shows environment *E*, which belongs to the state *e*. The behavior *u* of initialized agent defines relation $[u]: E \to E$, which is defined by the relation $[u](e) = e[u]$ and an insertional equivalence of agents $\sim_E$ relative to environment *E*, which is defined by relation $u \sim_E v \Leftrightarrow [u] = [u]$. This equivalence is usually weaker than bisimulation and plays main role in the applications, because a transformation of algorithms and software implementations of the agents which live in some environment should be executed as transformation which saves insertional equivalence.

In [17] some classification of the insertion functions and the obtained results on a reduction of the complex class of functions to the simple ones are presented.

4 Insertion Modeling System

Insertion modeling system [21] is an environment for the development of insertion machines and performing experiments with them. The notion of insertion machine was used as a tool for programming with some special class of insertion functions. Later this notion was extended for wider area of applications, different levels of abstraction, and multilevel structures.

Insertion model of a system represents this system as a composition of environment and agents inserted into it. Contrariwise the whole system as an agent can be inserted into another environment. In this case we talk about internal and external environment of a system. Agents inserted into the internal environment of a system themselves can be environments with respect to their internal agents. In this case we talk about multilevel structure of agent or environment and about high level and low level environments.

The general architecture of insertion machine is represented in Figure 1.

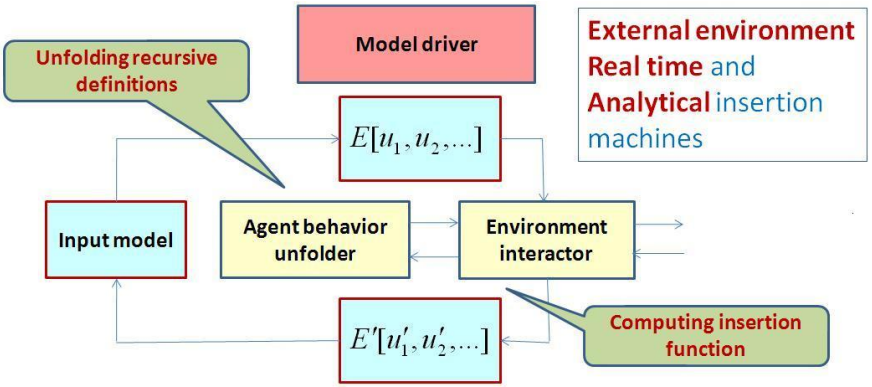


Figure 1. Architecture of Insertion Machine

The main component of insertion machine is model driver, the component which controls the machine movement along the behavior tree of a model. The state of a model is represented as a text in the input language of insertion machine and is considered as an algebraic expression. The input language includes the recursive definitions of agent behaviors, the notation for insertion function, and possibly some compositions for environment states. Before computing insertion function the state of a system must be reduced to the form $E[u_1,u_2,...]$. This functionality is performed by the module called agent behavior unfolder. To make the movement, the state of environment must be reduced to the normal form

$$\sum_{i \in I} a_i.E_i + \varepsilon ,$$

where a_i are actions, E_i are environment states, ε is a termination constant. This functionality is performed by the module environment interactor. It computes the insertion function calling if it is necessary the agent behavior unfolder. If the infinite set I of indices in the normal form is allowed, then the weak normal form $a.F+G$ is used, where G is arbitrary expression of input language.

Two kinds of insertion machines are considered: *real time* or *interactive* and *analytical* insertion machines. The first ones exist in the real or virtual environment, interacting with it in the real or virtual time. Analytical machines are intended for model analyses, investigation of its

properties, solving problems etc. The drivers for two kinds of machines correspondingly are also divided into interactive and analytical drivers.

Interactive driver after normalizing the state of environment must select exactly one alternative and perform the action specified as a prefix of this alternative. Insertion machine with interactive driver operates as an agent inserted into external environment with insertion function defining the laws of functioning of this environment.

Analytical insertion machine as opposed to interactive one can consider different variants of making decision about performed actions, returning to choice points (as in logic programming) and consider different paths in the behavior tree of a model. The model of a system can include the model of external environment of this system, and the driver performance depends on the goals of insertion machine. In the general case analytical machine solves the problems by search of states, having the corresponding properties (goal states) or states in which given safety properties are violated. The external environment for insertion machine can be represented by a user who interacts with insertion machine, sets problems, and controls the activity of insertion machine.

Analytical machine enriched by logic and deductive tools is used for generating traces of symbolic models of systems. The state of symbolic model is represented by means of properties of the values of attributes rather than their concrete values.

General architecture of insertion modeling system is represented in Figure 2. High level model driver provides the interface between the system and external environment including the users of the system. Design tools based on Algebraic Programming system *APS*[21] are used for the development of insertion machines and model drivers for different application domains and modeling technologies. Verification tools are used for the verification of insertion machines, proving their properties statically or dynamically. Dynamic verification uses generating symbolic model traces by means of special kinds of analytical model drivers and deductive components.

The repository of insertion machines collects already developed machines and their components which can be used for the development of

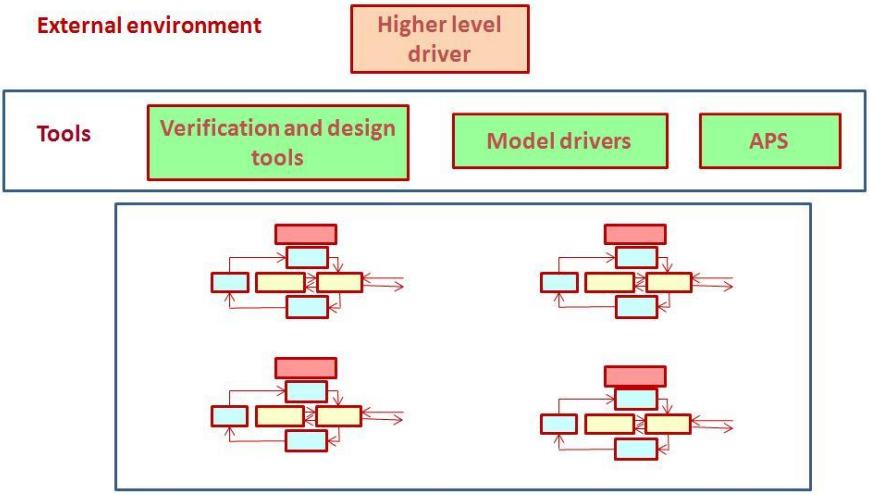


Figure 2. Architecture of Insertion Modeling System

new machines as their components or templates for starting. Special library of APLAN functions supports the development and design in new projects. The C++ library for *IMS* supports *APLAN* compilers and efficient implementation of insertion machines. Deductive system provides the possibility of verification of insertion models [22].

5 Applications

Based on the ideas of insertion modeling the Verification of Requirement Specification (VRS) system was developed by researchers from V.M. Glushkov Institute of Cybernetics of National Academy of Science of Ukraine.

The language of basic protocols is implemented in VRS, which supports the usage of numerical attributes and symbolic types, arrays, lists and functional data types. The deductive system provides proof of the identities in the theory of the first order logic, which is the integration of theories of real and integer linear inequalities, free uninterpreted function symbols and theory query. Symbolic modeling in the VRS is based on satisfiability checking and predicate transformer functions[23].

Proving Programming System is a new and modern system that is designed to maintain a high level of training of qualified specialists in

programming. This system is created based on the Insertion Modeling system and Algebraic Programming System which was developed at the V.M. Glushkov Institute of Cybernetics of NAS of Ukraine with the participation of authors of Kherson State University. This system implements Floyd's algorithm of proving partial correctness of annotated programs[24].

Insertion Modeling system was successfully used for implementation of theory for building of invariants of the models[25] and loops in software[26], for the set of school computer algebra systems[27], for interleaving reduction in symbolic insertion models[28].

6 Conclusion

In this paper the main notions of insertion modeling are given. Insertion modeling theory is one of the most general theory of process algebra. The main difference of it is that an environment is considered as an agent with insertion function. Insertion Modeling System was developed for supporting this theory in practice and is used for developing industrial and research insertion machines.

In the nearest future we are planning to use such theory and system for research of models which came from law and economics.

References

[1] D.R. Gilbert, A.A. Letichevsky. *A universal interpreter for nondeterministic concurrent programming languages*. In: M. Gabbrielli (Ed.), Fifth Compulog network area meeting on language design and semantic analysis methods, September 1996.

[2] A. Letichevsky and D. Gilbert. *Interaction of agents and environments*. In: Resent trends in Algebraic Development technique, LNCS 1827 (D. Bert and C. Choppy, eds.), Springer-Verlag, 1999.

[3] V.M. Glushkov. *The automata theory and design issues digital machines structures*. Cybernetics, Vol. 1 (1965), pp. 3-11.

[4] V. M. Glushkov and A. A. Letichevsky. Theory of algorithms and descrete processors. Advances in Information Systems Science (J. T. Tou, ed.), vol. 1, Plenum Press, 1969, 1-58.

[5] M. Kwan. *The design of the ICE encryption algorithm.* The 4th International Workshop, Fast Software Encryption - FSE '97 Proc. LNCS, vol. 1267 (1997), pp. 69-82.

[6] Y.V. Kapitonova, A.A. Letichevsky. *The mathematical theory of digital systems*. Moscow, Science, 1988, 295p.

[7] A. Letichevsky, J. Kapitonova, A. Letichevsky Jr., V. Volkov, S. Baranov, V.Kotlyarov, T. Weigert. *Basic Protocols, Message Sequence Charts, and the Verification of Requirements Specifications*. ISSRE 2004, WITUL (Workshop on Integrated reliability with Telecommunications and UML Languages), Rennes, 4 November 2005.

[8] S. Baranov, C. Jervis, V. Kotlyarov, A. Letichevsky, and T. Weigert. *Leveraging UML to deliver correct telecom applications in UML for Real*. Design of Embedded Real-Time Systems by L.Lavagno, G. Martin, and B. Selic (editors), 323–342, Kluwer Academic Publishers, 2003.

[9] J. Kapitonova, A. Letichevsky, V. Volkov, and T. Weigert. *Validation of Embedded Systems*. In R. Zurawski, editor. The Embedded Systems Handbook. CRC Press, Miami, 2005.

[10] A.A.Letichevsky, J.V.Kapitonova, V.A.Volkov, A.A.Letichevsky, jr., S.N.Baranov, V.P.Kotlyarov, T.Weigert. *System Specification with Basic Protocols*. Cybernetics and System Analyses, Vol. 4, 2005.

[11] R. Milner. *A Calculus of Communicating Systems*. LNCS, Vol. 92 (1980).

[12] R. Milner. *Communication and Concurrency*. Prentice Hall, 1989.

[13] R. Milner. *The polyadic π-calculus: a tutorial*. Tech. Rep. ECS–LFCS–91–180, Laboratory for Foundations of Computer Science, Department of Computer Science, University of Edinburgh, UK (1991).

[14] C. A. R. Hoare. *Communicating Sequential Processes*. Prentice Hall, 1985.

[15] J. A. Bergstra and J. W. Klop. *Process algebra for synchronous communications*. Information and Control, Vol. 60 (1/3) (1984), pp. 109-137.

[16] J. A. Bergstra, A. Ponce, and S. A. Smolka, eds. *Handbook of Process Algebra*. North- Holland, 2001.

[17] A.Letichevsky. *Algebra of behavior transformations and its applications*. In V.B.Kudryavtsev and I.G.Rosenberg eds. Structural theory of Automata, Semigroups, and Universal Algebra, NATO Science Series II. Mathematics, Physics and Chemistry, Vol. 207 (2005), pp. 241-272.

[18] D. Park. *Concurrency and automata on infinite sequences*. LNCS, Vol. 104 (1981).

[19] R. J. Glabbeek. *The linear time-branching time spectrum the semantics of concrete, sequential processes*. Handbook of Process Algebra (J. A. Bergstra, A. Ponce, and S. A. Smolka, eds.), North-Holland, 2001.

[20] M. Roggenbach and M. Majster-Cederbaum. *Towards a unified view of bisimulation: a comparative study*. TCS, Vol. 238 (2000), pp. 1–130.

[21] A.A. Letichevsky, O.A.Letychevskyi, V.S. Peschanenko. *Insertion Modeling System*. LNCS, Vol. 7162 (2011), pp. 262-274.

[22] A.A. Letichevsky, O.A. Letychevskyi, V.S. Peschanenko. *Algebraic Programming System APS and Insertion Modeling System IMS*, 2016. http://www.apsystems.org.ua.

[23] A. Letichevsky, O. Letychevskyi, V. Peschanenko, T. Weigert. *Insertion Modeling and Symbolic Verification of Large Systems*. Lecture Notes in Computer Science, Vol. 9369(2015), pp. 3-18.

[24] O. Letychevskyi, M. Morokhovets, V. Peschanenko. *System of Provable Programming*. /Control Systems and Computers, Vol. 6,(2012), pp. 64-71 (in russian).

[25] A. Letichevsky, A. Godlevsky, A. Guba, A. Kolchin, O. Letichevkyi, V. Peschanenko. *Usage of Invariants for Symbolic Verification of Requirements*. Risc-Linz report series, No. 13-06(2013), pp. 124-124.

[26] M.S. Lvov. *A Method of Proving the Invariance of Linear Inequalities for Linear Loops*. Cybernetics and Systems Analysis, Volu. 50, Issue 4(2014), pp. 643-648.

[27] *National Projects of Kherson State University*, 2016.

http://www.kspu.edu/About/DepartmentAndServices/DSAICI/internationalprojects/NationalProjects.aspx?lang=en.

[28] A. Letichevsky, O. Letychevskyi, V. Peschanenko. *An Interleaving Reduction for Reachability Checking in Symbolic Modeling*. In: Ermolayev, V. et al. (eds.) Proc. 11-th Int. Conf. ICTERI 2015, Lviv, Ukraine, May 14-16, 2015, CEUR-WS.org/Vol-1356, ISSN 1613-0073, 338-353, online ceur-ws.org/Vol-1356/paper_74.pdf.

Alexander Letichevsky[1], Oleksandr Letychevskyi[2], Vladimir Peschanenko[3]

[1]V.M. Glushkov Institute of Cybernetics of National Academy of Sciences of Ukraine
E-mail: aaletichevsky78@gmail.com
[2]V.M. Glushkov Institute of Cybernetics of National Academy of Sciences of Ukraine
E-mail: lit@iss.org.ua
[3]Kherson State University of Ukraine
E-mail: vladim@ksu.ks.ua

Proceedings of the Conference on Mathematical Foundations of Informatics
MFOI2016, July 25-29, 2016, Chisinau, Republic of Moldova

Agents in a Network Environment: Models and Methods (A Survey)

Volodymyr G. Skobelev, Volodymyr V. Skobelev

Abstract

The given paper presents some short survey of models and methods used in the agent-based approach to investigation of network environments. For understanding the value and the aim of this approach some short retrospective analysis is given. Bio-inspired agents-based meta-heuristics are presented briefly. Basic properties of software agents and features intended to distinguish different types of agents are considered. Variants of the graph exploration problem are characterized. Some applications of the agent-based approach are listed.

Keywords: networks, mobile agents, bio-inspired meta-heuristics, graph exploration

1 Introduction

Possibly, the first description of agent's behavior in a net environment has been presented in the ancient Greek legend "Ariadne's thread". This legend can be interpreted as follows.

Given maze is some finite anonymous graph, i.e. vertices are unlabeled, and edges are locally labelled at each vertex (the last condition gives the possibility to distinguish locally edges at a node). Theseus (mobile agent) and Minotaur (motionless goal) are located in two distinct vertices of this graph. Theseus has a thread the length of which is not less than the height of any spanning tree with the root at the vertex he occupies at initial instant (in the legend this hypothesis is

supposed implicitly). Using this thread Theseus applies backtracking to find the vertex in which Minotaur is located.

The first finite automaton algorithm for exploring any maze (with the range of 5×5 squares) via trial and error method has been designed in [1]. Within the next thirty years solvability of graph exploration problem via finite automata has been studied under different assumptions. Main results were established in [2–6]. Within the next decade behavior of automata in labyrinths [7, 8], models and methods for distributed problem solving [9, 10] and for analysis of social, economical and technical systems [11–14] has been investigated intensively. In the last decade of the XX century the emergence of modern information technologies, and especially the spread of large-scale networks, have outlined actuality of development models and methods intended for design and analysis of such complex systems as communication networks, networked systems (i.e. collections of dynamic units that interact over some information exchange network), critical infrastructures (i.e. physical and virtual assets, processes, facilities that provide fundamental services and play a vital role in the country's economy, health and security), etc.

These researches had significant impact on rapid emergence of "agent-based simulation" as some paradigm for investigation of complex systems [15, 16]. In particular, there has began formation of agent-oriented modeling with graph transformation [17, 18] and agent-oriented software engineering based on agents interaction analysis [19, 20]. As a result, some backgrounds for theory and applications of multi-agent systems (i.e. computational systems in which some agents interact or work together to perform some set of tasks or to achieve some common goals) has been laid at the end of the XX century [21–23].

An important part in this field of research forms the area known as "agents in a network environment". In what follows we consider some basic models and methods associated with it.

2 Bio-inspired agents-based meta-heuristics

These meta-heuristics form an important part of bio-inspired computing [24–26]. They are based on some advantages of collective behavior in real world, and are intended to solve complex and hard problems.

Possibly, the most known agents-based optimization meta-heuristic is ant colony optimization (ACO). It has been proposed in [27, 28], and the traveling salesman problem has been selected as the test case. ACO is based on the following idea.

Investigated optimization problem is transformed into the problem of finding the best path on some weighted graph. A set of software agents (ants) incrementally build solutions by moving on this graph. This process is stochastic and is based on a pheromone model, i.e. on some set of parameters associated with graph components (either nodes or edges) whose values are modified at runtime by the ants. Some pheromone is sequentially deposited by each ant on its path. More pheromone on a path increases probability for this path being followed. In the result some shortest designed path can be discovered via pheromone trails.

ACO applications to a wide range of networks analysis problems has been developed (some surveys are presented in [29–32]), such as routing in Internet-like networks [33], extracting communities in large-scale networks [34], file sharing in mobile networks [35], etc. It should be noted the following essential progress in ACO development:

1) ACO variants for dynamically changing problems [36, 37];
2) adaptive ACO variants [38, 39];
3) cooperative ACO variants [40, 41];
4) ACO parallel strategies [42, 43];
5) ACO variants for multilevel framework [44, 45].

Another agents-based optimization meta-heuristic for solving hard combinatorial problems is artificial bee colony (ABC) system. It has been proposed in [46]. ABC consists of employed bees, scouts and onlookers. The following actions are performed in each iteration.

Any employed bee analyzes some neighborhood of its current position with the aim to find the best next position. These bees memorize previously visited positions and their quality. Any scout chooses randomly some new position uncovered by employed bees. If new position is better than the previous one, then scout memorizes it and forgets its current position. Onlookers decide which positions are better on the base of information provided by employed bees and scouts.

Thus, ABC system combines local search, carried out by employed bees and onlookers, with global search, managed by onlookers and scouts. It should be noted that actions of scouts are intended to avoid searching to get trapped in local extremum.

ABC systems applications to a wide range of networks analysis problems have been developed, such as graph search problems [47], CrossOver design [48], wireless sensor network routing [49]. It should be noted the following essential progress in ABC system development:

1) parallel ABC system variants [50];
2) distributed ABC system variants [47];
3) adaptive ABC system variants [51].

The state of the art for ABC systems is presented in [52–55].

Above considered meta-heuristics had a significant impact on formation and development of the swarm intelligence [56, 57]. This approach is based on the following two principles:

1. Summary self-organization principle, i.e. activity amplification by positive feedback, activity balancing by negative feedback, amplification by random fluctuations, and multiple interactions.
2. Stigmergy (stimulation by work) principle, i.e. work as behavioural response to the environmental state, an environment that serves as a work state memory, and work that does not depend on specific agents (in the sense that work can be continued by any agent).

The state of the art of swarm intelligence is presented in [58–61].

3 Software agents

As it was noted in [62–64] any such agent is some hardware or software based computer system with at least the following properties:

1) autonomy, i.e. there are some means which provide the ability to operate without external interference, as well as to control its own actions and internal state;
2) social ability, i.e. there is some possibility to interact with other agents via some kind of agent-communication language;
3) reactivity, i.e. perception of the environment (via an interface, some set of other agents, the Internet and so on) and responding in a timely fashion to changes that occur in it;
4) proactivity, i.e. ability to carry out some goal-directed behaviour by taking the initiative.

It should be noted that reactivity and proactivity are precisely those two components, on the basis of which the ability to make decisions is provided for an agent.

To distinguish different types of software agents at least the following four features are commonly used:

1. Agent's mobility. An agent can be either static (i.e. it is located at the same place permanently), or mobile (i.e. it can change its location). Advantages of a mobile agent are evident when it is essential to use strong interactivity between components or special computing facilities.

It should be noted that providing agent's mobility can be closely linked with providing code mobility for different operating systems and computer networks.

2. Agent's intelligence level. This feature is informal and is closely related to the concept "agent's memory capacity". Starting from this concept the following two extreme cases can be highlighted naturally.

The first extreme case occurs when an agent has very restricted memory capacity, which is sufficient to perform only a very limited

number of the most primitive actions. In this case it is natural to refer that an agent does not possess intelligence at all.

Examples of these agents are scouts in ABC system and ants in classic ACO variant.

The second extreme case occurs when an agent has large memory capacity, which enables him to solve investigated problem without interaction with other agents. In this case it is natural to refer that we deal with intelligent agent.

Examples of these agents are expert systems and solvers used for solving any specific problem.

In any remaining case an agent has certain not very large memory capacity, which enables it to remember some set of pre-histories. On this base an agent can perform a certain range of non-trivial tasks. However, in order to solve successfully investigated problem any such agent needs to interact with other agents included in the given multi-agent system. In this case it is natural to refer that we deal with an agent, which has certain level of intelligence.

Examples of these agents are employed bees and onlookers in ABC system.

Numerous attempts have been made to develop measures for ordering these cases [65–67]. But until now there is no generally accepted approach for measuring agent's intelligence level, as well as any other agent's feature.

3. Agent's adaptation capability. This feature is intended to ensure agent's safe operation under changes in conditions of its work and/or in the environment. Perhaps one of the simplest examples of agent's adaptation capability is Glushkov's composition of control automaton (an agent) and operating automaton (an environment) [68]. It should be mentioned that agent's adaptation capability is a necessary feature for its robustness in the presence of unpredictable changes in any dynamic environment.

4. Learning ability. This feature is intended to reinforce agent's performance via modifying over time its behaviour. Learning can be done online (i.e. via data mining from data which are constantly collected

through interaction with users) or off-line (i.e. via pattern recognition prior to productive agent usage). Some surveys of existing approaches for agents learning are presented in [69–71].

It should be noted that in terms of the above considered features the notion "swarm intelligence" with reference to "agents in a network environment" can be described as follows. Some mobile software agents are utilized for network analysis or management. These agents are autonomous entities, both proactive and reactive, use communication through the environment, have the capability to adapt, to cooperate and to move intelligently from one location to the other in the network.

4 The graph exploration problem

Informally speaking "graph exploration" means that for initially unknown connected graph it is necessary to visit either every vertex, or every edge by agents being physically located in the graph (the last case is referred to as design of the map of the graph). This is one of the basic problems in mobile agents' paradigm, since it can be used for solving wide range of complex applied problems.

In order to solve this problem some requirements must be carried out [72]. In particular, any agent must have possibility to distinguish adjacent edges while visiting any vertex of the graph. For this reason it is supposed that there is local port labeling at each vertex, and this labelling is available for an agent visiting this vertex. It should be noted that in the absence of such unique labels, the task becomes much more difficult, but it can still be solved if agents are supplied with some means for marking the nodes or edges in the process of exploration.

The graph exploration problem has been studied under various assumptions due to investigated graph and to agents exploring it. Main of these assumptions are the following ones:

1. The unknown graph is either undirected [73–76], or directed [77–79].

2. Either deterministic [80,81] or probabilistic [82-84] approach is used in the investigation.

3. Either single agent or several agents are used.

4. Either the nodes of the graph are labelled with unique identifiers, or are unlabelled. In the last case we deal with anonymous graph.

5. Either agents must return to the starting points or not. In the last case two versions have been considered: exploration with stop (i.e. when each agent stops after completing exploration) and the Rendezvous problem (i.e. after completing exploration by each agent all agents must be gathered at a single node).

6. Either there are restrictions on agents memory size (measured in bits, or the number of states if finite automaton is used in the role of an agent), or not.

Dealing with the graph exploration problem the effectiveness of the proposed algorithms is measured via either the completion time of the task, measured in terms of the number of moves (edge traversals) [74], or the amount of memory (operational memory) used by agents [85, 86].

Regarding operational memory the following three cases have been investigated:

1. Agents with unbounded memory. The main aim, as a rule, is to minimize the time of completion of algorithm.

2. Agents with bounded memory. The main aim, as a rule, is the feasibility of performing the task under given memory constraint or the tradeoff between time and memory.

3. Agents with no memory (oblivious agents). The main aim, as a rule, is the feasibility of performing the given task.

The following additional assumptions has been used for investigation of the graph exploration problem:

1. Some global parameters of investigated graph can be given to agents, such as the size of the graph, its diameter and so on. This information can significantly impact on the solvability of the task, or can significantly reduce complexity of the task.

2. Capability of agents to interact with each other and with the environment. Two types of interactions have been considered.

Direct interactions include exchange of information by agents located at the same node (local communication) or at arbitrary locations

(global communication). Sometimes it is suggested that agents can write some information on nodes (whiteboards), or can leave movable or immovable tokens.

Indirect interactions consist of communications between agents through the environment. Sometimes environment is formed by onlookers.

3. Synchronization. Any algorithm executed by mobile agents is some sequence of Look-Compute-Move cycles. In any cycle, an agent scans its current position (Look), makes a decision (Compute) to stay idle or to move. The moving is the third phase of the cycle (Move). The following three models for synchronization of these cycles have been investigated.

In synchronous model, agents deal with global clock and in every round all agents execute each phase of each cycle simultaneously.

In semi-synchronous model each agent executes Look-Compute-Move cycle independently at unpredictable time instants.

In asynchronous model delays between each phase of Look-Compute-Move cycle can be arbitrarily long. Thus, agents can move on the base of sufficiently outdated perceptions.

5 Applied problems

Agents based approach in a network environment has been used for solving a wide range of applied and fundamental problems. We list briefly some of them:

1) network resource discovery [87];

2) searching in the WWW [88];

3) applications in the field of E-business [89];

4) design and analysis of sensor networks [90];

5) searching for black-hole faults in a network (i.e. faulty or malicious node in the network such that if an agent enters this node, then it immediately "dies") [91];

6) intrusions analysis in computer networks [92];

7) social networks analysis [93];

8) improving quality of service in vehicular ad hoc networks [94];
9) generalized traveling salesman problem [95].

6 Conclusions

The present paper is some attempt to describe models and methods used in agent-based approach to network environments investigation.

At present this trend of research is rapidly developing and it is closely connected with many other trends in AI and computer science. Just for this reason, in this paper many aspects were not covered, each of which can be the subject of a separate survey. Among them are analysis of labyrinths via collectives of finite automata, models and methods for agents collectives learning, elaboration of formal languages intended for agents interaction, fuzzy approach for agents interaction with network environments, agents-based approach in distributed computing, game-theoretic approach to agents interaction with network environments.

Within the last aspect it should be noted the paper [96]. In this paper it is elaborated and investigated some general discrete non-stationary probabilistic model for interaction of an agent with some layered network environment, which prevents it. The proposed model is presented via composition of two finite probabilistic automata with variable structure. This composition of automata represents some two-persons game, in which the player that makes a move inflicts some damage on its enemy.

Of particular note is also research in insertion simulation [97–99], which now has become a powerful theory and technology for verification and validation of information systems.

References

[1] C. Shannon. *Presentation of a Maze-Solving Machine*. Proc. of 8th Conf. of the J. Macy Jr. Found (Cybernetics), 1951, pp. 173–180.

[2] M.O. Rabin. *Maze Threading Automata*. Seminar talk presented at the University of California at Berkeley, October 1967.

[3] H. Mller. *Endliche Automaten und Labyrinthe*. EIK, No. 4 (1971), pp. 261–264.

[4] W. Coy. *Automata in labyrinths*. LNCS, Vol. 56 (1977), pp. 65–71.

[5] L. Budach. *Automata and labyrinths*. Math. Nachr, Vol. 86, Issue 1 (1978), pp. 195–282.

[6] H. Rollik. *Automaten in Planaren Graphen*. LNCS, Vol. 67 (1979), pp. 266–275.

[7] M. Ejsmont. *Problems in Labyrinths Decidable by Pebble Automata*. EIK. No. 12 (1984), pp. 623–632.

[8] A. Hemmerling. *Labyrinth Problems. Labyrinth-Searching Abilities of Automata*. Teubner-Texte zur Math., Vol. 114. Leipzig, 1989.

[9] G. Smith. *The Contract Net Protocol: High Level Communication and Control in a Distributed Problem Solver*. IEEE Trans. on Computers, No. 12 (1980), pp. 1104–1113.

[10] J.R. Galliers. *A Theoretical Framework for Computer Models of Cooperative Dialogue, Acknowledging Multi-Agent Conflict*. Ph.D. Thesis, Open University, UK, 1988.

[11] W.G. Astley, C.J. Fombrun. *Collective Strategy: Social Ecology of Organizational Environments*. The Academy of Management Review, No. 8 (1983), pp. 576–587.

[12] W. Weidlich, G. Haag. *Concepts and Models of a Quantitative Sociology*. Berlin: Springer Verlag, 1983.

[13] P.W. Anderson, K.J. Arrow, D. Pines. *The Economy as an Evolving Complex System*. Redwood City, CA: Addison-Wesley, 1988.

[14] D.M. Dilts, N.P. Boyd, H.H. Whorms. *The Evolution of Control Architectures for Automated Manufacturing Systems*. J. of Manufacturing Systems, No. 1 (1991), pp. 79–93.

[15] P. Maes. *Agents that Reduce Work and Information Overload*. Communications of the ACM, No. 7 (1994), pp. 31–40.

[16] *Agent Technology: Foundations, Applications, and Markets* / Eds. N.R. Jennings, M.J. Wooldridge. Berlin: Springer-Verlag, 1998.

[17] P. Knirsch, H.J. Kreowski. *A Note on Modeling Agent Systems by Graph Transformation*. LNCS, Vol. 1779 (1999), pp. 79–86.

[18] R. Depke, R. Heckel, J.M. Kster. *Agent-Oriented Modeling with Graph Transformation*. LNCS, Vol. 1957 (2000), pp. 105–120.

[19] C. Guilfoyle, E. Warner. *Intelligent Agents: The New Revolution in Software*. Ovum Report, 1994.

[20] M. Wooldridge. *Agent-Based Software Engineering*. IEE Proc. on Software Engineering, No. 1 (1997), pp. 26–37.

[21] N.R. Jennings, K. SYCARA, M. Wooldridge. *A Roadmap of Agent Research and Development*. Autonomous Agents and Multi-Agent Systems, No. 1 (1998), pp. 7–38.

[22] *Multi-Agent Systems: a Modern Approach to Distributed Artificial Intelligence* Ed. G. Weiss. MIT Press, 1999.

[23] M. J. Wooldridge. *Introduction to Multiagent Systems*. John Wiley&Sons, Inc., 2001.

[24] S. Binitha, S. Sathya. *A Survey of Bio Inspired Optimization Algorithms*. Int. J. of Soft Computing and Engineering, Vol. 2, Issue 2 (2012), pp. 137–151.

[25] I. Fister Jr., X.S. Yang, I. Fister, at al. *A Brief Review of Nature-Inspired Algorithms for Optimization*. Electrotehniški Vestnik (English Edition), No. 3 (2013), pp. 1–7.

[26] C. Rajan, K. Geetha, C.R. Priya, at al. *Investigation on Bio-Inspired Population Based Metaheuristic Algorithms for Optimization Problems in Ad Hoc Networks*. Int. J. of Math., Comput., Phys., Electr. and Computer Eng., No. 3 (2015), pp. 159–166.

[27] M. Dorigo, V. Maniezzo, A. Colorni. *Positive Feedback as a Search Strategy*. Techn. Rep. 91-016. Dipartimento di Elettronica, Politecnico di Milano, IT, 1991.

[28] M. Dorigo. *Optimization, Learning and Natural Algorithms* (in Italian). PhD Thesis, Dipartimento di Elettronica e Informazione, Politecnico di Milano, IT, 1992.

[29] C. Blum. *Ant Colony Optimization: Introduction and Recent Trends*. Physics of Life Reviews. Vol. 2 (2005), pp. 353–373.

[30] I. Benyahia. *A Survey of Ant Colony Optimization Algorithms for Telecommunication Networks*. Int. J. of Appl. Metaheuristic Computing, Vol. 3, Issue 2 (2012), pp. 18–32.

[31] A. Pandey, A.K. Singh. *Ant Colony Optimization Based Routing Algorithm in Various Wireless Sensor Network. A Survey*. J. of Adv. Comput. and Communicat. Techn., Vol. 3, Issue 4 (2015), pp. 79–83.

[32] S. Vyas, S. Sanadhya. *A Survey of Ant Colony Optimization with Social Network*. Int. J. of Computer Appl., No. 9 (2014), pp. 17–21.

[33] R. Schoonderwoerd, O. Holland, J. Bruten, at al. *Ant Based Load Balancing in Telecommunications Networks*. Adaptive Behavior, No. 5 (1996), pp. 169–207.

[34] Y. Liu, J. Luo, H. Yang, L. Liu. *Finding Closely Communicating Community Based on Ant Colony Clustering Model*. Proc. of Int. Conf. on Artif. Intell. and Comput. Intelligence, Sanya, China, Oct. 23-24, 2010, Vol.3, pp. 127–131.

[35] K. Borkar, V. Sahare. *A Survey on File Sharing in Mobile Network by using ACO*. Int. J. of Adv. Res. in Computer Sci. and Software Eng., Vol. 5, Issue 1 (2015), pp. 619–623.

[36] D. Angus, T. Hendlass. *Ant Colony Optimisation Applied to a Dynamically Changing Problem*. Proc. of the 15th Int. Conf. on Industrial and Engineering, Appl. of Artif. Intell. and Expert Syst., Cairns, Australia, June 17-20, 2002, pp. 618–627.

[37] C. Eyckelhof, M. Snoek, M. Vof. *Ant Systems for a Dynamic TSP: Ants Caught in a Traffic Jam*. LNCS, Vol. 2463 (2002), pp. 88–99.

[38] G. Di Caro. *Ant Colony Optimization and its Application to Adaptive Routing in Telecommunication Networks*. PhD Thesis, Brussels, Belgium: IRIDIA, Université Libre de Bruxelles, 2004.

[39] Y. Liu, L. Liu, J. Luo. *Adaptive Ant Colony Clustering Method Applied for Finding Closely Communicating Community*. J. of Networks, No. 2 (2012), pp. 249–258.

[40] M. Dorigo, V. Maniezzo, A. Colorni. *The Ant System: Optimization by a Colony of Cooperating Agents*. IEEE Trans. on Systems, Man, and Cybernetics-Part B, No. 1 (1996), pp. 29–41.

[41] M. Dorigo, L.M. Gambardella. *Ant Colony Sytem: A Cooperative Learning Approach to the Travelling Salesman Problem*. IEEE Trans. on Evolutionary Computation. No. 1 (1997), pp. 53–66.

[42] T. Stützle. *Parallelization Strategies for Ant Colony Optimization*. LNCS, Vol. 1498 (1998), pp. 722–731.

[43] M. Pedemonte, S. Nesmachnow, H. Cancela. *A Survey on Parallel Ant Colony Optimization*. Appl. Soft Comput., No. 8 (2011), pp. 5181–5197.

[44] P. Korošec, J. Šilc, B. Robič. *Solving the Mesh-Partitioning Problem with an Ant-Colony Algorithm*. Parallel Comput., Vol. 30 (2004), pp. 785–801.

[45] Z. Qiang, S. Yuqiang, C. Yang. *A Clustering Algorithm Based on Multi-agent Meta-heuristic Architecture*. Int. J. of Hybrid Information Technology, No. 2 (2014), pp. 227–236.

[46] D. Karaboga. *An Idea Based on Honey Bee Swarm for Numerical Optimization*. Techn. Rep. TR06, Erciyes Univ. Press, 2005.

[47] S. Ilie. *Survey on Distributed Approaches to Swarm Intelligence for Graph Search Problems*. Annals of the Univ. of Craiova, Math. and Computer Sci. Series, No. 2 (2014), pp. 251–270.

[48] S. Kumar, V.K. Sharma, R. Kumari. *Artificial Bee Colony Algorithm and Its Application to Generalized Assignment Problem*. Int. J. of Computer Applications, No. 8 (2013), pp. 18–25.

[49] S. Okdem, D. Karaboga, C. Ozturk. *An Application of Wireless Sensor Network Routing Based on Artificial Bee Colony Algorithm*. Proc. of IEEE CEC'2011, pp. 326–330.

[50] H. Narasimhan. *Parallel Artificial Bee Colony (PABC) Algorithm*. Proc. of NaBIC'09, IEEE, 2009, pp. 306–311.

[51] W. Yu, J. Zhang, W. Chen. *Adaptive Artificial Bee Colony Optimization*. Proc. of GECCO'13, July 6-10, 2013, Amsterdam, The Netherlands, pp. 153–157.

[52] A.L. Bolaji, A.T. Khader, M.A. Al-Betar, at al. *Artificial Bee Colony Algorithm, its Variants and Applications: a Survey*. J. of Theor. and Appl. Information Techn., No. 2 (2013), pp. 434–459.

[53] J.C. Bansal, P.K. Singh, M. Saraswat, et al. *Artificial Bee Colony Algorithm: a Survey*. Int. J. of Advanced Intelligence Paradigms No. 1-2 (2013), pp. 123–159.

[54] D. Teodorović, M. Šelmić, T. Davidović. *Bee colony optimization. Part I: The algorithm overview*. Yugoslav J. of Op. Res. No. 1 (2015), pp. 33–56.

[55] D. Teodorović, M. Šelmić, T. Davidović. *Bee Colony Optimization. Part II: The Application Survey*. Yugoslav J. of Op. Res. No. 2 (2015), pp. 185–219.

[56] J. Kennedy, R. Eberhart. *Particle Swarm Optimization*. Proc. of IEEE Int. Joint Conf. on Neural Networks, 1995, pp. 1942–1948.

[57] E. Bonabeau, M. Dorigo, G. Théraulaz. *Swarm Intelligence: From Natural to Artificial Systems*. Oxford Univ. Press, 1999.

[58] M. Saleem, G. Di Caro, M. Farooc. *Swarm Intelligence Based Routing Protocol for Wireless Sensor Networks: Survey and Future Directions*. Int. J. of Information Sci., Vol. 181, Issue 20 (2011), pp. 4597–4624.

[59] X.S. Yang. *Efficiency Analysis of Swarm Intelligence and Randomization Techniques*. http://arxiv.org/pdf/1303.6342.pdf

[60] S. Vanitha, T. Padma. *A Survey on Swarm Intelligence Algorithms*. Int. J. of Computer Sci. and Mobile Computing, Vol. 3, Issue 5 (2014), pp. 994–998.

[61] S. Roy, S. Biswas, S.S. Chaudhuri. *Nature-Inspired Swarm Intelligence and Its Applications*. Int. J. of Modern Education and Computer Sci., No. 12, (2014), pp. 55–65.

[62] E. Kranakis, D. Krizanc, S. Rajsbaum. *Mobile agent rendezvous: A survey*. LNCS, Vol. 4056 (2006), pp. 1–9.

[63] C. Bădică, Z. Budimac, H. Burkhard, at al. *Software Agents: Languages, Tools, Platforms*. Computer Sci. and Information Syst., No. 2 (2011), pp. 255–296.

[64] J. Tewari, S. Arya, P.N. Singh. *Approach of Intelligent Software Agents in Future Development*. Int. J. of Adv. Res. in Computer Sci. and Software Eng., Vol. 3, Issue 5 (2013), pp. 794–799.

[65] D. Franklin, A. Abrao. *Measuring Software Agents Intelligence.* Proc. of Int. Conference: Advances in Infrastructure for Electronical Business, Science and Education on the Internet, L'Aquila, Italy, August, 2000.

[66] S. Legg, M. Hutter. *Universal Intelligence: A Definition of Machine Intelligence.* Minds and Machines, No. 4 (2007), pp. 391–444.

[67] S. Mahar, P.K. Bhatia. *Measuring the Intelligence of Software Agent.* Int. J. of Innovative Sci., Eng. & Techn., Issue 6 (2014), pp. 1–11.

[68] V.M. Glushkov. *Synthesis of digital automata.* Moskow, Nauka, 1962. [in Russian]

[69] H.J. van den Heric, D. Hennes, M. Kaisers, at al. *Multi-Agent Learning Dyamics: A Survey.* LNAI, Vol. 4676 (2007), pp. 36–56.

[70] L. Buşoniu, R. Babuška, B. De Schutter. *A Comprehensive Survey of Multi-Agent Reinforcement Learning.* IEEE Trans. on Syst., Man, and Cybernetics. Part C. No. 2 (2008), pp. 156–172.

[71] D. Bloembergen, K. Tuyls, D. Hennes, at al. *Evolutionary Dynamics of Multi-Agent Learning: A Survey.* J. of Artificial Intelligence Research, Vol. 53 (2015), pp. 659–697.

[72] M. Yamashita, T. Kameda. *Computing on Anonymous Networks: Part I - Characterizing the Solvable Cases.* IEEE Trans. on Parallel and Distributed Systems, No. 1 (1996), pp. 69–89.

[73] B. Awerbuch, M. Betke, R. Rivest, at al. *Piecemeal Graph Learning by a Mobile Robot.* Inform. Comput., No. 2 (1999), pp. 155–172.

[74] P. Panaite, A. Pelc. *Exploring unknown undirected graphs.* J. of Algorithms, No. 2 (1999), pp. 281–295.

[75] P. Fraigniaud, L. Gasieniec, D. Kowalski, at.al. *Collective Tree Exploration.* LNCS, Vol. 2976 (2004), pp. 141–151.

[76] A. López-Ortiz, S. Schuierer. *On-Line Parallel Heuristics, Processor Scheduling and Robot Searching Under the Competitive framework*. Theoret. Comput. Sci., Vol. 310 (2004), pp. 527–537.

[77] P. Fraigniaud, D. Ilcinkas. *Directed Graphs Exploration with Little Memory*. LCNS, Vol. 1996 (2004), pp. 246–257.

[78] S. Albers, M.R. Henzinge. *Exploring Unknown Environments*. SIAM J. of Comput., Vol. 29 (2000), pp. 1164–1188.

[79] M. Bender, A. Fernandez, D. Ron, A. Sahai, at al. *The Power of a Pebble: Exploring and Mapping Directed Graphs*. Inform. Comput., No. 1 (2002), pp. 1–21.

[80] A. Dessmark, P. Fraigniaud, D. R. Kowalski, at al. *Deterministic Rendezvous in Graphs*. Algorithmica, No. 1 (2006), pp. 69–96.

[81] A. Pelc. *Deterministic Rendezvous in Networks: Survey of models and results*. LNCS, Vol. 6950 (2011), pp. 1–15.

[82] K. Efremenko, O. Reingold. *How Well do Random Walks Parallelize?*. LNCS, Vol. 5687 (2009), pp. 476–489.

[83] S. Ikeda, I. Kubo, M. Yamashita. *The Hitting and Cover Times of Random Walks on Finite Graphs Using Local Degree Information*. Theor. Computer Sci., No. 1 (2009), pp. 94–100.

[84] C. Cooper. *Random Walks, Interacting Particles, Dynamic Networks: Randomness Can be Helpful*. LNCS, Vol. 6796 (2011), pp. 1–14.

[85] K. Diks, P. Fraigniaud, E. Kranakis, at al. *Tree exploration with little memory*. J. of Algorithms, No. 1 (2004), pp. 38–63.

[86] P. Fraigniaud, D. Ilcinkas. *Directed graphs exploration with little memory*. LCNS, Vol. 1996 (2004), pp. 246–257.

[87] C.R. Dunne. *Using Mobile Agents for Network Resource Discovery in Peer-to-Peer Networks*. ACM: Special Interest Group on Electronic Commerce, Vol. 2.3 (2001).

[88] D.J. Grey, P. Dunne, R.I. Ferguson. *A Mobile Agent Architecture for Searching the WWW*. Proc. of Int. Symposium on Mobile Agent Applications, Baden-Baden, Germany, 2000.

[89] P. Karthikeyan, E. Sathiyamoorthy. *A Survey on Applications of Mobile Agents in E-Business*. Int. J. of Scientific & Engineering Res., Issue 3 (2012), pp. 1–2.

[90] M.A. Batalin, G.S. Sukhatme. *The Design and Analysis of an Efficient Local Algorithm for Coverage and Exploration Based on Sensor Network Deployment*. IEEE TRANS. on Robotics, No. 4 (2007), pp. 661–675.

[91] C. Cooper, R. Klasing, T. Radzik. *Searching for Black-Hole Faults in a Network Using Multiple Agents*. LNCS, Vol. 4305 (2006), pp. 318–330.

[92] K. Maskat, M.A.M. Shukran, M.Ad. Khairuddin, at al. *Mobile Agents in Intrusion Detection System: Review and Analysis*. Modern Applied Science, No. 6 (2011), pp. 218–231.

[93] A.G. Fetta. *Investigating Social Networks with Agent Based Simulation and Link Prediction Methods*. Ph.D. Thesis, Cardiff University, UK, 2014.

[94] R. Kumar, M. Dave. *Mobility Models and Their Affect on Data Aggregation and Dissemination in Vehicular Networks*. Wireless Personal Communications, No. 3 (2014), pp. 2237–2269.

[95] C.M. Pintea. *A Unifying Survey of Agent-Based Approaches for Equality-Generalized Traveling Salesman Problem*. Informatica, No. 3 (2015), pp. 509–522.

[96] V.G. Skobelev. *A Probabilistic Model for the Interaction of an Agent with a Network Environment*. Springer US, Cybernetics and Systems Analysis, No. 6 (2015), pp. 835–848.

[97] A. Letichevsky, D. Gilbert, *A Model for Interaction of Agents and Environments*. LNCS, Vol. 1827 (1999), pp. 311–328.

[98] J. Kapitonova, A. Letichevsky, V. Volkov, at al. *Validation of Embedded Systems*. In R. Zurawski, editor. The Embedded Systems Handbook. CRC Press, Miami, 2005. pp. 6–57.

[99] A.A. Letichevsky, J.V. Kapitonova, V.P. Kotlyarov, at al. *Semantics of Message Sequence Charts*. SDL Forum, 2005, pp. 117–132.

Volodymyr G. Skobelev, Volodymyr V. Skobelev Received May 2, 2016

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

Volodymyr V. Skobelev
V.M. Glushkov Institute of Cybernetics of NAS of Ukraine
40 Glushkova ave., Kyiv, Ukraine, 03187
Phone: +38 063 431 86 05
E-mail: vvskobelev@incyb.kiev.ua

Part 2

Regular papers

Web service transformations in a federated Enterprise Service Bus based on executable choreographies

Sînică Alboaie, Lenuta Alboaie, and Mircea-Florin Vaida

Abstract

The objective of this paper is to use the SwarmESB in the software architecture of the OPERANDO privacy platform, funded by the European Union in a Horizon 2020 project. SwarmESB is an open source Enterprise Service Bus (ESB) based on executable choreographies. We are approaching the concept of service transformations, presented as a bridge between the world of REST web services and the world of services implemented with executable choreographies. Five types of transformations that have been analysed and implemented as open source software have been integrated. This proposal is shaped around a common language capable of expressing all these five transformation types we have identified working for OPERANDO. Therefore, the Domain Specific Language proposed, renders the essential elements for transformations among functions, web services and executable choreographies. This unification will trigger a quantitative effect on the productivity of the teams creating or integrating web services in a federated service bus environment which is a key architectural component in the future Internet-of-Things and cloud systems.

Keywords: middleware, architectures, DSL executable choreographies, web service transformations

1 Introduction

The OPERANDO's [1] architecture presented in this article focuses on the usage of an Enterprise Service Bus (ESB) [2] based on the open source research project SwarmESB [3]. The main goal of the OPERANDO

project is to integrate and extend the existing privacy techniques to create a platform that will be used by independent organisations called Privacy Service Providers (PSPs) to ensure policies compliance regarding privacy laws and regulations. OPERANDO should ensure comprehensive user privacy enforcement in the form of a dedicated online service, called "Privacy Authority". The OPERANDO platform supports flexible and viable business models, including targeting of individual market segments such as public administration, social networks and Internet of Things. We are approaching the concept of service transformations, presented as a bridge between the world of REST web services and the world of services implemented with executable choreographies. Web services can be seen as working on a request/response communication pattern. Executable choreographies [4] can be intuitively seen as arbitrary complex workflows that get executed in systems belonging to multiple organisations or authorities. Executable choreographies are implemented in SwarmESB using the swarm communication idea [5]. Therefore, SwarmESB is a research and engineering effort to implement and adapt ideas specific to the mobile calculus theory . While theoretical research on mobile code [6] and on systems for asynchronous calculus have existed for many years, SwarmESB is a practical approach that can be appealing for the specialists used to program in mainstream languages Java, C#, Java Script and who will not easily switch to research programming languages (actor inspired languages[7], pi calculus[8] et al.).

In SwarmESB, messages have a long time identity during multiple communication events and during complex communication processes. Groups of related messages called swarms change their state after each communication event. In actor model inspired approaches, a message does not have identity or an associated behaviour. Identity, state changes or behaviours are associated only with the message receivers (actors). Associating state, mobile code and behaviour with its own messages is the main difference between swarm communication and the actor model. In the actual implementation message, queues are used and the mobile code is securely deployed on the processing nodes but swarm communication hides all the details of the code migration message queues. Executable choreographies are scripts that get executed in multiple processing nodes which may belong to multiple organisations. Swarm communication

environments can be easily integrated with web services by manually exposing remote endpoints in JavaScript functions. In OPERANDO project, we have decided to automate this process by creating methods of describing web service and providing multiple types of "transformations" between web services and executable choreographies. Choreographies can implement a larger number of communication patterns compared with web services. However, we are currently living in a world of web services and since OPERANDO is a complex project that uses existing components and technologies, we have found mandatory to automate the integration of the web services.

2 ESB middleware's based on choreography - concepts overview

An important concern in Service Oriented Architecture (SOA) [9] is to extract the business processes from the application code and orchestrate the business process grounded on services. When multiple organisations are involved in the same business process, we talk about choreography. When business processes spread over multiple organisations, governance, security and privacy aspects become suddenly critical and have a big influence on the business and technology choices. In OPERANDO, we have chosen to use executable choreography, a concept emerging from our previous research [4, 5, 13], [10]. Executable choreographies propose the existence of a business process description that is aware of the location aspects (which is the organisation). It also unifies short living processes as ESB routing and long living business processes (implemented as an extension to the routing). Executable choreographies are technical descriptions of business agreements among multiple organisations and should be treated as such.

One of the most popular integration methods is the nightly batch processing [11]. However, batch-processing integration strategies are prone to errors caused by multiple data changes on shared resources and are bound to cause delays in information retrieval. An ESB can eliminate many latency problems by providing real-time throughput of the data flows among applications and organisations. This real-time flow of data requires support for data transformations [12]. From the development process point of view, an ESB can be seen as the foundation of a SOA

architecture that may enable an agile style of working. Agile main goal of reducing waste is accomplished by lowering the need of complex ad-hoc architectures. The development team can understand the big picture from an early stage and actively contribute to defining the services scope and detailed requirements.

Executable choreographies that should be executed by multiple organisations will be manually or automatically verified and approved each time they get updated. Any ESB allows parallel development of integrated services, reducing the need of stubs or fake service implementation during development. The missing services can be simulated within the integration scripts (e.g. executable choreographies). The integration scripts as executable artefacts of the short/long living processes may be independently developed by each team which implements different services. Different versions of the choreographies can be merged at any time, usually without requiring any changes in the service implementation.

The typical ESB roles include connectivity, routing, transformations and various methods to represent short or long living business processes (integrations, orchestration or choreographies) [13].

Connectivity is the basic feature for any Service Bus. An ESB reduces the configuration efforts because the producers will send information only towards BUS and do not have to be aware of consumers.

Routing: beside connectivity, if integration is a subject of interest, the necessity to route the messages in an efficient way becomes apparent. A service consumer only receives that piece of information that should be handled. Typically, routing can take multiple approaches:

- The "pipe" pattern: a single event triggers a sequence of processing steps, each performing a specific function.
- The "content based router" pattern: the message content is used to take decisions about the receivers
- The "message dispatcher" pattern: a message is sent to a list of services
- The "scatter gather" pattern: a request is sent to a number of service providers but all the responses get aggregated into a single response message

In case of SwarmESB based choreographies, all these patterns and many others can be achieved in explicit declarative and imperative code.

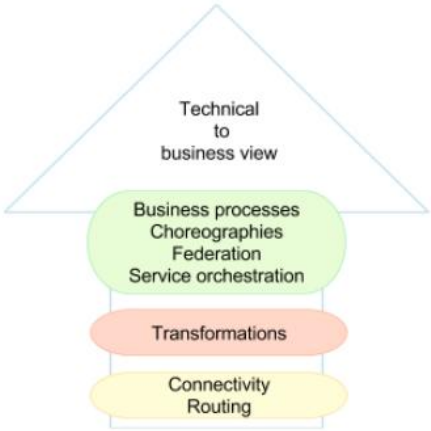


Figure 1. The main roles of an ESB

Transformation: integrated service and applications do not have the same data formats and the ESB is a good place to handle the transformations among these formats. The transformation services that are specialized in the needs of individual applications plugged into the bus can be located anywhere and accessible everywhere on the bus. The transformation can be implemented in the form of adapter nodes or can be implicit in the scripts describing the routing. In this paper, we present the transformation layer implemented in SwarmESB. The proposed methods are able to automate integration with web services, expose web services and perform complex data transformations related to integration or privacy concerns.

Business processes and Service Orchestration concepts are unifying concerns that can be explained meaningfully to the final user (map in scripts or descriptions specific user stories or use cases). They also provide useful abstractions for software analysts, software architects and developers. There are two main types of business processes: long living processes and short living processes. Long living processes are abstracting business concerns that take a long time to be executed (they have a persistent state stored in databases). Until the end of their execution, long living processes are prepared to receive various human inputs or special

events in their execution environment (time events, changes in data structures, creation of new objects, etc.).

Human intervention in business processes is usually described by the "workflow" concept. It is quite tempting to use workflow and business process concept as synonyms. This is justified and it is acceptable because, in execution, any manual intervention from a dedicated operator is almost identical to non-human change. In both cases, we are talking about a set of events and changes in databases or in data structures.

A key point is that workflows follow the opposite paradigm of state-based approach rather than a flow-based one like Business Process Execution Language (BPEL) orchestrators. In some cases, the workflow approach is better adapted to long-lived processes, without being restricted from sitting on top of orchestrated services. Hence, workflow servers are usefully complemented by "straight" orchestrators and we may find solutions that are deploying two business process-oriented servers. Additionally, in many ESBs, short living processes are represented by the routing mechanism and in some others by the BPEL type of orchestration. Unfortunately, this approach exposes the developers to too many different languages or approaches when describing short living processes (integration processes). In OPERANDO, SwarmESB choice avoids the complexity and the redundancy of effort and resources caused by the usage of three quite different process description languages. The usage of orchestration concept is discussed in multiple contexts and sometimes with different meanings. We can talk about orchestrations in the context of provisioning in the virtualized deployment environments (in dynamic data-center use cases) and in Service Oriented Architectures. In both cases, orchestration is about aligning the business request with the applications, data, and infrastructure. It defines the policies and service levels through automated workflows, provisioning, and change management. From the data-center or deployment management perspective, orchestration creates an application-aligned infrastructure that can be scaled up or down based on the needs of the applications. For simplicity, we will call this kind of orchestration, *orchestration for deployments and provisioning.*

A somewhat different usage of the orchestration concept is related to the process of coordinating an exchange of information through

interactions of web services. We will call this kind of orchestration service *orchestration*. Advanced Service Oriented Architectures could try to decouple the orchestration layer from the service layer in the form of the service orchestration or service choreography. Systems like ESB or integration Platform as a Service (iPaaS) are typically deployed and fine-tuned in order to perform this role. For dynamic data-center use cases, the orchestration is typically related and closely connected to monitoring infrastructure and to the management of the virtualisation solutions. As it will be explained below, the choreography concept and especially the executable choreography is a technique that offers an alternative implementation for the service orchestration. The final results of the service orchestration and of the choreography may look identical (some services are mixed together) but from the point of view of performance, scalability, security and privacy, the decentralised way of choreography brings important benefits. An ESB is a strategic component in any complex system as it succeeds in reducing coupling between solution's components. Reduced coupling enables parallel work to be performed by multiple teams that use separate tools, processes and even platforms/technologies (Java, C#, PHP, node.js etc.). An ESB enables an SOA that is an alternative to the client server model. An ESB promotes agility and flexibility regarding communication between applications and subsystems.

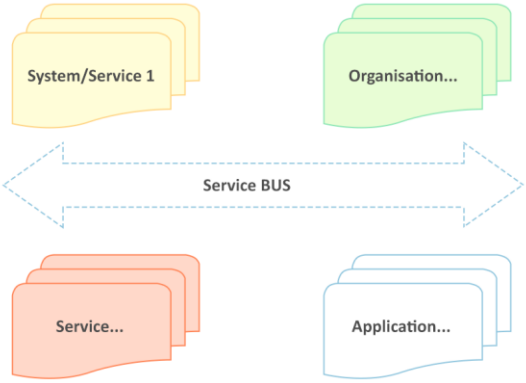


Figure 2. The generic architecture for ESB based systems

The purpose of the integration bus is to provide a flexible method to compose services and components, ensure security and scalability of the system and to allow development towards a federated system between multiple ESBs. Enterprise Service Bus systems must be seen as an architectural pattern. An ESB offers a standard way of integration between applications, services or other kinds of integration objects. An ESB mediates between service providers and service consumers. Integration of loosely coupled services within or across organizations can be obtained.

The SwarmESB current architecture starts from the premises that we are supporting the federation of services among multiple organisations. This perspective implies a technology capable of executing business processes among multiple organisations (choreography). Any usage of centralised message queues or centralised Business Process Management (BPM) engines will not be sufficient because of the security and privacy issues raised by centralisation. SwarmESB uses a script based on the routing method that circumvents these privacy concerns.

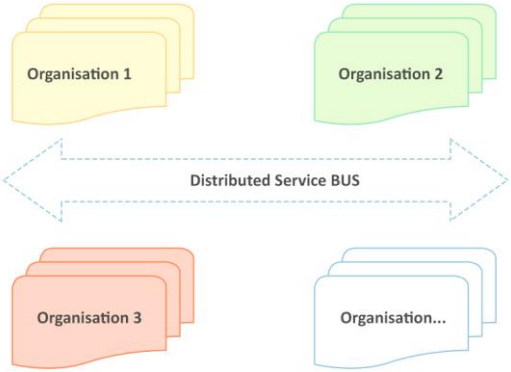


Figure 3. The architecture for choreography based integration offering federation

3 Web Service transformation language proposal

To enable complex communication between the distributed bus provided by SwarmESB and the external world, we have analysed the types of

transformation that we have to create in order to enable inbound and outbound usage of web services. A typical integration case is the need to call existing web services inside executable choreography scripts. Another case is the requirement of a new or existing application to communicate with the ESB using web services. These capabilities were not available by default in SwarmESB and as workaround, we used to create custom code for each case. Beyond these two cases, our research for OPERANDO has shown that other three types of transformations exist. The current implementation can be found in the TransRest open source project [14]. The resulted five types of transformations are presented in Table 1.

Table 1. Types of transformations

Name	Description
Service to Functions transformations (SF)	This transformation can translate a REST service into functions usable in a processing node (e.g. Swarm ESB adapter) and from choreographies. Intuitively, this transformation is just a quick method to generate some functions that asynchronously call remote web services. This simple transformation allows documenting the web service and it also permits a uniform working style inside the SwarmESB based project in which the adapters are plain JavaScript functions.
Choreography to Service transformations (CS)	This transformation exposes a swarm workflow (choreography) as a REST web service. Since the same based systems are real time systems that allow push notification and multiple results for a call, this transformation offers a bridge to the applications that are designed to work in an ask/request method promoted by REST services. The CS transformation allows that existing services to be refactored to use SwarmESB and allows the reuse of the existing skills and tools.

Function to Service transformations (FS)	The FS transformation exposes functions as REST web APIs. This type of transformation is very useful for testing and mocking web services but also for the creation of REST web services with very little code. As we see bellow, the transformation language hides all the wiring usually required to create web services. This transformation will work together with CS and I transformations allowing to expose an enriched set of services.
Service to Choreography transformations (SC)	This transformation can change a REST Service into a workflow/choreography (swarm description/script) based on an existing template. This kind of transformation is complex and requires metaprogramming capabilities from the choreography implementation. This transformation has not been implemented yet in SwarmESB. The SF transformation allows manual creation of new choreography based on existing web services so basically the SC transformations should be manually programmed.
Interceptor transformations (I)	This kind of transformation can be seen as a combination between SC and CS transformations. An Interceptor transformation can be seen as a smart proxy between some arbitrary REST APIs and an exposed REST APIs. The benefit will be that the transformation can intercept every call and can enrich each call with some arbitrary logic that will be hosted in a swarm workflow description.

All five types of transformations may be described in a common language called Swarm to web service Transformation (SwarmTL). For syntax description, we used Backus-Naur Form notation. SwarmTL DSL is an internal DSL (Domain Specific Language) so all JavaScript syntactic and semantic rules should be considered. By using an internal DSL we

can benefit from existing tools for debugging, Integrated Development Environments and programming expertise, therefore we reduce adoption risks for this new technology.

SwarmTL language is presented below:

```
<transformation>    :==   "{" <properties> "," <blockList> "}"
<properties>        :==   "" | <property> | <property> <opt-comma>
                          <properties>
<blockList>         :==   <block> | <block> <opt-comma> <blockList>
<block>             :==   <blockName> <opt-whitespace>":" <opt
                          whitespace>
                          "{" <blockPropertyList> "}"
blockPropertyList   :==   "" | <blockProperty> | <blockProperty> <opt
                          comma>
                          <blockPropertyList>
<blockProperty>     :==   <mandatoryProperty> | <specificProperty>
<property>          :==   <globalKey> <equal> <value>
<mandatoryPropert   :==   <mandatoryKey> <equal> <value>
y>
<specificProperty>  :==   <specificKey> <equal> <value>
<mandatoryKey>      :==   "method" | "params" | "path"
<globalKey>         :==   "baseUrl" | "port" | "swarm"
<specificKey>       :==   "code" | "phase"
<value>             :==   jsString | jsAnonymousFunction | jsArray
<opt-comma>         :==   <opt-whitespace> "," <opt-whitespace> | ""
<equal>             :==   <opt-whitespace> "=" <opt-whitespace>
<opt-whitespace>    :==   " " <opt-whitespace> | ""
```

In order to get an intuitive image about the syntax of the transformations we are exemplifying a SF transformation that takes a remote REST web service from http://localhost:3000 and exposes a set of functions with the name of the blocks (e.g. baseUrl or createEntity).

```
{
  baseUrl:  'http://localhost:3000',
  getEntity: {
     method:'get',
     params: ['entity', 'token'],
     path:'/$entity/$token'
```

```
  },
  createEntity: {
    method: 'put',
    params: ['entityId', 'token', '__body'],
    path : '/?id=$entityId&token=$token'
  }
}
```

Any transformation is composed of global properties and a list of transformation blocks. The global properties are basically *key value* assignments. Each block is composed of a list of properties known as 'block' properties. A set of properties is present in all the transformations (and are called mandatory properties) but the others are optional or transformation specific. The mandatory properties are "method", "params" and "path". The values for "method" are "get", "post", "put", "delete" corresponding to the HTTP verbs. The "path" parameter specifies the part of the URL that is used to route the request to the actual implementation. The path value is a string that consists of fixed strings and "parameters". All parameters are prefixed by an "$" character that enables the url parse to determine the place of the corresponding values in the actual urls. The values for "params" property are a JavaScript array of string denoting the parameters names. The actual usage of the parameter depends on the type of the transformation. These parameters should appear as strings in the url prefixed by a "$". To terminate a parameter placeholder and to begin a new string or a new parameter, the "/" character should be used. As we can see, this scheme is similar to the ones used in the other routing web engines. A similar naming scheme for routing is used to connect node.js framework but instead of "$" they use ":".

Additionally, we support variables that are not part of the URL, specifically the "__body" parameter that will contain the content of the POST and PUT requests. All the names of variables prefixed with "__" are reserved to be used with the parameters of the POST and PUT body content. In the global section, a set of attributes can be used. "baseURL" key means the base url of the rest services. The "node" means the group (or the node type for the processing nodes) on which the transformation

will be executed. Other specific properties are specific to particular transformation types as we can see in Table 2.

Table 2. Swarm Transformation Language property names

Property	**Transformations**	**Semantic description**	**Possible value**
baseUrl	CS,SC,SF,I	Global property that specifies the base url for a remote service,	A remote URL
swarm	I	Global property that specifies the name of a swarm used in I transformations to actually call the remote REST service.	String
template	SC	Global property that specifies the name of a swarm used as template in SC transformations	A swarm name
method	CS,FS,SC,SF,I	A block property that specifies the HTTP method used for routing in local and remote services	GET | POST | PUT | DELETE
path	CS,FS,SC,SF,I	A block property that specifies the path in the url for remote services or for the local router	specially formatted string

params	CS,FS,SC,SF,I	A block property having as value an array with the name of the parameters used in the choreography constructors, of the generated the functions in all transformations	jsArray: JavaScript array with strings
phase	CS	A block property specifying the phase name that is transformed as a service in CS transformations	String
Code	FS	A block property used by FS transformations to specify the actual implementation of the service. The value is just a plain JavaScript function returning a value asynchronously.	*jsAnonymous Function:* anonymous function
Result-Phase	CS	A block property used by CS transformations to specify the phase name of the result.	String

Tests and code demonstrating the transformations can be found in the TransREST open source project [14].

4 Web service transformations applied in OPERANDO

OPERANDO system is built around a Shared Bus that supports federation and advanced transformation capable of integrating internal and third party web services and functionalities.

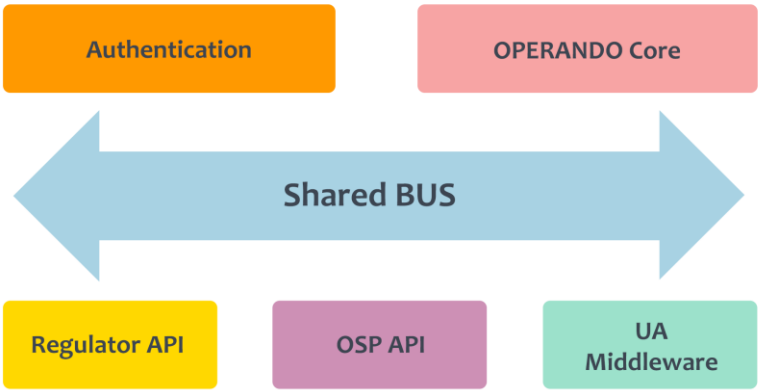


Figure 4. OPERANDO architecture. The high-level view diagram

The major components or layers in the OPERANDO architecture consist of:

- Authentication layer: a set of services and components responsible with the authentication and monitoring of all the business processes involving OPERANDO
- OPERANDO Core services: a collection of complex services, techniques and algorithms that offer functionalities to OSPs such as secure data vaults, anonymization, data mining, etc.
- REGULATOR API: a collection of web services offered to legal authorities (regulators) to monitor and control OPERANDO's features regarding privacy laws and regulations
- Online Service Providers APIs (OSP APIs) refers to a set of extensible APIs that can be integrated and transformed by the

OPERANDO to be made available for use in applications developed by third party developers called OSPs

- UA Middleware (User Agent Middleware) : a collection of services and workflows used by the OPERANDO client side components

For OPERANDO we have found three generic use cases where we may use web service transformations:

a) composition of multiple services from the OPERANDO's internal services (OPERANDO Core in figure 4) For this use case, we use SF transformations to translate external web services to the bus into JavaScript functions. These web services are external from the point of view of the bus but are internal for OPERANDO. These functions are exposed to choreographies and used by processing nodes that are called adapter nodes in SwarmESB [6]. With this type of transformation, we can automatically integrate multiple services developed in various languages and make them accessible to the bus without writing any code. In SwarmTL only the declarative descriptions is required and it reduces risks of bugs of using lower level libraries to do REST remote calls.

b) exposition of a single service from Core that will be directly exposed almost unchanged . In this case, the existing web services are enriched by adding only a layer of authentication or by filtering the data within a logical layer responsible with transparent data transformation consisting in real-time anonymization. For this use case we can use an I transformation that can enrich an existing web services while exposing web services to the external environment.

c) creation of custom made web services that have to fit with the need of particular OSP APIs and UA Middleware.

For this use case, we make combinations of FS, SF and CS transformations. SF transformations are capable of exposing various Web Services (implemented with various technologies and by different partners) to the Shared Bus. FS and CS transformation are capable of exposing web services towards outside parties (OSPs, clients, legal regulators) by translating custom made functions and SwarmESB choreographies in web services.

For OPERANDO project, we have analysed the short and medium term quantitative and qualitative effects of the web service

transformations. By unifying a set of 5 complementary operations between functions, web services and choreographies we have managed to reduce the quantity of conventions that a programmer has to gasp. An obvious quantitative effect is the reduction of the number of code lines required to create a web service or to use existing web services in choreographies. The reduction in the number of code lines correlates with the reduction in the number of bugs as it is commonly accepted [15].

We constantly evolve SwarmESB in area of building better, generic error handling mechanisms. Our perspective is that every step that increases the use of these generic mechanisms instead of relying on custom code - created by the programmers using lower level libraries- is very important for the reduction of the programming costs and can increase the maintainability of the resulted systems.

5 Conclusion

ESBs created around the concept of executable choreographies and other classical ESBs that are using orchestration engines for web services may have similar purposes. However, as it has been demonstrated in the previous research [4] executable choreographies are designed to provide federation concepts and better privacy ensuring capabilities in complex solutions involving multiple organisations. Executable choreographies do not have a direct correspondent in the web service world and in this paper we have presented five types of web service transformations that enable a bridge between REST web services programming environments and the executable choreography environments. Providing real time messaging [5], the swarm communication pattern can be seen as a generalization for request/response case of the http communication. Likewise, web service transformations are a general case for the more well-known concept of data transformation [12]. The service transformations can be used to implement the well-known concept of data transformations but can also be used for other integration purposes that typically do not belong to data transformation. The most widely used description languages for web services do not annotate data for privacy concerns. Therefore, it makes sense to extend the descriptions used for web service transformations in order to add support for automated checks or automated anonymization of the choreographies. We have already allocated research efforts in this

direction. Nevertheless, the ubiquity of web services encouraged our efforts to extend the executable choreographies with deeper support for web services and this has turned out to be an opportunity to create technologies that provides qualitative improvements for programmers' productivity.

Acknowledgments. This work was partially supported by the European Commission under the Horizon 2020 Programme (H2020), as part of the OPERANDO project (Grant Agreement no. 653704).

References

[1] OPERANDO: http://cordis.europa.eu/project/rcn/194891_en.html.

[2] Chappell, D.A.: Enterprise Service Bus: Theory in Practice, Publisher: O'Reilly Media (2009).

[3] SwarmESB open source project: http://github.com/salboaie/SwarmESB.

[4] Sinica Alboaie, Lenuta Alboaie, Andrei Panu. Levels of Privacy for e-Health systems in the cloud era, 24th International Conference on Information Systems Development Harbin, China, August 25-27 (2015).

[5] Lenuta Alboaie, Sinica Alboaie, Panu Andrei. Swarm Communication - a Messaging Pattern proposal for Dynamic Scalability in Cloud, 15th IEEE International Conference on High Performance Computing and Communications, China, pp: 1930 – 1937, 10.1109/HPCC.and.EUC.2013.277 (2013).

[6] Carzaniga, A., Gian Pietro P., Vigna, G.: Designing distributed applications with mobile code paradigms, Proceedings of the 19th international conference on Software engineering, ACM (1997).

[7] Gul Agha. "Actors: A Model of Concurrent Computation in Distributed Systems". Doctoral Dissertation. MIT Press. (1986).

[8] Milner, Robin. *Communicating and Mobile Systems: The π-calculus*. Cambridge, UK: Cambridge University Press. ISBN 0-521-65869-1 (1999).

[9] Erl, T.: SOA Design Patterns, Publisher Prentice Hall, ISBN: 013613516, 9780136135166 (2009).

[10] Florin-C. Pop, Marcel Cremene, Mircea-F. Vaida, Michel Riveill. Natural language service composition with request disambiguation, ICSOC 2010, 7-10 December USA, Lecture Notes in Computer Science, volume: 6470, http://dx.doi.org/10.1007/978-3-642-17358-5, pp. 670-677 (2010).

[11] JSR-000352 Batch Applications for the Java™ Platform (2014).

[12] Cuzzocrea, A.: A framework for modeling and supporting data transformation services over data and knowledge grids with real-time bound constraints. Concurrency Computat.: Pract. Exper., 23: 436–457. doi:10.1002/cpe.1648 (http://onlinelibrary.wiley.com/doi/10.1002/cpe.1648/full) (2011).

[13] Lenuta Alboaie, Sinica Alboaie, Tudor Barbu. Extending swarm communication to unify choreography and long-lived processes, 23rd International Conference on Information Systems Development, Croatia, ISBN 978-953-6071-43-2, pp: 375-382(2014).

[14] [TransRest] implementation: http://github.com/salboaie/transrest.

[15] Steve McConnell. Code Complete, 2nd Edition. Redmond, Wa.: Microsoft Press (2004).

Sînică Alboaie[1,2], Lenuta Alboaie[3], Mircea-Florin Vaida[4]

Sînică Alboaie[1,2]

[1]Technical University of Cluj-Napoca, Gh. Baritiu Street, 26-28, Cluj-Napoca, Romania
[2]RomSoft Srl, Iasi, Romania, Research Department
E-mail: salboaie@gmail.com

Lenuta Alboaie[3]
[3]Faculty of Computer Science of the University "Al. I Cuza" of Iasi, Romania
E-mail: adria@info.uaic.ro

Mircea-Florin Vaida[4]
[4]Technical University of Cluj-Napoca, Communication Department, Gh. Baritiu Street, 26-28, Cluj-Napoca, Romania
E-mail: mircea.vaida@com.utcluj.ro

Proceedings of the Conference on Mathematical Foundations of Informatics
MFOI2016, July 25-29, 2016, Chisinau, Republic of Moldova

Countable Sets in Finitely Supported Mathematics

Andrei Alexandru, Gabriel Ciobanu

Abstract

This paper presents the notion of countable set in a recently developed framework, named Finitely Supported Mathematics. This new framework actually represents a reformulation of Zermelo-Fraenkel mathematics in the world of invariant or finitely supported structures. It was introduced in order to characterize infinite objects by using their finite supports. We prove some algebraic properties of the countable sets, and establish several relationships between countable union theorems and countable choice principles in Finitely Supported Mathematics.

Keywords: countable sets, finitely supported structures, invariant sets, countable choice principles, countable union theorems.

1 Introduction

Finitely Supported Mathematics (FSM) generalizes classical Zermelo-Fraenkel (ZF) mathematics, and represents an appropriate framework to work with (infinite) structures in terms of finitely supported objects (see [2, 3]). FSM is a mathematics which is consistent with the axioms of Fraenkel-Mostowski (FM) set theory. FM set theory has its origins in an approach developed initially by Fraenkel and Mostowski in 1930s, in order to prove the independence of the axiom of choice and other axioms in classical ZF set theory [7, 12]. In 2000s, the FM permutation model of Zermelo-Fraenkel set theory with atoms (ZFA) was axiomatized and presented as an independent set theory, named FM set

theory [8]. Rather than using a non-standard set theory, one could alternatively work with nominal sets [13], which are defined within ZF as usual sets endowed with some group actions satisfying a finite support requirement. There also exists an alternative definition for nominal sets in the FM framework. They can be defined as sets constructed according to the FM axioms with the additional property of being empty supported (invariant under all finitary permutations of atoms). These two ways of defining nominal sets finally lead to similar properties [2]. We use the terminology "invariant" for "nominal" in order to establish a connection between approaches in the FM framework and in the ZF framework.

Actually, FSM represents ZF set theory reformulated in terms of finitely supported objects. The theory of nominal sets, rephrased for possible non-countable sets of atoms, could be considered as a tool for defining FSM. The principles of constructing FSM have historical roots both in the definition of 'logical notions' in Tarski's view [14] and in the Erlangen Program of Felix Klein for the classification of various geometries according to invariants under suitable groups of transformations [10]. There also exist some connections between FSM, the Gandy machines from [9] and the admissible sets from [5], which are presented in [2]. The general principle of defining FSM states that all the structures have to be invariant or finitely supported. As a consequence, we cannot obtain a property in FSM only by involving a ZF result without an appropriate proof reformulated according to the related finite support requirement. Moreover, as we proved in [2], not every ZF result can be directly reformulated in FSM, in terms of finitely supported objects. This is because given an invariant set, some of its subsets may be non-finitely supported. A related example is represented by a simultaneously infinite and coinfinite subset of the invariant set of all atoms. Therefore, the translation of a ZF result into the framework of invariant sets deserves a special attention.

The consistency of choice principles in various models of ZFA (including the permutation models) was studied in depth in the last century. Since FSM generalizes/extends the related permutation models,

it became an important problem to study the consistency of choice principles in FSM. In [4] we proved that the choice principles generally denoted by **AC**, **DC**, **ZL**, **CC**, **PCC**, **AC(fin)**, **Fin**, **PIT**, **UFT**, **OP**, **KW**, and **OEP**, reformulated in terms of finitely supported objects, are all inconsistent in FSM. Since FSM is consistent even if the set of atoms is not countable, such results do not overlap on some related properties in the basic or in the second Fraenkel modes of ZFA set theory [11] nor on some related properties in the theory of nominal sets (which are defined by involving countable sets of atoms) [13].

In this paper our goal is to continue the development in [4], and to introduce and study the notion of countable set internally in FSM, in order to establish a connection between countable union theorems and countable choice principles in FSM.

2 Invariant Sets

Let A be a fixed infinite ZF-set. The following results also make sense if A is considered to be the set of atoms in the ZFA framework and if 'ZF' is replaced by 'ZFA' in their statement.

Definition 2.1. *i)* A transposition *is a function* $(a\,b) : A \to A$ *defined by* $(a\,b)(a) = b$, $(a\,b)(b) = a$ *and* $(a\,b)(n) = n$ *for* $n \neq a, b$.

ii) A permutation *of* A *is a one-to-one and onto function on* A *which interchanges only finitely many elements.*

Let S_A be the group of all permutations; in our approach S_A is not the entire set of bijections of A, but the set of those bijections of A which can be expressed by composing finitely many transpositions.

Definition 2.2.

- *Let* X *be a ZF-set. An* S_A-action *on* X *is a function* $\cdot : S_A \times X \to X$ *having the properties that* $Id \cdot x = x$ *and* $\pi \cdot (\pi^{'} \cdot x) = (\pi \circ \pi^{'}) \cdot x$ *for all* $\pi, \pi' \in S_A$ *and* $x \in X$.
- *An* S_A-set *is a pair* $(X, \cdot)$ *where* X *is a ZF-set, and* $\cdot : S_A \times X \to X$ *is an* S_A*-action on* X. *We simply use* X *whenever no confusion arises.*

- *Let $(X, \cdot)$ be an S_A-set. We say that $S \subset A$* supports *x whenever for each $\pi \in Fix(S)$ we have $\pi \cdot x = x$, where $Fix(S) = \{\pi \mid \pi(a) = a, \forall a \in S\}$.*

- *Let $(X, \cdot)$ be an S_A-set. We say that X is an* invariant set *if for each $x \in X$ there exists a finite set $S_x \subset A$ which supports x (i.e. x is finitely supported). Invariant sets are also called* nominal sets *if we work in the ZF framework [13], or* equivariant sets *if they are defined as elements in the cumulative hierarchy $FM(A)$ [8].*

Theorem 2.3 ([13])**.** *Let X be an S_A-set, and for each $x \in X$ let us consider $\mathcal{F}_x = \{S \subset A \mid S \text{ finite}, S \text{ supports } x\}$. If $\mathcal{F}_x$ is non-empty (particularly if X is an invariant set), then it has a least element which also supports x. We call this element* the support of x*, and we denote it by $supp(x)$.*

Proposition 2.4 ([13])**.** *Let $(X, \cdot)$ be an S_A-set and let $\pi \in S_A$ be an arbitrary permutation. Then for each $x \in X$ which is finitely supported we have that $\pi \cdot x$ is finitely supported and $supp(\pi \cdot x) = \pi(supp(x))$.*

Example 2.5.

1. *The set A of atoms is an S_A-set with the S_A-action $\cdot : S_A \times A \to A$ defined by $\pi \cdot a := \pi(a)$ for all $\pi \in S_A$ and $a \in A$. $(A, \cdot)$ is an invariant set because for each $a \in A$ it follows that $supp(a) = \{a\}$.*

2. *The set A of atoms is an S_A-set with the S_A-action $\cdot : S_A \times A \to A$ defined by $\pi \cdot a := a$ for all $\pi \in S_A$ and $a \in A$. $(A, \cdot)$ is an invariant set because for each $a \in A$ it follows that $supp(a) = \emptyset$.*

3. *The set S_A is an S_A-set with the S_A-action $\cdot : S_A \times S_A \to S_A$ defined by $\pi \cdot \sigma := \pi \circ \sigma \circ \pi^{-1}$ for all $\pi, \sigma \in S_A$. $(S_A, \cdot)$ is an invariant set because for each $\sigma \in S_A$ it follows that $supp(\sigma) = \{a \in A \mid \sigma(a) \neq a\}$.*

4. *Any ordinary ZF-set X (such as $\mathbb{N}, \mathbb{Z}, \mathbb{Q}$ or $\mathbb{R}$ for example) is an S_A-set with the trivial S_A-action $\cdot : S_A \times X \to X$ defined by*

$\pi \cdot x := x$ *for all* $\pi \in S_A$ *and* $x \in X$. *Also* X *is an invariant set because for each* $x \in X$ *it follows that* $supp(x) = \emptyset$.

5. *If* $(X, \cdot)$ *is an* S_A*-set, then* $\wp(X) = \{Y \mid Y \subseteq X\}$ *is also an* S_A*-set with the* S_A*-action* $\star : S_A \times \wp(X) \to \wp(X)$ *defined by* $\pi \star Y := \{\pi \cdot y \mid y \in Y\}$ *for all* $\pi \in S_A$, *and all* $Y \subseteq X$. *For each invariant set* $(X, \cdot)$, *we denote by* $\wp_{fs}(X)$ *the set formed from those subsets of* X *which are finitely supported according to the action* $\star$. *According to Proposition 2.4,* $(\wp_{fs}(X), \star|_{\wp_{fs}(X)})$ *is an invariant set, where* $\star|_{\wp_{fs}(X)} : S_A \times \wp_{fs}(X) \to \wp_{fs}(X)$ *is defined by* $\pi \star |_{\wp_{fs}(X)} Y := \pi \star Y$ *for all* $\pi \in S_A$ *and* $Y \in \wp_{fs}(X)$.

6. *Let* $(X, \cdot)$ *and* $(Y, \diamond)$ *be* S_A*-sets. The Cartesian product* $X \times Y$ *is also an* S_A*-set with the* S_A*-action* $\star : S_A \times (X \times Y) \to (X \times Y)$ *defined by* $\pi \star (x, y) = (\pi \cdot x, \pi \diamond y)$ *for all* $\pi \in S_A$ *and all* $x \in X$, $y \in Y$. *If* $(X, \cdot)$ *and* $(Y, \diamond)$ *are invariant sets, then* $(X \times Y, \star)$ *is also an invariant set.*

Definition 2.6. *Let* $(X, \cdot)$ *be an invariant set. A subset* Z *of* X *is called* finitely supported *if* $Z \in \wp_{fs}(X)$.

Note that an equivariant subset of an invariant set is itself an invariant set.

Recall that a function $f : X \to Y$ is a particular relation. Precisely, a function $f : X \to Y$ is a subset f of $X \times Y$ characterized by the property that for each $x \in X$ there is exactly one $y \in Y$ such that $(x, y) \in f$.

Definition 2.7. 1. *Let* X *and* Y *be invariant sets. A function* $f : X \to Y$ *is* finitely supported *if* $f \in \wp_{fs}(X \times Y)$.

2. *Let* X *be a finitely supported subset of an invariant set* X_1, *and* Y *a finitely supported subset of an invariant set* Y_1. *A function* $f : X \to Y$ *is* finitely supported *if* $f \in \wp_{fs}(X_1 \times Y_1)$.

Let $Y^X = \{f \subseteq X \times Y \mid f$ is a function from the underlying set of X to the underlying set of $Y\}$.

Proposition 2.8 ([13])**.** *Let $(X, \cdot)$ and $(Y, \diamond)$ be invariant sets. Then Y^X is an S_A-set with the S_A-action $\star : S_A \times Y^X \to Y^X$ defined by $(\pi \star f)(x) = \pi \diamond (f(\pi^{-1} \cdot x))$ for all $\pi \in S_A$, $f \in Y^X$ and $x \in X$. A function $f : X \to Y$ is finitely supported in the sense of Definition 2.7 if and only if it is finitely supported with respect to $\star$.*

Proposition 2.9 ([13])**.** *Let $(X, \cdot)$ and $(Y, \diamond)$ be invariant sets. Let $f \in Y^X$ and $\sigma \in S_A$ be arbitrary elements. Let $\star : S_A \times Y^X \to Y^X$ be the S_A-action on Y^X, defined by: $(\pi \star f)(x) = \pi \diamond (f(\pi^{-1} \cdot x))$ for all $\pi \in S_A$, $f \in Y^X$ and $x \in X$. Then $\sigma \star f = f$ if and only if for all $x \in X$ we have $f(\sigma \cdot x) = \sigma \diamond f(x)$.*

Proposition 2.10. *Let $(X, \cdot)$ and $(Y, \diamond)$ be invariant sets, and let Z be a finitely supported subset of X. Let $f : Z \to Y$ be a function. The function f is finitely supported if and only if there exists a finite set S of atoms such that for all $x \in Z$ and all $\pi \in Fix(S)$ we have $\pi \cdot x \in Z$ and $f(\pi \cdot x) = \pi \diamond f(x)$.*

3 Countable Sets in FSM

Definition 3.1.

1. *Let Y be a finitely supported subset of an invariant set X. Then Y is* of cardinality at most k in FSM *if there exists a finitely supported onto application $f : Z \to Y$, where Z is an ordinary ZF set (i.e. a trivial invariant set) of cardinality k.*

2. *Let Y be a finitely supported subset of an invariant set X. Then Y is* countable in FSM (or FSM countable) *if there exists a finitely supported onto application $f : \mathbb{N} \to Y$. The countable sets in FSM are precisely those sets of cardinality at most $\aleph_0$ in FSM.*

Proposition 3.2. *Let Y, Z be finitely supported subsets of an invariant set X.*

1. *Y is countable in FSM if and only if there exists a finitely supported one-to-one application $g : Y \to \mathbb{N}$.*

2. *If Y and Z are countable in FSM, then so is $Y \times Z$.*

3. *If Y is countable in FSM, then any subset of Y which is finitely supported as a subset of X is also countable in FSM.*

Proof. 1. Suppose that Y is countable in FSM. Then there exists a finitely supported onto application $f : \mathbb{N} \to Y$. We define $g : Y \to \mathbb{N}$ by $g(y) = min[f^{-1}(\{y\})]$, for all $y \in Y$. According to Proposition 2.10, g is supported by $supp(f) \cup supp(Y)$. Obviously, g is one-to-one. Conversely, if there exists a finitely supported one-to-one application $g : Y \to \mathbb{N}$, then $g(Y)$ is supported by $supp(g) \cup supp(Y)$ (it is also equivariant as a subset of the trivial nominal set $\mathbb{N}$). Thus, there exists a finitely supported bijection $g : Y \to g(Y)$, where $g(Y) \subseteq \mathbb{N}$. We define $f : \mathbb{N} \to Y$ by

$$f(n) = \begin{cases} g^{-1}(n) & \text{if } n \in g(Y) \\ \\ t & \text{if } n \in \mathbb{N} \setminus g(Y) \end{cases},$$

where t is a fixed element of Y. According to Proposition 2.10, we have that f is supported by $supp(g) \cup supp(Y) \cup supp(t)$. Moreover, f is onto.

2. First we remark that $\mathbb{N} \times \mathbb{N}$ is countable in FSM because there exists the equivariant one-to-one application $f : \mathbb{N} \times \mathbb{N} \to \mathbb{N}$ defined by $f(m, n) = 2^m 3^n$. Since Y and Z are countable in FSM, there are finitely supported onto applications $f_1 : \mathbb{N} \to Y$ and $f_2 : \mathbb{N} \to Z$. Thus, $f_1 \times f_2 : \mathbb{N} \times \mathbb{N} \to Y \times Z$ defined by $(f_1 \times f_2)(m, n) = (f_1(m), f_2(n))$, $\forall m, n \in \mathbb{N}$ is onto. Moreover, $f_1 \times f_2$ is supported by $supp(f_1) \cup supp(f_2)$. Thus, $(f_1 \times f_2) \circ f^{-1} : \mathbb{N} \to Y \times Z$ is a finitely supported onto application, and so $Y \times Z$ is countable in FSM.

3. Let K be a subset of Y, such that K is finitely supported as a subset of X. Since Y is countable in FSM, according to Lemma 3.2(1), there exists a finitely supported one-to-one application $g : Y \to \mathbb{N}$. The restriction of g to K, $g|_K : K \to \mathbb{N}$ defined by $g|_K(k) = g(k)$ for all $k \in K$, is still injective, and it is supported by $supp(K) \cup supp(g)$. Therefore, by applying again Lemma 3.2(1), we obtain that K is countable in FSM. □

The Countable Union Theorem in ZF, **CUT**, states that a countable union of countable sets is itself countable. It is well known that **CUT** is valid if we assume the validity in ZF of the axiom of countable choice **CC** claiming that given any countable family (sequence) of non-empty sets $\mathcal{F}$, it is possible to select a single element from each member of $\mathcal{F}$ (i.e. there exists a choice function on $\mathcal{F}$). However, the reverse implication, namely **CUT** implies **CC**, is not necessarily valid. As we prove in Proposition 3.3, **CC** reformulated in terms of finitely supported objects, is inconsistent in FSM.

Proposition 3.3 ([4])**.** *The strongest form of countable choice principle, **CC**, claiming that "Given any invariant set X, and any countable family $\mathcal{F} = (X_n)_n$ of subsets of X such that the mapping $n \mapsto X_n$ is finitely supported, there exists a finitely supported choice function on $\mathcal{F}$" is inconsistent in FSM.*

Proof. Let us assume that **CC** is valid in FSM. We consider the countable family $(X_n)_n$, where X_n is the set of all injective n-tuples from A. Since A is infinite, it follows that each X_n is non-empty. In FSM, each X_n is equivariant because A is an invariant set and each permutation is a bijective function. Thus, by applying a permutation π to an n-tuple of atoms, we get another n-tuple of atoms. Therefore, the family $(X_n)_n$ is equivariant and the mapping $n \mapsto X_n$ is also equivariant.

If we assume that **CC** is valid, then according to the formulation of **CC** in FSM, there exists a finitely supported choice function f on $(X_n)_n$. Let $f(X_n) = x_n$ with each $x_n \in X_n$. Let $\pi \in Fix(supp(f))$. According to Proposition 2.10, and because each element X_n is equivariant according to its definition, we obtain that $\pi \cdot x_n = \pi \cdot f(X_n) = f(\pi \star X_n) = f(X_n) = x_n$, where by $\star$ we denoted the S_A-action on $(X_n)_n$ and by $\cdot$ we denoted the S_A-action on $\underset{n}{\cup} X_n$. Therefore, each element x_n is supported by $supp(f)$. However, since each x_n is a finite tuple of atoms, we have $supp(x_n) = x_n$, $\forall n \in \mathbb{N}$. Since $supp(x_n) \subseteq supp(f)$, $\forall n \in \mathbb{N}$, we obtain $x_n \subseteq supp(f)$, $\forall n \in \mathbb{N}$. Since each x_n has exactly n elements, this contradicts the finiteness of $supp(f)$. □

However, since no information regarding the countability of A are

available, the consistency of various weaker countable choice principles, such as **CC(k)**, **CC(fin)**, **CC(2)** presented in Definition 3.4, in FSM, remains an open problem. There are no results proving that the related countable choice principles are also inconsistent in FSM. We will prove several relationship results between Countable Union Theorems in FSM and countable choice principles in FSM.

Definition 3.4. *1. The Countable Union Theorem in FSM,* ***CUT****, has the form "Given any invariant set X and any countable family $\mathcal{F} = (X_n)_n$ of countable subsets of X in FSM such that the mapping $n \mapsto X_n$ is finitely supported, then there exists a finitely supported onto application $f : \mathbb{N} \to \bigcup\limits_n X_n$"*

2. The Countable Union Theorem for finite sets in FSM, ***CUT(fin)****, has the form "Given any invariant set X and any countable family $\mathcal{F} = (X_n)_n$ of finite subsets of X such that the mapping $n \mapsto X_n$ is finitely supported, then there exists a finitely supported onto application $f : \mathbb{N} \to \bigcup\limits_n X_n$"*

3. The Countable Union Theorem for 2-element sets in FSM, ***CUT(2)****, has the form "Given any invariant set X and any countable family $\mathcal{F} = (X_n)_n$ of 2-element subsets of X such that the mapping $n \mapsto X_n$ is finitely supported, then there exists a finitely supported onto application $f : \mathbb{N} \to \bigcup\limits_n X_n$"*

4. The Countable Choice Principle for sets of cardinality at most k in FSM, ***CC(k)*** *has the form "Given any invariant set X, and any countable family $\mathcal{F} = (X_n)_n$ of subsets of X of cardinality at most k in FSM such that the mapping $n \mapsto X_n$ is finitely supported, there exists a finitely supported choice function on $\mathcal{F}$."*

5. The Countable Choice Principle for finite sets in FSM, ***CC(fin)*** *has the form "Given any invariant set X, and any countable family $\mathcal{F} = (X_n)_n$ of finite subsets of X such that the mapping $n \mapsto X_n$ is finitely supported, there exists a finitely supported choice function on $\mathcal{F}$."*

6. *The Countable Choice Principle for 2-element sets in FSM,* ***CC(2)*** *has the form "Given any invariant set X, and any countable family $\mathcal{F} = (X_n)_n$ of 2-element subsets of X such that the mapping $n \mapsto X_n$ is finitely supported, there exists a finitely supported choice function on $\mathcal{F}$."*

Theorem 3.5. *In FSM, the following implications hold.*

1. ***CUT*** $\Rightarrow$ ***CC($\aleph_0$)***;
2. ***CUT(fin)*** $\Rightarrow$ ***CC(fin)***;
3. ***CC($\aleph_0^{\aleph_0}$)*** $\Rightarrow$ ***CUT***;
4. ***CC($\aleph_0$)*** $\Rightarrow$ ***CUT(fin)***;
5. ***CUT(2)*** $\Leftrightarrow$ ***CC(2)***;

Proof. 1. Let us assume that **CUT** is valid in FSM. We consider the countable family $\mathcal{F} = (X_n)_n$ in FSM, where each X_n is a non-empty countable subset X in FSM.

From **CUT**, there exists a finitely supported onto application $f : \mathbb{N} \to \underset{n}{\cup} X_n$. Since f is onto and each X_n is non-empty, we have that $f^{-1}(X_n)$ is a non-empty subset of $\mathbb{N}$ for each $n \in \mathbb{N}$. Consider the function $g : \mathcal{F} \to \cup\mathcal{F}$, defined by $g(X_n) = f(min[f^{-1}(X_n)])$. We claim that $supp(f) \cup supp(n \mapsto X_n)$ supports g. Let $\pi \in Fix(supp(f) \cup supp(n \mapsto X_n))$. According to Proposition 2.10, and because $\mathbb{N}$ is a trivial nominal set and each element X_n is supported by $supp(n \mapsto X_n)$, we have $\pi \cdot g(X_n) = \pi \cdot f(min[f^{-1}(X_n)]) = f(\pi \cdot min[f^{-1}(X_n)]) = f(min[f^{-1}(X_n)]) = g(X_n) = g(\pi \star X_n)$, where by $\star$ we denoted the S_A-action on $\mathcal{F}$ and by $\cdot$ we denoted the S_A-action on $\cup\mathcal{F}$. Therefore, g is finitely supported. Moreover, $g(X_n) \in X_n$, and so g is a choice function on $\mathcal{F}$.

2. The proof is similar with the one presented on item 1, with the mention that we consider the countable family $\mathcal{F}$ formed by non-empty finite subsets of X.

3. Let $\mathcal{F} = (X_n)_n$ be a countable family of countable subsets of X in FSM such that the mapping $n \mapsto X_n$ is finitely supported. Since each X_n is countable in FSM, according to Lemma 3.2(1), for any X_n there exists a finitely supported one-to-one application from X_n to $\mathbb{N}$. Let I_n be the set of all finitely supported one-to-one applications from X_n to $\mathbb{N}$. We claim that every I_n is supported by $supp(n \mapsto X_n)$. Let us denote $S = supp(n \mapsto X_n)$. According to Proposition 2.9, it follows that S supports X_n for all $n \in \mathbb{N}$. Let $\pi \in Fix(S)$. We have $\pi \star X_n = X_n$, $\forall n \in \mathbb{N}$. Let us consider an arbitrary $n \in \mathbb{N}$ and an arbitrary element $f \in I_n$. We have that $f : X_n \to \mathbb{N}$ is a finitely supported one-to-one application. According to Proposition 2.4 we have that $\pi\widetilde{\star}f$ is also finitely supported, where by $\widetilde{\star}$ we denoted the standard S_A-action on $\mathbb{N}^X$. Moreover, $\pi\widetilde{\star}f$ is a function whose domain is $\pi \star domain(f) = \pi \star X_n = X_n$. In order to prove the injectivity of f, suppose that $(\pi\widetilde{\star}f)(x) = (\pi\widetilde{\star}f)(y)$. According to Proposition 2.8, we have $f(\pi^{-1} \cdot x) = f(\pi^{-1} \cdot y)$, where $\cdot$ signifies the S_A-action on X. Since f is one-to-one, we have $\pi^{-1} \cdot x = \pi^{-1} \cdot y$, and, because $\cdot$ is a group action, (by composing π on both sides of the previous relation) we obtain $x = y$. We conclude that S supports I_n. Since n was arbitrary chosen, we obtain that S supports the mapping $n \mapsto I_n$. Thus, $(I_n)_n$ is a countable family in FSM.

Now we claim that each I_n is a set of cardinality at most $\aleph_0^{\aleph_0}$ in FSM. Fix some $n \in \mathbb{N}$. Since X_n is countable (i.e. of cardinality of at most $\aleph_0$), there are at most $\aleph_0^{\aleph_0}$ many ZF injections from X_n to $\mathbb{N}$. Thus, there is a ZF onto application $\varphi : Z \to I_n$, where Z is an ordinary ZF set of cardinality at most $\aleph_0^{\aleph_0}$. Our goal is to prove that φ is finitely supported. We know that I_n is not empty, i.e. there exists a finitely supported one-to-one application $f : X_n \to \mathbb{N}$. Since f is finitely supported and $\mathbb{N}$ is a trivial invariant set, according to Proposition 2.9, for any $\pi \in Fix(supp(f))$ we have $f(\pi \cdot x) = f(x)$, $\forall x \in X_n$. Since f is one-to-one it follows that for any $\pi \in Fix(supp(f))$ we have $\pi \cdot x = x$, $\forall x \in X_n$. This means that any element of X_n is supported by the same set $supp(f)$. Therefore, any element from I_n (i.e. any injective function from X_n to $\mathbb{N}$) is finitely supported by $supp(f)$. Thus, I_n

is uniformly supported by $supp(f)$. According to Proposition 2.9, it follows that any function from the trivial invariant set Z to I_n is also finitely supported by $supp(f)$. It means that φ is finitely supported, and so each I_n is a set of cardinality at most $\aleph_0^{\aleph_0}$ in FSM.

We obtained that $(I_n)_n$ is a countable family of subsets of $\mathbb{N}^X$ (any function from X_n to $\mathbb{N}$ can be expressed as a function from X to $\mathbb{N}$ by adding one element to its codomain to represent the image of all $x \in X \setminus X_n$) of cardinality at most $\aleph_0^{\aleph_0}$ in FSM. By $\mathbf{CC}(\aleph_0^{\aleph_0})$ we have that there exists a finitely supported choice function g on $(I_n)_n$. Let $i_n = g(I_n) \in I_n$. Recall that all I_n are supported by $supp(n \mapsto X_n)$. Let $\pi \in Fix(supp(g) \cup supp(n \mapsto X_n))$. For an arbitrary n, we have $\pi \cdot_I i_n = \pi \cdot_I g(I_n) = g(\pi \star_I I_n) = g(I_n) = i_n$, where by $\star_I$ we denoted the S_A-action on $(I_n)_n$ and by $\cdot_I$ we denoted the S_A-action on $\underset{n}{\cup} I_n$. Thus, $supp(g) \cup supp(n \mapsto X_n)$ supports i_n for all $n \in \mathbb{N}$.

Let $\psi : \underset{n}{\cup} X_n \to \mathbb{N} \times \mathbb{N}$ be defined by $\psi(x) = (n_0, i_{n_0}(x))$, where n_0 is the smallest natural number such that $x \in X_{n_0}$, i.e $n_0 = min\{n \,|\, x \in X_n\}$. We claim that $supp(g) \cup supp(n \mapsto X_n)$ supports ψ. Indeed, let $\pi \in Fix(supp(g) \cup supp(n \mapsto X_n))$, and consider an $x \in \underset{n}{\cup} X_n$. We have $\psi(\pi \cdot x) = (m_0, i_{m_0}(\pi \cdot x))$, where m_0 is the smallest natural number such that $\pi \cdot x \in X_m$, i.e $m_0 = min\{m \,|\, \pi \cdot x \in X_m\}$. However, $\pi \cdot x \in X_m$ if and only if $x \in \pi^{-1} \star X_m$. Moreover, since $\pi \in Fix(supp(n \mapsto X_n))$, we have $\pi^{-1} \in Fix(supp(n \mapsto X_n))$, and so $\pi^{-1} \star X_n = X_n$, $\forall n \in \mathbb{N}$. Thus, $\{m \,|\, \pi \cdot x \in X_m\} = \{m \,|\, x \in \pi^{-1} \star X_m\} = \{m \,|\, x \in X_m\}$. So, m_0 coincides to n_0. Thus, $\psi(\pi \cdot x) = (n_0, i_{n_0}(\pi \cdot x))$. Since $supp(g) \cup supp(n \mapsto X_n)$ supports i_{n_0}, it follows that $i_{n_0}(\pi \cdot x) = i_{n_0}(x)$, and so $\psi(\pi \cdot x) = \psi(x)$. Since x was arbitrary chosen form $\underset{n}{\cup} X_n$, we have that ψ is finitely supported. Since each i_n is one-to-one, it follows that ψ is one-to-one. Furthermore, from Lemma 3.2(2), we know that there exists an equivariant bijection $\alpha : \mathbb{N} \times \mathbb{N} \to \mathbb{N}$. Thus, $\alpha \circ \psi$ is a finitely supported injection from $\underset{n}{\cup} X_n$ to $\mathbb{N}$. According to Lemma 3.2(1), we have that $\underset{n}{\cup} X_n$ is countable in FSM.

4. The proof is similar with the one presented at item 3. Let $\mathcal{F} = (X_n)_n$ be a countable family of finite subsets of X such that

the mapping $n \mapsto X_n$ is finitely supported. Since X_n is finite, there exists a finitely supported one-to-one application from X_n to $\mathbb{N}$. Let I_n be the set of all finitely supported one-to-one applications from X_n to $\mathbb{N}$. Similarly as at item 3 we have that $(I_n)_n$ is a countable family in FSM. Moreover, each I_n is a countable set in FSM. Indeed, fix an $n \in \mathbb{N}$. Since X_n is finite, there are at most countably many ZF injections from X_n to $\mathbb{N}$, i.e. there is a ZF injection $\varphi : I_n \to \mathbb{N}$. However, if $X_n = \{y_1, y_2, \ldots, y_m\}$, from Proposition 2.9 we have that $S' = supp(y_1) \cup supp(y_2) \cup \ldots \cup supp(y_m)$ supports any function $f : X_n \to \mathbb{N}$. Thus, S' supports any element from I_n, and so it supports any function $\phi : I_n \to \mathbb{N}$, and particularly it supports φ. It means that each I_n is a countable set in FSM.

We obtained that $(I_n)_n$ is a countable family of countable sets in FSM, and by $\mathbf{CC(\aleph_0)}$ we have that there exists a finitely supported choice function g on $(I_n)_n$. Let $i_n = g(I_n) \in I_n$. As at the previous point, we have that $supp(g) \cup supp(n \mapsto X_n)$ supports i_n for all $n \in \mathbb{N}$.

Let $\psi : \underset{n}{\cup} X_n \to \mathbb{N} \times \mathbb{N}$ be defined by $\psi(x) = (n_0, i_{n_0}(x))$, where n_0 is the smallest natural number such that $x \in X_{n_0}$. As at the previous item, it follows that ψ is a one-to-one finitely supported function. According to Lemma 3.2(1), we have that $\underset{n}{\cup} X_n$ is countable in FSM.

5. The proof of $\mathbf{CUT(2)} \Rightarrow \mathbf{CC(2)}$ is similar with the one presented on item 1, with the mention that we consider the countable family $\mathcal{F}$ formed by non-empty 2-element subsets of X.

For proving $\mathbf{CC(2)} \Rightarrow \mathbf{CUT(2)}$, let $\mathcal{F} = (X_n)_n$ be a countable family of 2-element subsets of X such that the mapping $n \mapsto X_n$ is finitely supported. According to $\mathbf{CC(2)}$ we have that there exists a finitely supported choice function g on $(X_n)_n$. Let $x_n = g(X_n) \in X_n$. We know that all X_n are supported by the same finite set $supp(n \mapsto X_n)$. Let $\pi \in Fix(supp(g) \cup supp(n \mapsto X_n))$. For an arbitrary n, we have $\pi \cdot x_n = \pi \cdot g(X_n) = g(\pi \star X_n) = g(X_n) = x_n$, where by $\star$ we denoted the S_A-action on $(X_n)_n$ and by $\cdot$ we denoted the S_A-action on $\underset{n}{\cup} X_n$. Thus, $supp(g) \cup supp(n \mapsto X_n)$ supports x_n for all $n \in \mathbb{N}$.

For each n, let y_n be the unique element of $X_n \backslash \{x_n\}$. Since for any n

both x_n and X_n are supported by the same set $supp(g) \cup supp(n \mapsto X_n)$, it follows that y_n is also supported by $supp(g) \cup supp(n \mapsto X_n)$ for all $n \in \mathbb{N}$.

Define $f : \mathbb{N} \to \underset{n}{\cup} X_n$ by $f(n) = \begin{cases} x_{\frac{n}{2}} & \text{if } n \text{ is even} \\ y_{\frac{n-1}{2}} & \text{if } n \text{ is odd} \end{cases}$. We can equivalently describe f as being defined by $f(2k) = x_k$ and $f(2k+1) = y_k$. Clearly, f is onto. Furthermore, because all x_n and all y_n are uniformly supported by $supp(g) \cup supp(n \mapsto X_n)$, we have that $f(n) = \pi \cdot f(n)$, for all $\pi \in Fix(supp(g) \cup supp(n \mapsto X_n))$ and all $n \in \mathbb{N}$. Thus, according to Proposition 2.9, we obtain that f is also supported by $supp(g) \cup supp(n \mapsto X_n)$. Therefore, $\underset{n}{\cup} X_n$ is FSM countable. □

In [13] the set A of atoms used to describe the theory of nominal sets is assumed to be countable. However, this assumption is not so well explained because it is not consistent in the world of nominal sets where all the structures should be finitely supported. More precisely, even if we admitted an injection $f : A \to \mathbb{N}$ (as it is assumed in [13]) in the ZF framework, the related injection could not be itself finitely supported. Thus, the set of atoms cannot be countable internally in the framework of nominal sets. In order to be more clear, we present the following proposition.

Proposition 3.6. *There does not exist a finitely supported injection $f : A \to \mathbb{N}$, and so the set of atoms cannot be countable in FSM.*

Proof. Suppose that there exists a finitely supported injection $f : A \to \mathbb{N}$. Let us consider two atoms $a, b \notin supp(f)$ with $a \neq b$. Thus, $a, b \in Fix(supp(f))$, and so $(a\,b) \star f = f$. Let us denote $(a\,b)$ by π. Since the S_A-action $\cdot$ on A is defined as in Example 2.5(1), according to Proposition 2.8, we have $f(a) = (\pi \star f)(a) = f(\pi(a)) = f(b)$ which contradicts the injectivity of f. □

In order to preserve the validity of counting results from the theory of nominal sets, we define ZF countable sets in FSM without requiring

the related sets to be themselves FSM countable. More exactly, we define a set ZF countable set as an FSM set which has the cardinality at most $\aleph_0$ in ZF. This is the only way we could express A as a countable set. However, we remind that in FSM we do not require A to be countable (as Pitts did in the construction of nominal sets [13]).

Definition 3.7. *Let Y be a finitely supported subset of an invariant set X. Then Y is* ZF countable *if there exists an onto application $f : \mathbb{N} \to Y$.*

Definition 3.8. *1. The FSM Strong Countable Union Theorem, **SCUT**, has the form "Given any invariant set X and any FSM countable family $\mathcal{F} = (X_n)_n$ of finitely supported subsets of X which are ZF countable, then $\underset{n}{\cup} X_n$ in FSM countable, i.e. there exists a finitely supported onto application $f : \mathbb{N} \to \underset{n}{\cup} X_n$"*

*2. The FSM Strong Countable Choice Principle for ZF countable sets, **SCC($\aleph 0$)** has the form "Given any invariant set X, and any FSM countable family $\mathcal{F} = (X_n)_n$ of finitely supported subsets of X which are ZF countable, there exists a finitely supported choice function on $\mathcal{F}$."*

Theorem 3.9. *1. If the set A of atoms is ZF countable, then **SCUT** is inconsistent in FSM.*

*2. If the set A of atoms is not ZF countable, then **SCUT** $\Rightarrow$ **SCC($\aleph_0$)**;*

Proof. 1. We consider the countable family $\mathcal{F} = (X_n)_n$ in FSM, where each X_n is a non-empty, ZF countable, finitely supported subset A defined as follows. We define X_n as the set of all injective n-tuples from A. Since A is infinite, it follows that each X_n is non-empty. In FSM, each X_n is equivariant because A is an invariant set and each permutation is a bijective function. Thus, the application of a permutation to an n-tuple of atoms leads to another n-tuple of atoms. Therefore, the family $(X_n)_n$ is equivariant and the mapping $n \mapsto X_n$ is

also equivariant. Moreover, it is obvious to remark that each X_m is a ZF subset of the m times Cartesian product of A. Since A is ZF countable in FSM, we have that the Cartesian product of m copies of A is ZF countable. Moreover, each finitely supported subset of the Cartesian product of m copies of A is also ZF countable. Thus, X_n is ZF countable for all $n \in \mathbb{N}$. From **SCUT**, there exists a finitely supported onto application $f : \mathbb{N} \to \underset{n}{\cup} X_n$. Since f is onto and each X_n is non-empty, we have that $f^{-1}(X_n)$ is a non-empty subset of $\mathbb{N}$ for each $n \in \mathbb{N}$. Consider the function $g : \mathcal{F} \to \cup\mathcal{F}$, defined by $g(X_n) = f(min[f^{-1}(X_n)])$. We claim that $supp(f)$ supports g. Let $\pi \in Fix(supp(f))$. According to Proposition 2.10, and because $\mathbb{N}$ is a trivial nominal set and each element X_n is equivariant, we have $\pi \cdot g(X_n) = \pi \cdot f(min[f^{-1}(X_n)]) = f(\pi \cdot min[f^{-1}(X_n)]) = f(min[f^{-1}(X_n)]) = g(X_n) = g(\pi \star X_n)$, where by $\star$ we denoted the S_A-action on $\mathcal{F}$ and by $\cdot$ we denoted the S_A-action on $\cup\mathcal{F}$. Therefore, g is finitely supported. Moreover, $g(X_n) \in X_n$. Let us denote $f(min[f^{-1}(X_n)])$ by x_n. Obviously, $x_n \in X_n$ and $g(X_n) = x_n$. Let $\pi \in Fix(supp(g))$. According to Proposition 2.10, and because each element X_n is equivariant, we obtain that $\pi \cdot x_n = \pi \cdot g(X_n) = g(\pi \star X_n) = g(X_n) = x_n$. Therefore, each element x_n is supported by $supp(g)$. However, since each x_n is a finite n-tuple of atoms, we have $supp(x_n) = x_n$, $\forall n \in \mathbb{N}$. Since $supp(x_n) \subseteq supp(g)$, $\forall n \in \mathbb{N}$, we obtain $x_n \subseteq supp(g)$, $\forall n \in \mathbb{N}$. Since each x_n has exactly n elements, this contradicts the finiteness of $supp(g)$.

2. If $\mathcal{F} = (X_n)_n$ is an FSM countable family of finitely supported subsets of X, it follows that there exists a one-to-one finitely supported application ψ from $\mathcal{F}$ to $\mathbb{N}$. Since ψ is one-to-one, it follows that $supp(\psi)$ supports each X_n, and so the mapping $n \mapsto X_n$ is finitely supported. The implication **SCUT** $\Rightarrow$ **SCC($\aleph_0$)** can be proved similarly as in Theorem 3.5(1). □

4 Conclusion

FSM is a new framework that generalizes the classical ZF mathematics. It provides adequate tools for modelling (infinite) structures in terms

of finitely supported objects. More precisely, FSM represents a reformulation of the ZF algebra in terms of finitely supported structures, i.e. it is a theory of invariant or finitely supported algebraic structures. FSM has strong connections with the FM permutative model of ZFA set theory [7, 12], the axiomatic theory of FM sets [8], the theory of nominal sets [13] and the theory of generalized nominal sets [6]. All these connections are explained in [2] where we also mentioned several practical applications of FSM in areas such as logic, algebra process calculi, fuzzy sets and abstract interpretation.

The consistency of the several choice principles with the axioms of ZF set theory was studied in depth during the last century. In [4] several choice principles, such as **AC**, **ZL**, **DC**, **CC**, **PCC**, **AC(fin)**, **Fin**, **PIT**, **UFT**, **OP**, **KW**, **RKW** and **OEP**, were proved to be inconsistent in FSM. This paper continues the work in [4] by introducing the concept of countable set in FSM. A set is countable (or more generally, of cardinality at most k) in FSM (in the sense of Definition 3.1) if and only if it is countable (or more generally, of cardinality at most k) in ZF and uniformly supported (i.e. all its elements are supported by the same set of atoms) as a subset of an invariant set. We present several connections between countable unions theorems in FSM and countable choice principles in FSM (Theorem 3.5). A particular case of inconsistency of a countable union theorem in FSM is presented in Theorem 3.9. The results in this paper do not follow immediately from [13] because in [13] the nominal sets are defined over countable sets of atoms, whilst we defined invariant sets over possible non-countable sets of atoms. Moreover, since the theory of invariant sets is consistent even when the set of atoms is not countable, the results presented in this paper do not overlap on some related properties in the basic or in the second Fraenkel models of ZFA set theory which are defined using countable sets of atoms [11]. Furthermore, the assumption from [13] regarding the countability of the set of all atoms is not very well explained. This is because in Proposition 3.6 we proved that the set of atoms can be countable only in ZF, but not in FSM.

It is known that various relationship results between several forms

of countable choice principles and countable union theorems hold in the ZF framework. However, nobody guarantees that such results remain valid in FSM. When working in FSM we cannot involve a ZF theorem in order to prove that a certain countable union theorem or a certain countable choice principle is valid or not. Therefore, all FSM relationship results between various countable choice principles and countable union theorems have to be independently proved according to the finite support requirement. The general methods for translating a ZF result into FSM and for proving the existence of finite supports for several structures are explained in [2]. The related methods were applied in Section 3 of this paper.

References

[1] A. Alexandru, G. Ciobanu. *Mathematics of multisets in the Fraenkel-Mostowski framework*. Bulletin Mathematique de la Societe des Sciences Mathematiques de Roumanie, vol. 58/106, no.1 (2015), pp. 3–18.

[2] A. Alexandru, G. Ciobanu. *Main steps in defining Finitely Supported Mathematics*. 11th ICTERI, Lviv, Ukraine, Revised Selected Papers. CCIS vol. 594, Springer (2016), pp. 73–91.

[3] A. Alexandru, G. Ciobanu. *Abstract interpretations in the framework of invariant sets*. Fundamenta Informaticae, vol. 144 (2016), pp. 1–22.

[4] A. Alexandru, G. Ciobanu. Consistency of choice principles in Finitely Supported Mathematics (submitted).

[5] J. Barwise. *Admissible Sets and Structures: An Approach to Definability Theory*. Perspectives in Mathematical Logic, vol.7, Springer, 1975.

[6] M. Bojanczyk. *Nominal Monoids*. Theory of Computing Systems, vol. 53, no. 2 (2013), pp. 194–222.

[7] A. Fraenkel. *Zu den grundlagen der Cantor-Zermeloschen mengenlehre*. Mathematische Annalen, vol. 86 (1922), pp. 230–237.

[8] M.J. Gabbay, A.M. Pitts. *A New Approach to Abstract Syntax with Variable Binding*. Formal Aspects of Computing, vol. 13 (2001), pp. 341–363.

[9] R. Gandy. *Church's thesis and principles for mechanisms*. In: J.Barwise, H.J.Keisler, K.Kunen (eds.) The Kleene Symposium (1980), North-Holland, pp. 123–148.

[10] F. Klein. *Vergleichende betrachtungen uber neuere geometrische forschungen (A comparative review of recent researches in geometry)*. Mathematische Annalen, vol. 43 (1893), pp. 63–100.

[11] T.J. Jech. *The Axiom of Choice*. Studies in Logic and the Foundations of Mathematics, North-Holland, 1973.

[12] A. Lindenbaum, A. Mostowski. *Uber die unabhangigkeit des auswahlsaxioms und einiger seiner folgerungen*. Comptes Rendus des Seances de la Societe des Sciences et des Lettres de Varsovie, vol. 31 (1938), pp. 27–32.

[13] A.M. Pitts. *Nominal Sets Names and Symmetry in Computer Science*, Cambridge University Press, 2013.

[14] A. Tarski. *What are logical notions?*. History and Philosophy of Logic, vol. 7 (1986), pp. 143–154.

Andrei Alexandru, Gabriel Ciobanu

Affiliation/Institution: Romanian Academy, Institute of Computer Science, Iaşi
Email: andrei.alexandru@iit.academiaromana-is.ro, gabriel@info.uaic.ro

Proceedings of the Conference on Mathematical Foundations of Informatics
MFOI2016, July 25-29, 2016, Chisinau, Republic of Moldova

Distances on Monoids of Strings and Their Applications

Mitrofan M. Cioban, Ivan A. Budanaev

Abstract

In this article it is proved that for any quasimetric d on alphabet A there exists a maximal invariant extension ρ^* on the free monoid $L(A)$ of all strings. If d is a discrete metric, then the metric ρ^* allows for a special decomposition of two strings, which is important in solving the approximate string matching problem.

Keywords: free monoid, invariant distance, quasimetric, Levenshtein distance, Hamming distance.

1 Introduction

Dynamic transition of our technological civilization to digital processing and data transmission systems created many problems in the design of modern systems in computer science and telecommunications. Providing robustness and noise immunity is one of the most important and difficult tasks in the data transmission, recording, playback and storage. The distance between information plays nowadays a paramount role in mathematics, computer science and other interdisciplinary research areas. The first among many scientists in the field, who presented the theoretical solutions to error detecting and error correction problems, were C. Shannon, R. Hamming and V. Levenshtein (see [12, 7, 8]). We begin this section with introductions into the field, mainly about abstract monoid of strings $L(A)$.

A monoid is a semigroup with an identity element. Fix a non-empty set A. The set A is called an alphabet. Let $L(A)$ be the set of all finite strings $a_1a_2\dots a_n$ with $a_1, a_2, \dots, a_n \in A$. Let ε be the empty string. Consider the strings $a_1a_2\dots a_n$ for which $a_i = \varepsilon$ for some $i \leq n$. If $a_i \neq \varepsilon$, for any $i \leq n$ or $n = 1$ and $a_1 = \varepsilon$, the string $a_1a_2\dots a_n$ is called a canonical string. The set $Sup(a_1a_2\dots a_n) = \{a_1, a_2, \dots, a_n\} \cap A$ is the support of the string $a_1a_2\dots a_n$ and $l(a_1a_2\dots a_n) = |Sup(a_1a_2\dots a_n)|$ is the length of the string $a_1a_2\dots a_n$. For two strings $a_1\dots a_n$ and $b_1\dots b_m$, their product(concatenation) is $a_1\dots a_nb_1\dots b_m$. If $n \geq 2, i < n$ and $a_i = \varepsilon$, then the strings $a_1\dots a_n$ and $a_1\dots a_{i-1}a_{i+1}\dots a_n$ are considered equivalent. In this case any string is equivalent to one unique canonical string. We identify the equivalent strings. In this case $L(A)$ becomes a monoid with identity ε. Let $Sup(a, b) = Sup(a) \cup Sup(b) \cup \{\varepsilon\}$, and $Sup(a, a) = Sup(a) \cup \{\varepsilon\}$.

It is well known that any subset $L \subset L(A)$ is an abstract language over the alphabet A.

2 Distances on spaces

Let A be a non-empty set and $d : X \times X \to \mathbb{R}$ be a mapping such that for all $x, y \in X$ we have:

(i_m) $d(x, y) \geq 0$;

(ii_m) $d(x, x) = 0$.

Then (X, d) is called a *pseudo-distance space* and d is called a *pseudo-distance* on X. If

(iii_m) $d(x, y) + d(y, x) = 0$ if and only if $x = y$,

then (X, d) is called a *distance space* and d is called a *distance* on X.

If

(iv_m) $d(x, y) = 0$ if and only if $x = y$,

then (X, d) is called a *strong distance space* and d is called a *strong distance* on X.

General problems in distance spaces were studied by distinct authors (see [1, 2, 10, 5, 6]). The notion of a distance space is more

general than the notion of o-metric spaces in sense of A. V. Arhangelskii [1] and S. I. Nedev [10]. A distance d is an o-metric if from $d(x, y) = 0$ it follows that $x = y$, i.e. d is a strong distance.

Let X be a non-empty set and d be a pseudo-distance on X. Then

- (X, d) is called a *pseudo-symmetric space* and d is called a *pseudo-symmetric* on X if for all $x, y \in X$ we have

  (v_m) $d(x, y) = d(y, x)$,

- (X, d) is called a *pseudo-symmetric space* and d is called a *symmetric* on X if d is a distance and a pseudo-symmetric,

- (X, d) is called a *pseudo-quasimetric space* and d is called a *pseudo-quasimetric* on X if for all $x, y, z \in X$ we have

  (vi_m) $d(x, z) \leq d(x, y) + d(y, z)$,

- (X, d) is called a *pseudo-metric space* and d is called a *pseudo-metric* if d is a pseudo-symmetric and a pseudo-quasimetric simultaneously,

- (X, d) is called a *metric space* and d is called a *metric* if d is both symmetric and quasimetric,

- a distance d is called discrete if $d(x, y) \in \omega = \{0, 1, 2, \ldots\}$ for all $x, y \in X$.

Let G be a semigroup and d be a pseudo-distance on G. The pseudo-distance d is called

- Left (respectively, right) invariant if $d(xa, xb) \leq d(a, b)$ (respectively, $d(ax, bx) \leq d(a, b)$) for all $x, a, b \in G$,

- Invariant if it is both left and right invariant.

A distance d on a semigroup G is called *stable* if $d(xy, uv) \leq d(x, u) + d(y, v)$ for all $x, y, u, v \in G$.

Proposition 2.1. *Let d be a pseudo-quasimetric on a semigroup G. The next assertions are equivalent:*

1. *d is invariant.*
2. *d is stable.*

3 Extension of pseudo-quasimetrics on free monoids

Fix an alphabet A and let $\bar{A} = A \cup \{\varepsilon\}$. We assume that $\varepsilon \in \bar{A} \subseteq L(A)$ and ε is the identity of the monoid $L(A)$. Let ρ be a pseudo-quasimetric on the set $\bar{A}$. Let $Q(\rho)$ be the set of all stable pseudo-quasimetrics d on $L(A)$ for which $d(x,y) \leq \rho(x,y)$ for all $x, y \in \bar{A}$. The set $Q(\rho)$ is non-empty, since it contains the trivial pseudo-quasimetric $d(x,y) = 0$ for all $x, y \in L(A)$. For all $a, b \in L(A)$ let $\hat{\rho}(a,b) = sup\{d(a,b) : d \in Q(\rho)\}$. We say that $\hat{\rho}$ is the maximal stable extension of ρ on $L(A)$.

Property 3.1. $\hat{\rho} \in Q(\rho)$.

Proof. Let $d \in Q(\rho)$. Fix two points $a, b \in L(A)$. There is a $n \in \mathbb{N}$ and $x_1, y_1, x_2, y_2, \ldots, x_n, y_n \in \bar{A}$ such that $a = x_1 x_2 \ldots x_n$ and $b = y_1 y_2 \ldots y_n$. Then

$$d(a,b) \leq \Sigma\{d(x_i, y_i) : i \leq n\} \leq \Sigma\{\rho(x_i, y_i) : i \leq n\}.$$

Hence

$$\begin{aligned} \hat{\rho}(a,b) &\leq sup\{\Sigma\{d(x_i, y_i) : i \leq n\} : d \in Q(\rho)\} \\ &\leq \Sigma\{\rho(x_i, y_i) : i \leq n\} \\ &< +\infty. \end{aligned}$$

Therefore $\hat{\rho}$ is a stable pseudo-quasimetric from the set $Q(\rho)$. □

For any $r > 0$ let $d_r(a,a) = 0$ and $d_r(a,b) = r$ for all distinct points $a, b \in L(A)$. Then d_r is an invariant metric on $L(A)$.

Property 3.2. *Let $r > 0$ and $\rho(x, y) \geq r$ for all distinct points $x, y \in A$. Then $\hat{\rho}$ is a quasimetric on $L(A)$, $d_r \in Q(\rho)$ and $\hat{\rho}(a, b) = r$ for all distinct points $a, b \in L(A)$.*

For any $a, b \in L(A)$ let

$$\bar{\rho}(a, b) = inf\{\Sigma\{\rho(x_i, y_i) : i \leq n\}\},$$

where $n \in \mathbb{N} = \{1, 2, \ldots\}$, $x_1, y_1, x_2, y_2, \ldots, x_n, y_n \in \bar{A}$, $a = x_1 x_2 \ldots x_n, b = y_1 y_2 \ldots y_n$. Next, we put

$$\rho^*(a, b) = inf\{\bar{\rho}(a, z_1) + \cdots + \bar{\rho}(z_i, z_{i+1}) + \cdots + \bar{\rho}(z_n, b)\},$$

where $n \in \mathbb{N}, z_1, z_2, \ldots, z_n \in L(A)$.

Property 3.3. *$\bar{\rho}$ is a pseudo-distance on $L(A)$ and $\bar{\rho}(x, y) \leq \rho(x, y)$ for all $x, y \in \bar{A}$.*

Proof. Obviously, $\bar{\rho}$ is a pseudo-distance and $\bar{\rho}(x, y) \leq \rho(x, y)$ for all $x, y \in \bar{A}$. □

Property 3.4. *$\bar{\rho}(x, y) = \rho(x, y)$ for all $x, y \in X$.*

Proof. Assume that $n \in \mathbb{N}$, $x_1, y_1, x_2, y_2, \ldots, x_n, y_n \in \bar{A}$, $x = x_1 x_2 \ldots x_n$ and $y = y_1 y_2 \ldots y_n$. There are $i, j \leq n$ for which $x = x_i$ and $y = y_j$. If $i = j$, then $\Sigma\{\rho(x_i, y_i) : i \leq n\} \geq \rho(x_i, y_i) = \rho(x, y)$. If $i \neq j$, as was mentioned in Proposition 2.1, we have $x_j = y_i = e$. Hence

$$\sum_{i \leq n}\{\rho(x_i, y_i)\} \geq \rho(x_i, y_i) + \rho(x_i, y_i) + \rho(x_j, y_j) = \rho(x_i, \varepsilon) + \rho(\varepsilon, y_j)$$
$$\geq \rho(x, y).$$

The proof is complete. □

Property 3.5. *The pseudo-distance $\bar{\rho}$ is invariant on $L(A)$.*

Proof. Fix $a, b, c \in L(A)$ and $r > 0$. Let $c = z_1 z_2 \dots z_m$. There is a $n \in \mathbb{N}$ and the strings $a = x_1 x_2 \dots x_n$, $b = y_1 y_2 \dots y_n$ such that $\bar{\rho}(a, b) \leq \Sigma\{\rho(x_i, y_i) : i \leq n\} < \rho(a, b) + r$. Then

$$\begin{aligned}\bar{\rho}(ac, bc) &= \bar{\rho}(x_1 x_2 \dots x_n z_1 z_2 \dots z_m, y_1 y_2 \dots y_n z_1 z_2 \dots z_m) \\ &\leq \Sigma\{\rho(x_i, y_i) : i \leq n\} < \bar{\rho}(a, b) + r.\end{aligned}$$

Hence $\bar{\rho}(ac, bc) \leq \bar{\rho}(a, b)$. The proof of inequality $\bar{\rho}(ca, cb) \leq \bar{\rho}(a, b)$ is similar. Proposition 2.1 completes the proof. □

Property 3.6. *The pseudo-distance ρ^* is a stable pseudo-quasimetric on $L(A)$ and $\rho^* \in Q(\rho)$.*

Proof. Follows from Proposition 2.1, Properties 3.3 and 3.5. □

Property 3.7. *If ρ is a quasimetric on X, then $\bar{\rho}$ is a distance on $L(A)$.*

Proof. Assume that ρ is a quasimetric on $\bar{A}$ and $\bar{\rho}$ is not a distance on $L(A)$. There are two distinct points $b, c \in L(A)$ such that $\bar{\rho}(b, c) = \bar{\rho}(c, b) = 0$. Suppose that $n \geq 2$ and $l(b) + l(c) \leq n$. Then

$$\bar{\rho}(b, c) = inf\{\Sigma\{\rho(x_i, y_i) : i \leq m\}\}$$

for $m \in \mathbb{N}, m \leq 4n^2, x_1, x_2, \dots, x_m \in Sup(b, b)$, $y_1, y_2, \dots, y_m \in Sup(c, c)$, $b = x_1 x_2 \dots x_m$, $c = y_1 y_2 \dots y_m$.

Since $\bar{\rho}(b, c) = 0$, there is a $m \in \mathbb{N}$, $x_1, x_2, \dots, x_m \in Sup(b, b)$, $y_1, y_2, \dots, y_m \in Sup(c, c)$ such that $b = x_1 x_2 \dots x_m$, $c = y_1 y_2 \dots y_m$ and $\bar{\rho}(b, c) = \{\Sigma\{\rho(x_i, y_i) : i \leq m\} = 0$. Since $\bar{\rho}(c, b) = 0$, there is a $k \in \mathbb{N}$, $c_1, c_2, \dots, c_k, \in Sup(c, c)$, $b_1, b_2, \dots, b_k \in Sup(b, b)$ such that $b = b_1 b_2 \dots b_k$, $c = c_1 c_2 \dots c_k$ and $\bar{\rho}(c, b) = \{\Sigma\{\rho(c_j, b_j) : j \leq k\} = 0$.

Fix $i_1 \leq m$, then $\rho(x_{i_1}, y_{i_1}) = 0$. There is j_1 such that $c_{j_1} = y_{i_1}$. Then $\rho(c_{j_1}, b_{j_1}) = 0$. There is i_2 such that $x_{i_2} = b_{j_1}$. Then $\rho(x_{i_2}, y_{i_2}) = 0$ and so on. As a result we obtain a sequence x_{i_1}, $y_{i_1} = c_{j_1}$, $b_{j_1} = x_{i_2}$, $y_{i_2} = c_{j_2}$, …, x_{i_p}, $y_{i_p} = c_{j_p}$, $b_{j_p} = x_{i_{p+1}}$, $y_{i_{p+1}} = c_{j_{p+1}}$, … such that $\rho(x_{i_p}, y_{i_p}) = \rho(c_{j_p}, b_{j_p}) = 0$ for any $p \in \mathbb{N}$. Since $x_{i_1}, x_{i_2}, \dots, x_{i_p}, \dots$ are elements of a finite set $Sup(b, b) = Sup(b) \cup \{\varepsilon\}$, there are two

numbers $p, q \in \mathbb{N}$ such that $q < p$ and $x_{i_q} = x_{i_p}$. Hence $\rho(x_{i_q}, y_{i_q}) = 0$ and

$$\begin{aligned} 0 &\leq \rho(y_{i_q}, x_{i_q}) = \rho(y_{i_q}, x_{i_p}) \\ &\leq \rho(y_{i_q}, c_{j_q}) + \rho(c_{j_q}, b_{j_q}) + \cdots + \rho(c_{j_{p-1}}, b_{p_{p-1}}) + \rho(b_{j_{p-1}}, x_{i_p}) \\ &= 0, \end{aligned}$$

a contradiction. The proof is complete. □

Property 3.8. *Let $a, b \in L(A)$ be two distinct points in $L(A)$ and $r(a,b) = min\{\rho(x,y) : x \in Sup(a,a), y \in Sup(b,b), x \neq y\}$. Then $\hat{\rho}(a,b) = \rho^*(a,b) \geq r(a,b)$.*

Proof. Assume that $r(a,b) - \rho^*(a,b) = 3\delta > 0$. There exist $n \in \mathbb{N}$ and $z_1, z_2, \ldots, z_n \in L(A)$ such that

$$\rho^*(a,b) \leq \bar{\rho}(a,z_1) + \cdots + \bar{\rho}(z_i, z_{i+1}) + \cdots + \bar{\rho}(z_n, b) < \rho^*(a,b) + \delta.$$

Let $z_0 = a$ and $z_{n+1} = b$. For each $i \in \{0,1,2,\ldots,n\}$ there are representations $z_i = u_{(i,1)}u_{(i,2)}\ldots u_{(i,m_i)}$ and $z_{i+1} = v_{(i,1)}v_{(i,2)}\ldots v_{(i,m_i)}$ such that $\{u_{(i,1)}, u_{(i,2)}, \ldots, u_{(i,m_i)}\} \subseteq Sup(z_i, z_i)$, $\{v_{(i,1)}, v_{(i,2)}, \ldots, v_{(i,m_i)}\} \subseteq Sup(z_{i+1}, z_{i+1})$ and $\bar{\rho}(z_i, z_{i+1}) \leq \Sigma\{\rho(u_{(i,j)}, v_{(i,j)} : j \leq m_i\} \leq \bar{\rho}(z_i, z_{i+1}) \leq \delta/(n+1)$. Without loss of generality, assume that there is $m \in \mathbb{N}$ such that $m_i = m$ for each $i \in \{0,1,2,\ldots,n\}$. Then for each $i \in \{0,1,2,\ldots,n\}$ there is a one-to-one mapping $h_i : \{1,2,\ldots,m\} \longrightarrow \{1,2,\ldots,m\}$ such that $v_{(i,j)} = u_{(i+1,h_i(j))}$ for each $j \leq m$. Then for any $j \leq m$ we have the chain $j_0 = j$, $j_1 = h_1(j)$, $j_2 = h_2(j_1)$, …, $j_n = h_n(j_{n-1})$ and number $r_j = \rho(u_{(0,j_0)}, v_{(0,j_0)}) + \rho(u_{(1,j_1)}, v_{(1,j_1)}) + \cdots + \rho(u_{(n,j_n)}, v_{(n,j_n)}) \geq \rho(u_{(0,j_0)}, v_{(n,j_n)})$. Let $h(j) = j_n$, then $h : \{1,2,\ldots,m\} \longrightarrow \{1,2,\ldots,m\}$ is a one-to-one mapping as the composition of the mappings $h_1, h_2, \ldots, h_n$. We obtain that

$$\rho^*(a,b) + 3\delta \leq \bar{\rho}(a,z_1) + \ldots + \bar{\rho}(z_i, z_{i+1}) + \cdots + \bar{\rho}(z_n, b) \geq \bar{\rho}(a,b) \geq r(a,b).$$

The proof is complete. □

The following properties follow from Property 3.8.

Property 3.9. *If ρ is a quasimetric on $\bar{A}$, then ρ^* and $\hat{\rho}$ are quasimetrics on $L(A)$.*

Property 3.10. *If ρ is a strong quasimetric on $\bar{A}$, then ρ^* and $\hat{\rho}$ are strong quasimetrics on $L(A)$.*

Property 3.11. *Let ρ be a pseudo-quasimetric on A, Y- a subspace of A and $\varepsilon \in \bar{Y}$. Let $M(Y) = L(Y)$ be the submonoid of the monoid $L(A)$ generated by the set Y, and by d_Y- the extension $\hat{\rho|Y}$ on $M(Y)$ of the pseudo-quasimetric ρ_Y on Y, where $\rho_Y(y,z) = \rho(y,z)$ for all $y,z \in \bar{Y}$. Then*

1. *$d_Y(a,b) = \hat{\rho}(a,b)$ for all $a,b \in M(Y)$,*
2. *If ρ is a (strong) quasimetric on Y, then $\hat{\rho}$ is a (strong) quasimetric on $M(Y)$,*
3. *If ρ is a metric on Y, then $\hat{\rho}$ is a metric on $M(Y)$,*
4. *If $a,b \in L(A)$ are distinct points and ρ is a quasimetric on $Sup(a,b)$, then $\hat{\rho}(a,b) + \hat{\rho}(b,a) > 0$,*
5. *If $a,b \in L(A)$ are distinct points and ρ is a strong quasimetric on $Sup(a,b)$, then $\hat{\rho}(a,b) > 0$ and $\hat{\rho}(b,a) > 0$,*
6. *For any $a,b \in L(A)$ there exist $n \in \mathbb{N}$, $x_1, x_2, \ldots, x_n \in Sup(a,a)$ and $y_1, y_2, \ldots, y_n \in Sup(b,b)$ such that $a = x_1 x_2 \cdots x_n$, $b = y_1 y_2 \cdots y_n$ ρ, $n \leq (l(a) + l(b) + 1)^2$ and $\bar{\rho}(a,b) = \Sigma\{\rho(x_i, y_i) : i \leq n\}$,*
7. *$\hat{\rho} = \bar{\rho} = \rho^*$.*

Property 3.12. *For any $a{=}a_1a_2...a_n$ we put $a^{-1}{=}a_n...a_2a_1$. Then $\rho^*(a,b){=}\rho(a^{-1},b^{-1})$ and $(ab)^{-1}{=}b^{-1}a^{-1}$ for all $a,b \in L(A)$.*

Stable metrics on free algebras were considered in [2]. Invariant quasimetrics on free groups were constructed in [3] and [11].

4 Discrete distances on $L(A)$

Fix an alphabet A and $\bar{A}{=}A\cup\{\varepsilon\}$. Consider on A some linear ordering for which $\varepsilon < x$ for any $x \in A$. On $\bar{A}$ consider the following distances ρ_l, ρ_r, ρ_s, where $\rho_l(x,x) = \rho_r(x,x) = 0$ for any $x \in \bar{A}$; if $x, y \in \bar{A}$ and $x < y$, then $\rho_l(x,y) = 1, \rho_l(y,x) = 0$, $\rho_r(x,y) = 0, \rho_r(y,x) = 1$, $\rho_s(x,y) = \rho_l(x,y) + \rho_r(x,y)$. By construction, ρ_l and ρ_r are quasimetrics and ρ_s is a metric on $\bar{A}$. Then ${\rho_l}^*(x,y)$ and ${\rho_r}^*(x,y)$ are invariant discrete quasimetrics on $L(A)$ and ${\rho_s}^*$ is a discrete invariant metric on $L(A)$. This metric we consider below.

Theorem 4.1 *Let ρ be a quasimetric on $\bar{A}$, and $\rho(a,\varepsilon) = \rho(b,\varepsilon)$ for all $a, b \in A$. Then $\rho^*(ac,bc) = \rho^*(ca,cb) = \rho^*(a,b)$ for all $a, b, c \in L(A)$.*

Proof. It is suffiecient to prove the assertion of the theorem for $c \in A$. Assume that $\rho^*(ac,bc) = r < \rho^*(a,b)$, where $a, b \in L(A)$ and $c \in A$. Then, by definition, there exist the representations $ac = x_1x_2\cdots x_p$ and $bc = y_1y_2\cdots y_p$, such that $\rho^*(ac,bc) = \Sigma\{d(x_i,y_i) : i \le p\}$.

Case 1: $x_p = y_p = c$.
In this case $a = x_1x_2\cdots x_{p-1}$, $b = y_1y_2\cdots y_{p-1}$ and $\rho^*(a,b) \le \Sigma\{d(x_i,y_i) : i \le p-1\} = \Sigma\{d(x_i,y_i) : i \le p\} = \rho^*(ac,bc)$, a contradiction.

Case 2: $x_p = c$ and $y_p \ne c$.
Then $y_p = \varepsilon$, and there exists $q < p$ for which $y_q = c$ and $y_j = \varepsilon$ for $j > q$. We put $y'_i = y_i$ for $i \ne q$ and $y'_q = \varepsilon$. Then $a = x_1x_2\cdots x_{p-1}$, $b = y'_1y'_2\cdots y'_{p-1}$, $\rho^*(a,b) \le \Sigma\{d(x_i,y'_i) : i \le p-1\} = \Sigma\{d(x_i,y_i) : i < q\} + d(x_q,\varepsilon) + \Sigma\{d(x_i,y_i) : q < i < p\}$. We have that $d(x_q,\varepsilon) \le d(c,\varepsilon) = d(x_p,y_p)$ and, by definition, $d(x_q,y_q) \ge 0$. Then we have that $\rho^*(a,b) \le \Sigma\{d(x_i,y_i) : i < q\} + d(x_q,y_q) + \Sigma\{d(x_i,y_i) : q < i < p\} + d(x_p,y_p) = \Sigma\{d(x_i,y_i) : i \le p\} = \rho^*(ac,bc)$, a contradiction.

Case 3: $x_p \ne c$ and $y_p = c$.
This case is similiar to Case 2.
Therefore, we proved that $\rho^*(ac,bc) = \rho^*(a,b)$ for all $a, b, c \in L(A)$. By virtue of Property 3.12, we have $\rho^*(ca,cb) = \rho^*(a^{-1}c^{-1}, b^{-1}c^{-1}) = \rho^*(a^{-1},b^{-1}) = \rho^*(a,b)$ for all $a, b, c \in L(A)$. The proof is complete. □

Corollary 4.2 *If* $\rho^* = \rho_s^*$, *then* $\rho^*(ac, bc) = \rho^*(ca, cb) = \rho^*(a, b)$ *for all* $a, b, c \in L(A)$.

5 Parallel decompositions of two strings

The longest common substring and pattern matching in two or more strings is a well known class of problems. From this point of view, for any two strings $a, b \in L(A)$ we find the decompositions of the form $a = v_1u_1v_2u_2 \cdots v_ku_kv_{k+1}$ and $b = w_1u_1w_2u_2 \cdots w_ku_kw_{k+1}$, where

- u_i is a canonical substring of the strings a and b, and u_i may be an empty string;
- v_j is a canonical substring of a and v_j may be an empty string, i.e. $v_j = \varepsilon$;
- w_j is a canonical substring of b and w_j may be an empty string.

The decompositions of the above form are called the *parallel decompositions* of the strings a and b. For any parallel decompositions $a = v_1u_1v_2u_2 \cdots v_ku_kv_{k+1}$ and $b = w_1u_1w_2u_2 \cdots w_ku_kw_{k+1}$ the number

$$E(v_1u_1 \cdots v_ku_kv_{k+1}, w_1u_1 \cdots w_ku_kw_{k+1}) = \sum_{i \leq k+1} \{\max\{l(v_i), l(w_i)\}\}$$

is called the efficiency of the given parallel decompositions. The number $E(a, b)$ is equal to the minimum of the efficiencies of all parallel decompositions of the strings a, b and is called the *common efficiency of the strings a,b*. It is obvious that $E(a, b)$ is well determined. We say that the parallel decompositions $a = v_1u_1v_2u_2 \cdots v_ku_kv_{k+1}$ and $b = w_1u_1w_2u_2 \cdots w_ku_kw_{k+1}$ are optimal if $E(v_1u_1v_2u_2 \cdots v_ku_kv_{k+1}, w_1u_1w_2u_2 \cdots w_ku_kw_{k+1}) = E(a, b)$. This type of decompositions are associated with the problem of approximate string matching [9].

On $L(A)$ we consider the maximal invariant discrete metric $\rho^* = \rho_s^*$, where $\rho^*(x, y) = 1$ for distinct $x, y \in \bar{A} \subseteq L(A)$.

Theorem 5.1. *Let $a, b \in L(A)$ and $n = max\{l(a), l(b)\}$. Then*

$$\rho^*(a,b) = E(a,b) \leq n.$$

Proof. It is obvious that $E(a,b) \leq n$ and $\rho^*(a,b) \leq n$. Let $E(a,b) < n$. Then we fix a pair of optimal parallel decompositions $a = v_1u_1v_2u_2 \cdots v_ku_kv_{k+1}$ and $b = w_1u_1w_2u_2 \cdots w_ku_kw_{k+1}$. Since $\rho^*(v_i, w_i) \leq max\{l(v_i), l(w_i)\}$, in this case $\rho^*(a,b) \leq \Sigma\{\rho^*(v_i, w_i) : i \leq k+1\} \leq E(a,b) \leq n$. Assume now that $\rho^*(a,b) \leq n-1$. Then there exist the representations $a = x_1x_2 \cdots x_p$, $b = y_1y_2 \cdots y_p$ such that $\rho^*(a,b) = \Sigma\{d(x_i, y_i) : i \leq p\}$. Let $P = \{i : x_i = y_i\}$. The set P is non-empty. If i, j are natural numbers and $i \leq j$, then $[i,j] = \{s : i \leq s \leq j\}$. Then there exist numbers $n_1, m_1, \ldots, n_k, m_k \in P$ such that

1. $n_i \leq m_i$ for any $i \leq k$,
2. $P = \cup\{[n_i, m_i] : i \leq k\}$.

Then $u_i = \{x_j : n_i \leq j \leq m_i\} = \{y_j : n_i \leq j \leq m_i\}$. We put $v_1 = \{x_i : i < n_1\}, w_1 = \{y_i : i < n_1\}, v_2 = \{x_i : m_1 < i < n_2\}, w_2 = \{y_i : m_1 < i < n_2\}, \ldots, v_{k+1} = \{x_i : m_k < i < n\}, w_{k+1} = \{y_i : m_k < i < n\}$. We obtain two parallel decompositions of a, b for which $E(v_1u_1v_2u_2 \cdots v_ku_kv_{k+1}, w_1u_1w_2u_2...w_ku_kw_{k+1}) \leq \rho^*(a,b)$. Hence $E(a,b) \leq \rho^*(a,b)$. The proof is complete. □

6 Relations to Hamming and Levenshtein Distances

If $a, b \in L(a,b)$ and $a = a_1a_2 \cdots a_n, b = b_1b_2 \cdots b_m$ are the canonical decompositions, then for $m \leq n$ the number

$$d_H(a,b) = d_H(b,a) = |\{i \leq m : a_i \neq b_i\}| + n - m$$

is called the *Hamming distance* [7] between strings a and b.

The *Levenshtein distance* [8] between two strings $a = a_1a_2 \cdots a_n$ and $b = b_1b_2 \cdots b_m$ is defined as the minimum number of insertions,

deletions, and substitutions required to transform one string to the other. A formal definition of Levenshtein's distance $d_L(a,b)$ is given by the following formula:

$$d_L(a_1\cdots a_i, b_1\cdots b_j)= \begin{cases} i, & \text{if j=0,} \\ j, & \text{if i=0,} \\ \min \begin{cases} d_L(a_1\cdots a_{i-1}, b_1\cdots b_j)+1 \\ d_L(a_1\cdots a_i, b_1\cdots b_{j-1})+1 \\ d_L(a_1\cdots a_{i-1}, b_1\cdots b_{j-1})+1_{(a_i\neq b_j)}, \end{cases} \end{cases}$$

where $1_{(a_i\neq b_j)}$ equals to 0 if $a_i = b_j$ and to 1 otherwise.

Theorem 6.1. *$d_L(a,b) = \rho^*(a,b) \leq d_H(a,b)$ for any $a, b \in L(A)$.*

Proof. To prove the equality $d_L(a,b) = \rho^*(a,b)$, we will first prove that $d_L(a,b) \leq \rho^*(a,b)$, and then that $d_L(a,b) \geq \rho^*(a,b)$.

We begin with the observation that the parallel decompositions of two strings a, b allow the evaluation of the Levenshtein distance $d_L(a,b)$. If $a = v_1u_1v_2u_2\cdots v_n$ and $b = w_1u_1w_2u_2\cdots w_n$ are optimal parallel decompostions, then for transformation of b to a is sufficient to transform any w_i to v_i. The cost of transformation of w_i to v_i is $\leq \max\{l(w_i), l(v_i)\}$. Hence $d_L(a,b) \leq \rho^*(a,b)$.

The proof of the inequality $d_L(a,b) \geq \rho^*(a,b)$ is based on the Levenshtein distance formula, as well as the construction of the transformation of string a to string b. We observe that the Levenshtein distance is calculated recursively using the *memoization* matrix and *dynamic programming* technique [4, pp. 350–355]. A small snapshot of the memoization matrix calculation is presented below.

Table 1. Construction of memoization matrix for Levenshtein distance

Diag	Above
Left	min(Above + delete, Left + insert, Diag + $1_{a_i\neq b_j}$)

Distance d_L calculated on subtrings $a_1\cdots a_i$ of string a and substring $b_1\cdots b_j$ of string b is equal the minimum of the following values:

- $d_L(a_1 \cdots a_{i-1}, b_1 \cdots b_j) + 1,$ (1)
- $d_L(a_1 \cdots a_i, b_1 \cdots b_{j-1}) + 1,$ (2)
- $d_L(a_1 \cdots a_{i-1}, b_1 \cdots b_{j-1}) + 1_{a_i \neq b_j}.$ (3)

Note : operation (1) is the delete operation, (2) is the insert operation, and operation (3) is the substitution operation.

Once all of the above values are calculated and the memoization matrix is filled, the distance is given by the value in the cell on the n^{th} row and m^{th} column.

The construction of the transformation of string a into string b is based on the values of the memoization matrix. At each point of the construction process, we will execute operations on both strings a and b, and obtain another pair of strings a' and b' equivalent to the initial pair a and b. We use the top-down analysis approach to describe the transformation process step by step. The process below starts with $i = n, j = m$, $p = 0$, $q = 0$ and both a', b' as empty strings:

- if when calculating $d_L(a_1 \cdots a_i, b_1 \cdots b_j)$ we used operation (1), then we deleted a character from string a at position i, which is equivalent to inserting the ε character in string b at the corresponding position. In this case, in the building process of a' and b', we put $p := p+1$, $v'_p = \{a_i\}$,$w'_p = \{\varepsilon\}$, $a' := v'_p \cup a'$, $b' := w'_p \cup b'$. Next, we proceed to calculate $d_L(a_1 \cdots a_{i-1}, b_1 \cdots b_j)$.

- if when calculating $d_L(a_1 \cdots a_i, b_1 \cdots b_j)$ we used operation (2), then we inserted the ε character in string a at position i. In this case, in the building process of a' and b', we put $p := p + 1$, $v'_p = \{\varepsilon\}$,$w'_p = \{b_j\}$, $a' := v'_p \cup a'$, $b' := w'_p \cup b'$. Next, we proceed to calculate $d_L(a_1 \cdots a_i, b_1 \cdots b_{j-1})$.

- if when calculating $d_L(a_1 \cdots a_i, b_1 \cdots b_j)$ we used operation (3), then we either substituted the character at position i of string a with the character at position j of string b, or we did not make any change in case if $a_i = b_j$. If $a_i = b_j$, we put $q =: q + 1$,$u'_q = \{a_i\}$, $a' := u'_q \cup a'$, $b' := u'_q \cup b'$. If $a_i \neq b_j$, we put $p =: p+1$, $v'_p = \{a_i\}$,

$w'_p = \{b_j\}$, $a' := v'_p \cup a'$, $b' := w'_p \cup b'$. Next, we proceed to calculate $d_L(a_1 \cdots a_{i-1}, b_1 \cdots b_{j-1})$.

According to the above steps, we observe that string a' is equivalent to string a, and string b' is equivalent to b by construction. But, we also have that the decomposition $a' = v'_p u'_q v'_{p-1} u'_{q-1} \cdots u'_1 v'_1$ and $a' = w'_p u'_q w'_{p-1} u'_{q-1} \cdots u'_1 w'_1$ obtained from the above construction process, represent a parallel decomposition of strings a and b. Thus, we have that $d_L(a,b) = E(a,b) \geq \rho^*(a,b)$. This completes the proof of the equality $d_L(a,b) = \rho^*(a,b)$.

We will now prove the second part of the theorem, namely that $\rho^*(a,b) \leq d_H(a,b)$. Let $d_H(a,b) < max\{l(a), l(b)\} = n$, where $n = l(a) \geq l(b) = m$. Then $a = a_1 a_2 \cdots a_n, b = b_1 b_2 \cdots b_m$, $a_i \neq \varepsilon$ for any $i \leq n$, and or $m = 1$ and $b_1 = \varepsilon$, or $b_j \neq \varepsilon$ for any $j \leq m$. In this case $d_H(a,b) = n - |\{i \leq m : a_i = b_i\}|$ and we have the representations $a = (a_1)(a_2) \cdots (a_m)(a_{m+1} \cdots a_n)$ and $b = (b_1)(b_2) \cdots (b_m)(\varepsilon)$ which generates two parallel decompositions α, β with $E(\alpha, \beta) = d_H(a,b)$.Therefore $\rho^*(a,b) \leq E(\alpha,\beta) = d_H(a,b)$. The proof is complete. □

Corollary 6.2 *Distance d_L is strictly invariant, i.e. $d_L(ac,bc) = d_L(ca,cb) = d_L(a,b)$ for any $a,b,c \in L(A)$.*

Remark 6.3 *The Hamming distance d_H is not invariant.*

Example 6.4 *Let $n = m + p$ and strings $a = (01)^n, b = (10)^m$, $c = (01)^p$. We obtain the following distance values for the above strings:*

$$d_L(a,b) = 2p, \rho^*(a,b) = 2p, d_H(a,b) = 2n,$$
$$d_L(ac,bc) = 2p, \rho^*(ac,bc) = 2p, d_H(ac,bc) = 2n.$$

Example 6.5 *Let $n \geq 2$, $a = (1)^n$, $b = 1$, $c = (0)^{n-1}1$. We obtain the following distance values for the above strings:*

$$d_L(a,b) = n-1, \rho^*(a,b) = n-1, d_H(a,b) = n-1,$$
$$d_L(ac,bc) = n-1, \rho^*(ac,bc) = n-1, d_H(ac,bc) = 2n-1.$$

Remark 6.6 *If $l(a) = l(b)$, then $d_H(ac,bc) = d_H(a,b)$ for any $a,b,c \in L(A)$. Additionally, the following equality always holds:*

$$d_H(ca, cb) = d_H(a, b).$$

7 Conclusion

We showed that there exist invariant distances on $L(A)$ closely related to Levenshtein's distance, which allows to solve approximate string matching problems. The results can be applied in different areas, including data correction of signals transmitted over channels with noise, finding matching DNA sequence after mutations, text searching with possible typing errors, and estimation of the dialects pronounciations proximity [5, 6, 9]. Our distances of ρ^* type can be defined for distinct values $\rho(a, b)$ of strings a,b, in general, and for $\rho(a, b) \neq \rho(b, a)$.

References

[1] Arhangel'skii, A.V., *Mappings and spaces*, Uspehi Matem. Nauk, vol. 21 (1966), vyp. 4, pp. 133–184. (English translation: Russian Math. Surveys, vol. 21 (1966), no. 4, pp. 115–162).

[2] Choban, M.M., *The theory of stable metrics*, Math. Balkanica, vol. 2 (1988), pp. 357–373.

[3] Choban, M.M., Chiriac L.L., *On free groups in classes of groups with topologies,* Bul. Acad. Ştiinţe Repub. Moldova, Matematica (2013), Number 2-3, pp. 61–79.

[4] Cormen, T.H., Leiserson, C.E., Rivest, R.L and Stein, C., *Introduction to Algorithms (2nd ed.)*, MIT Press and McGraw-Hill, 2001.

[5] Deza, M.M., Deza E., *Dictionary of Distances*, Elsevier, Amsterdam, 2006.

[6] Deza, M.M., Deza E., *Encyclopedia of Distances*, Springer, Berlin, 2014.

[7] Hamming, R.W., *Error Detecting and Error Correcting Codes*, Bell System Technical Journal, vol. 29 (1950), no 2, pp. 147–160.

[8] Levenshtein, V.I., *Binary codes capable of correcting deletions*, insertions, and reversals. DAN SSSR, vol. 163 (1965), no 4, pp. 845–848 (in Russian) (English translation: Soviet Physics Doklady, vol. 10 (1965), no 8, pp. 707–710).

[9] Navarro, G., *A guided tour to approximate string matching*. ACM Computing Surveys, vol. 33 (2001,) no 1: pp. 31–88.

[10] Nedev, S.I., *o-metrizable spaces*, Trudy Moskov. Mat.Ob-va **24** (1971), 201–236 (English translation: Trans. Moscow Math. Soc., vol. 24 (1974), pp. 213–247).

[11] Romaguera S., Sanchis M., Tkachenko M., *Free paratopological groups*, Topology Proceed., vol. 27 (2003), no 2, pp. 613–640.

[12] Shannon C., *A Mathematical Theory of Communication*, The Bell System Technical Journal, vol. 27 (1948), pp. 279-423, pp. 623–656.

Mitrofan M. Cioban[1], Ivan A. Budanaev[2]

[1]Professor, Doctor of Science,
Academician of the Academy of Science of Moldova
Tiraspol State University, Republic of Moldova
Email: mmchoban@gmail.com

[2] Doctoral School of Mathematics and Information Science
Institute of Mathematics and Computer Sciences of ASM
Tiraspol State University, Republic of Moldova
Email: ivan.budanaev@gmail.com

On Technology for Digitization of Romanian Historical Heritage Printed in the Cyrillic Script

Svetlana Cojocaru, Lyudmila Burtseva, Constantin Ciubotaru, Alexandru Colesnicov, Valentina Demidova, Ludmila Malahov, Mircea Petic, Tudor Bumbu, Ştefan Ungur

Abstract

This paper describes the elaboration of the technology for digitization of the Romanian historical heritage printed in the Cyrillic script in the 17th–20th centuries.

The attention is focused to transliteration of recognized Cyrillic texts to the modern Latin script, to difficulties with older alphabets that are not fully supported by modern OCR engines, and to other concomitant problems.

We proposed solutions for these problems and integrated them into a corresponding technology and a tool pack that includes: alphabets, dictionaries, glyph patterns, transliteration and glyph restoration utilities, virtual keyboards, fonts, and user's manual.

Keywords: historical Romanian texts, OCR of Romanian Cyrillic scripts, 17th-20th century, software tools for OCR, transliteration utility.

1 Introduction

Problem of digitization and conservation of historic, literary, and cultural treasures represents a domain of priority in the Digital Agenda for Europe. The EU admits the necessity of coordinated efforts in this domain and manifests vast actions to activate this process. These actions include development of the European Digital Library *Europeana*, supported by the European Parliament resolution on the 5th of May, 2010 and the adopted EU Programs for Culture. Diverse aspects of this problem were treated in many European research projects [1]. In particular, the problem

of creation of linguistic resources, digitization and recognition of historic and literary heritage is attended in many European countries [6]–[13]. Regrettably, scientific centers of the Republic of Moldova aren't involved in these actions.

Massive usage of information technologies and communications (ITC) strongly stimulates development of the modern society and substantially contributes to the conception of information society. The Digital Agenda presented by the European Commission forms one of the seven pillars of the Europe 2020 Strategy. It adds dynamics and optimizes ITC benefits for economic growth, creation of new jobs, increase peoples' quality of life.

The Decision of the Government of the Republic of Moldova No. 857 of the 31st of October 2013 approves the National Strategy for the development of information society "Digital Moldova 2020", and the Plan of Actions for implementation of this Strategy. The Program "Creation, development and evaluation of the digital content in the RM in 2016–2020" is in the process of approbation.

Digitization and conservation of the cultural historic and linguistic heritage that includes old literature, archive documents, folklore records, etc., represent one of key domains affected by the Digital Agenda. This process will be related to the heritage preservation while its placement in the Internet will considerably simplify its usage, will extend area and possibilities for research, including in humanitarian domains, through modifications in international media of communication. In addition, execution of the planned works will permit unification, homogenization, and integration of national and cultural media in the international information society, and will confirm status of the Romanian language as language of communication in the European continent.

Although the cultural heritage domain has been intently researched during last decades, today the research will more focus on its multilingual nature and specific for each culture features. Digital age arrival passed the problem of cultural heritage preservation from conservation laboratories to computers. The cultural heritage presented in text form has showed the most suitable and informative digital representations. Texts processing is a highly developed domain today. The research of historical texts has developed specific methods of text processing, mostly, tools for

representation of unusual today scripting. Following the distribution of the Unicode over operating systems, the problem of encoding was solved for any historical script. To materialize old text in electronic form, we need now only specialized fonts covering the corresponding code points. Let's note that several Romanian Cyrillic letters (e.g., ѧ) were included in the Unicode only since 2009. But, being appropriate for preservation of textual cultural heritage, unusual fonts are difficult for perception even for linguistics professionals. Therefore, solving the problem of textual cultural heritage dissemination supposes the development of tools for transliteration or just reading in common script.

Solving these problems for the Republic of Moldova confronts difficulties and specific aspects: the number of existing resources is relatively small but they are kept in many book deposits; they were printed in a lot of diverse alphabets. Thus, old manuscripts and books in Moldova and Romania were produced, as a rule, in the old Romanian Cyrillic script (RC) [2], that differs from standard Church-Slavonic or Russian ones. The definitive formation of RC is dated back to the 17th century. The first Romanian grammar was printed by D. Evstatievici in 1757. Since 1830 until the official adoption of the Latin alphabet (RL) in 1862 several transitional alphabets (TR) were used; they were based on the Simplified Romanian Cyrillic script (SRC) but some letters were Latin [5]. The modern Romanian Latin alphabet (MRL) was adopted in 1904; with small variations, it is used till present. Variants of the Cyrillic alphabet that were used in the Moldavian ASSR in 1924–1940 and in the Moldavian SSR in 1940–1989 (the Moldavian Cyrillic script, MC) were an integral and irregular application of the Russian alphabet for the Romanian language.

Electronic sources exist mostly for old Romanian books printed in the Latin script, while those for the Cyrillic script practically don't exist except as scanned images. That's why the problem appears to create electronic Romanian resources of manuscripts and old books in the Cyrillic script. To create electronic resources of the cultural heritage printed in the corresponding periods we could use, for example, catalogs [3] and [4] from the old book repository at the "A. Lupan" central scientific library of the Academy of Sciences of Moldova (ASM).

This paper describes a technology for digitization and recognition of the historic and linguistic Romanian heritage printed in the Cyrillic script in the 17th–20th centuries. The technology is supported by a pack of the following tools and utilities:

- Alphabets for ABBYY FineReader (AFR).
- Dictionaries (word lists) for AFR.
- Recognition patterns as trained under AFR.
- Utility of transliteration from Cyrillic to Latin and vice versa for MC.
- Conversion utility for TR.
- Conversion utility for RC.
- Font that covers rare glyphs from RC and TR.
- Virtual keyboards.

Algorithm of verification of resulting text in the Latin script and semi-automated word recognition could use the Romanian spellchecker RomSP [15] and reusable linguistic resources [14].

We will concentrate on details of the transliteration and conversion rules.

2 Recognition of the Romanian Cyrillic Script

We began our work as we desired to re-publish some books printed in this alphabet. Under the USSR, the editorial activity produced many useful and interesting texts, but they are of no use to contemporary Romanian audience being printed in the Cyrillic script.

In the period of our interest (1951-1989) the printing quality was quite satisfactory, and scanning goes smoothly. The Moldavian Cyrillic script (MC) was used. It is the Russian alphabet without letters **ё**, **щ**, **ъ** and, since 1967, with one additional letter **ӂ**. At OCR, we were to add letter **ӂ** to the Russian alphabet and provide the dictionary. The dictionary was extracted from recognized texts themselves with manual corrections; then we repeated OCR. See details in [17].

The second referred variant is the Romanian Cyrillic alphabet of 1830–1860. The script was transitional from Cyrillic to Latin (TR). It was Cyrillic in its base with some letters replaced progressively by Latin ones.

We used two approaches to OCR of Romanian transitional scripts. The first approach is to reproduce the scanned text after OCR in its

original glyphs. It is possible with the corresponding AFR configuring and training, and by providing the proper dictionary. It produced up to 7% of erroneous words.

The second approach was invented to solve the problem of alphabet variation. AFR permits to output the result in original glyphs, or replace any glyph by a sequence of letters from the selected alphabet of recognition. AFR proposes this mode for ligatures but it may be used more generally for arbitrary substitution. For TR, we formed a version of the AFR output alphabet that can be set in one-to-one mapping with any transitional alphabet. For example, both **т** (Cyrillic) and **t** (Latin) will be recognized as **t**.

Another problem common for all variants of TR and RC is the absence of their glyphs in the usual system fonts. As the result, we do not see them in AFR dialogs during alphabet preparation, training, manual text correction, etc. The use of glyph substitution solves this problem also [18].

The third period of specific Romanian Cyrillic script usage is since the mid 18th century till 1830 (referred for simplicity as the 18th century). The Romanian typography practices of the 18th century had had two substantial differences from that of the older time, with the same RC of up to 47 letters. First, the usual Arabic number system is used. Second, upper accents had become rare and may be ignored. Therefore, the recognition doesn't imply sophisticated training.

AFR recognizes RC of the corresponding period. Small problems arose due to absence of necessary glyphs in system fonts, as it was already noted

The recognition of texts of the 18th century resulted in 4.5% of erroneous words with original glyphs, and only 3% of erroneous words with ligatures. We observed this effect with transitional scripts also.

The most plausible explanation is that, in the training mode, AFR skips some glyphs that are supposed to be recognized properly. With original glyphs, AFR skips more glyphs, while, at the glyph substitution, AFR should train substituted glyphs and performs more scrupulous training.

A special utility was developed that restores the original glyphs after recognition with substitutions for the texts of the 18th century.

The fourth period covers the 17th century and the 1st half of the 18th century when the Romanian typographies had strictly adhered the previous manual writing practices. This means that the numbers were encoded by letters with special ascending strokes, and accents over the line were substantial. Some words were traditionally printed with abbreviations and were also marked over them. Skipped letters were frequently set over the precedent letter, also with a special marker.

The recognition of such printing implies very subtle and thorough training. For example, each pair of a letter and another letter over it should be trained as a ligature.

Numbers (one or several letters with a marker) should also be trained as ligatures. This increases the number of recognition patterns, but, without ligatures, OCR for RC of the 17th century produces errors in more than 50% words, while with trained ligatures only in 6%.

3 Transliteration of the Recognized Text

3.1 Older Cyrillic Scripts in Unicode

The first problem is presentation of recognized Cyrillic text in computer, especially for TR and RC. In fact, only three fonts in the whole world have old Romanian Cyrillic letters: Kliment STD (ѳѳѢѣѧѧ), Unifont (ѳѵѢѣѧѧ), and Everson Mono (ѳѵѢѣѧѧ), and only since 2009. That's why we are developing for our tool pack our own font covering all necessary Unicode points. MC poses no such problems. In the period of our interest (1951–1989) the difference with the Russian alphabet was made by a single letter ѫ that is presented in commonly used fonts.

Some accented or combined letters are meanwhile missing and should be specially treated, for example, ӑ (ă) or ӗ̆ (ĭă) in TR. To present them in Unicode, it is necessary to use combining accents, and we can't fully reproduce subtle details of the graphical presentation of the original text.

Table 1. Correspondence of Some RC Specific Letters to MRL and Unicode.

Ѣ	Ea	0462	К	C, Ch (before **е**, **и**)	041A
ѣ	ea	0463	к	c, ch (before **е**, **и**)	043A
Ю̆	Ia	0465	Ĭ	Ĭ	012C
ю̆	ia	0464	ĭ	ĭ	012D
Ѧ	Â	0466	Ъ	Ă	042A
ѧ	â	0767	ъ	ă	044A
Ѧ̣	Î, Îm, În	A64E	Щ	Ș̌t	0429
ѧ̣	î, îm, în	A65F	щ	ș̌t	0449
Ꙉ	U	A64A	Ц	G	049F
ꙉ	u	A64B	**ц**	g	044F

3.2 MC: Bidirectional Transliteration

The transliteration MC→MRL was discussed in details in [17]. There are three groups of rules. Most letters (26 of 31) can be mapped one-to-one as shown in Table 2.

Table 2. MC→MRL: one-to-one letter mapping.

MC→MRL		**MC→MRL**		**MC→MRL**		**MC→MRL**	
а	a	з	z	п	p	ц	ț
б	b	и	i	р	r	ш	ș
в	v	й	i	с	s	ь	i
д	d	л	l	т	t	э	ă
е	e	м	m	у	u	ю	iu
ж	j	н	n	ф	f		
ӂ	g	о	o	х	h		

Context rules exist for three letters as shown in Table 3.

Table 3. MC→MRL: Context Rules in the Order of Application.

MC→MRL		Context
г	**gh**	before **е**, **и**, **ь**, **ю**, **я**
г	**g**	otherwise
кс	**x**	exceptions: **eczema** and derivatives, **Alecsandri**
к	**k**	as exception, examples: **kilogram**, **Kogălniceanu**, etc.
к	**ch**	before **е**, **и**, **ь**, **ю**, **я**
к	**c**	otherwise
ч	**c**	before **е**, **и**, **ь**, **я**
ч	**ce**	before **а**
ч	**ci**	otherwise

The letter **ы→â**, **î**, where **î** is written at the beginning or end of words, while **â** inside words. The difficulty is that **î** is kept after prefix, for example, **ne**+**însoți**t = **neînsoți**t (**unaccompanied**).

The letter **я→ea**, **ia**, **a** presents the biggest problem that can't be fully solved without access to dictionaries. Rules are mostly heuristic and statistical, and more than 20 rules do not cover all cases. This situation exists because MC was not thoroughly designed but is an irregular mapping of Romanian sounds to the Russian letters.

There are words that can't be transliterated according to these rules: foreign proper nouns and words of foreign origin that keep their writing in MRL. We use the exception dictionary for them.

The inverse transliteration MRL→MC (1967–1989) was mainly necessary to produce word list in MC from existing word list in MRL. This task equally meets difficulties, mainly with letter **i**. In particular, at the word ends **i** may be omitted, or converted to **и**, **й**, **ь**. Examples: **arici→арич** (**hedgehog**, singular), **arici→аричь** (**hedgehogs**, plural), [**а**] **cheltui→**[**а**] **келтуи** ([**to**] **count**; stress on **i**), [**eu**] **cheltui→**[**еу**] **келтуй** (**I count**: stress on **u**). Analogous problems appear at the transliteration of diphthongs and triphthongs. For example, diphthong **ia→я**, **ия**, **иа**: **soia→соя** (**soybean**), **caucazian→кауказиян** (**Caucasian**), **cartezian→ картезиан** (**Cartesian**).

Some of these problems could be solved by consulting Morpho-Syntactic Data (MSD), which were proposed in the framework of the project *MULTEXT-East* [19]. In the remaining cases, context analysis or even manual intervention could be performed.

The whole transliteration process is implemented as a set of filters each modelling a separate situation. The filters are:

- prefix filters;
- suffix filters;
- diphthong and triphthong filters;
- final filters (**letter→letter**).

Prefix filters are created separately for words that begin with the same letters (**creast*→кряст***, **crea*→креа***, **paie*→пае***, etc.). At transliteration, these filters are applied first.

Suffix filters are common for all words in the lexicon. They can be divided in two classes: conditional depending of the MSD value, and unconditional.

Diphthong and triphthong filters aim to transliteration of some letter combinations like: **ie**, **io**, **eio**, **chio**, etc. They are applied independently of position and context.

Final filters transliterate all letters that remain after application of other filters. For example: **d→д**, **c→к**, **ş→ш**.

Some filters can use rezults from the previous filters. Such filters may look like: **celуй→челуй**, **ieпь→ень**, combining Latin and Cyrillic letters.

If the situation is ambiguous, and expert's intervention (maual selection) is necessary, alternatives can be generated, for example: **кафен**[**иул**][**юл**] with the result **→кафениул** (**brownish**, **coffee color**; with the definite article); **ча**[**иул**][**юл**]**→чаюл** (**the tea**; with the definite article).

This algorithm was applied to the lexicon elaborated at the Al. I. Cuza University in Iași [20]. Automation rate of transliteration was approx. 90%.

3.3 Transitional Alphabets

Sources count approx. 17 versions of TR. In this paper we deal with 36 Cyrillic letters (from 43) that were found in the analyzed texts. Meanwhile, our algorithm permits simple addition of new letters would they be found during the future text analysis. The problem is much

simpler than with MC. Two types of rules are used, simple one-to-one mapping, and context rules.

Transliteration of 32 letters is performed under simple rules (Table 4).

Table 4. TR→MRL: one-to-one letter mapping.

TR→MRL		**TR→MRL**		**TR→MRL**		**TR→MRL**	
а	a	й	i*	т	t	ю	iu
б	b	л	l	ф	f	ѣ	ea
в	v	м	m	х	h	є	ia
д	d	н	n	ц	ţ	ѧ	â
е	e	о	o	ш	ş	ȣ	u*
ж	j	п	p	щ	şt	ĭ	i*
з	g	р	r	ь	i*	ъ	ă
и	i	с	s	э	ă	ѫ	g**

* At linguists' request, rules **й**,**ь**,**ĭ**→ **ĭ** and **ȣ**→**ŭ** may be applied.
Before **e, **i** only.

The four remaining letters are transliterated under context rules (Table 5).

Table 5. TR→MRL: Context Rules in the Order of Application.

TR→MRL		**Context**
г	**gh**	before **е**, **и**, **ю**
г	**g**	otherwise
кс	**x**	exceptions: Table 3
к	**ch**	before **е**, **и**, **ю**
к	**c**	otherwise
ч	**c**	before **е**, **и**
ч	**ce**	before **а**
ч	**ci**	otherwise
ѧ	**î***	before **m**, **n**
ѧ	**îm**	before **b**, **p**
ѧ	**în**	otherwise

*In some texts, always **ѧ**→**î** (simple rule).

3.4 Glyphs and Transliteration Rules for RC

Conform *Gramatica românească* (*The Romanian Grammar*) of 1797 by Radu Tempea, RC contains 43 letters: **Аа Бб Вв Гг Дд Єє Жж Ѕѕ Зз Ии Її Кк Лл Мм Нн Оо Пп Рр Сс Тт Ѹѹ Оуоу Фф Хх Ѡѡ Цц Чч Шш Щщ Ъъ Ыы Ьь Ѣѣ Жж Юю Ꙗꙗ Ѧѧ Θθ Ѱѱ Ѯѯ Vv Ѧ̱ѧ̱ Ц̨ц̨**.

The mid line of letters **Нн** and **Ии** in old scripts is inclined from horizontal only slightly, so both may look very like to **Нн**.

Glyphs like **й** and **ѹ** weren't treated as separate letters in RC, but as **и** and **ѹ** with diacritic sign.

Most letters (37) are transliterated under simple rules (Table 6).

Table 6. RC→MRL: one-to-one letter mapping.

RC→MRL		RC→MRL		RC→MRL		RC→MRL	
а	a	л	l	ф	f	ю	iu
б	b	м	m	х	h	ꙗ	ia
в	v	н	n	ѡ	o	θ	t
д	d	о	o	ц	ţ	ѱ	ps
е	e	п	p	ш	ş	ѯ	x
ж	j	р	r	щ	şt	v	i
ѕ	dz	с	s	ъ	ă	ѧ	g*
з	z	т	t	ы	î		
и	i	ѹ	u	ь	i		
ї	i	оу	u	ж	î		

*Before **е**, **і** only.

The remaining 6 letters need context rules (Table 7).

Table 7. RC→MRL: Context Rules in the Order of Application.

RC→MRL		Context
г	**gh**	before **е**, **и**, **ї**, **ю**
г	**g**	otherwise
кс	**x**	
к	**ch**	before **е**, **и**, **ї**, **ю**
к	**c**	otherwise
ч	**c**	before **е**, **и**, **ѣ**
ч	**ce**	before **а**
ч	**ci**	otherwise

ѣ	**e**	after ч; exception **чѣ→cea**
ѣ	**ea**	otherwise
ѧ	**a**	at the beginning of word; after **ї**, **ц**
ѧ	**e**	after **ч**
ѧ	**ea**	after another consonant; at the end of word
ѧ	**ia**	otherwise
ѧ̱	**îm**	before **b**, **p**
ѧ̱	**în**	otherwise

3.5 Examples of Transliteration

Figure 1 presents an example from "William Shakespeare Biography" book of 1849, which illustrates the complexity of the problem. In addition, this example demonstrates how useful the proposed instrument can be for specialists, especially for those who are not familiar with Cyrillic writing. Text of the 18th century is presented in Figure 2.

D. ѵititорї dе амвеле сексе.

Прiїмiцї дѵедстъ традѫкцiе а меа шi ѵiтiцї о кѫ сiнсерiтате, жѫдекжнd деспре джнса къ нѫ аре де скоп а амѫза нiчї а дикжита пе ѵine-ва, ѵi нѫмаї а моралiза. О сокотеск, дѫпе пъререа меа, ка ѫнiкъ дп фелѫл еї; къчї де шi афаръ дъ водевiлѫрї шi дестѫле комедiї ѵе сжит ла лѫмiнъ дп лiмба патрiеi, маї сжит шi оаре-каре трацедiї;' дар сжит преа сiгѫр къ din нiчї ѫна нѫ вецї пѫтеа траце маї мѫлт фолос ка dintr' аѵесте кап d'опере а ле ѵелебрѫлї Шекспiр, дитжiѫл шi неiмiтабiлѫл поет драматiк пжнъ дп секолѫл актѫал. Въ рекоmand дикъ шi ѵiтiреа вiецiї дѵестѫї ценѫ din каре пѫтецi траце ѫи фолос нѫ маї пѫцin прin моралѫл ѵе ea кѫprinde, мѫлцѫmindѫ-въ тот о датъ шi кѫрiосiтатеа прin пътрѫndереа вiртѫцiлор шi вiцiiлор ѵе карактерiсеаzъ пе Фаїмосѫл шi етернiсатѫл Енглезiлор поет. Iар еѫ дndестѫлжndѫ-мъ де зелѫл Domnii- водстре, мъ воiѫ сiлi маї мѫлт спре а въ мѫлцѫmi.

1847, Феврѫарiе 25.

Тома А. Бардат.

D. cititorї de ambele sexe.

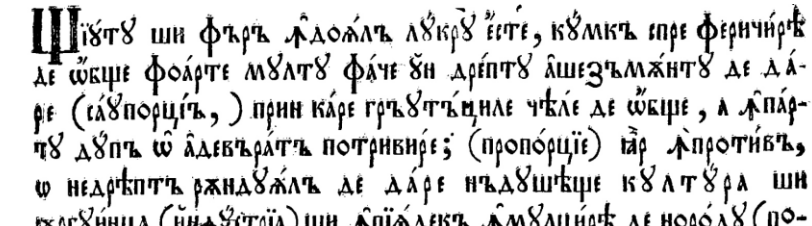
Priїmiţї această traducţie a mea şi citiţї o cu sinceritate, judecând despre dânsa că nu are de scop a amuza nicїї a încânta pe cine-va, ci numaї a moraliza. O socotesc, dupe părerea mea, ca unică în felul eї; căcї de şi afară dă vodeviluрї şi destule comediї ce sânt la lumină în limba patriei, maї sânt şi oare-care traghediї; dar sânt prea sigur că din nicї una nu veţї putea trage maї mult folos ca dintr' aceste cap d'opere a le celebruluї Ṣexpir, întâiul şi neimitabilul poet dramatic până în secolul actual. Vă recomand încă şi citirea vieţiї acestuї geniu din care puteţї trage un folos nu maї puţin prin moralul ce ea cuprinde, mulţumindu-vă tot o dată şi curiositatea prin părtrunderea virtuţilor şi viţiilor ce caracterisează pe Faïmosul şi eternisatul Englezilor poet. Iar eu îndestulându-mă de zelul Domnii- voastre, mă voiu sili maї mult spre a vă mulţumi.

1847, Fevruarie 25.

Toma A. Bagdat.

Figure 1. Translator's Introduction to the Book of 1849 (Biography of Shakespeare, Romeo and Juliette, Othello). In TR.

a)



b)

Щїѹтѹ ши фъръ ѧдоѧлъ лѹкрѹ есте, кѹмкъ спре феричирѣ де ѡбще фоарте мѹлтѹ фаче ѹн дрептѹ ашезъмжнтѹ де даре (саѹпорцїѣ,) прин каре гръѹтъциле чѣле де ѡбще, а ѧпартѹ дѹпъ ѡ адевъратъ потривире;(пропорцїе) ар ѧпротивъ, ѡ недрѣптъ ржндѹалъ де даре нъдѹшѣще кѹлтѹра ши сжргѹинца (индѹстрїа) ши ѧмпїадекъ ѧмѹлцирѣ де нородѹ (попѹлацїа).

c)

Ştiutu şi fără îndoială lucru este, cumcă spre fericirea de obşte foarte multu face un dreptu aşezămîntu de dare (sauporţiă,) prin care grăutăţile cele de obşte, a împartu după o adevărată potrivire; (proporţie) iar împrotivă, o nedreaptă rînduială de dare năduşeşte cultura şi sîrguinţa (industria) şi împiadecă înmulţirea de norodu(populaţia).

Figure 2. Political Text of the 18th Century: a) Image; b) OCR; c) Transliteration (MRL).

3.6 Transliteration utility (Cyrillic→Latin and vice versa)

Formally speaking, transliteration is a system of parametrized rules that are applied to each i-th character x_i of a Romanian word-form X in the Cyrillic script. The result $y_i = \mathrm{Trans}(x_i, \mathrm{Pos}(i, X))$ is a sequence of

characters whose concatenation produces the converted word-form Y in the Latin script.

To avoid ambiguity, the exception dictionary with foreign words, proper nouns, and difficult variants is used.

The accuracy of conversion is up to 95% for MC, up to 96% for TR, and up to 98% for RC. We should conclude that the old Romanian Cyrillic script reflected the word composition mostly accurate.

The utility is written in Java that fully supports Unicode. If the font is properly registered in the operating system, Java programming tools for the interface solve the visualization problem by a simple addressing to this font.

The transliteration utility has a user friendly interface. Files can be opened through menu or by drag-and-drop. The historical period can be selected by user or auto-detected. Supported file formats are TXT, RTF, DOC, DOCX.

The inverse transliteration (MRL→MC) is also provided as an experimental option.

3.7 Comparative Analysis of the Transliteration Process for Cyrillic Script of Different Periods

At this section we present the comparative analysis of transliteration process for historical Romanian Cyrillic scripts of different periods.

Comparing transliteration of 1830–1860 and 1945–1989 Cyrillic scripts we will mention the following important aspects. For letters that are identical in both scripting and are transliterated applying elementary rules, the process is exactly the same. There are some letters (**г**, **к ч**, **ц**) the transliteration rules for which are not so elementary, but are also identical for any Cyrillic scripting.

Transliteration of 1830–1860 Cyrillic script gives, nevertheless, better results than processing of 1945–1989 Cyrillic script. Transliteration of transitional alphabets was successful for 99% of words while for MC this fraction was 91%. TR of 1830–1860 has no problems with letters **ы** and **я**. The rule for **ы** that should be converted to **â** or **î** has some fuzziness, namely, keeping of **î** after prefixes. Transliteration of **я** from MC creates a number of ambiguous situations and strongly depends on the context. The most complicated case is the occurrence of **я** inside words. Three variants are **я→ea**, **я→ia**, **я→a**. We use some heuristically and

statistically motivated rules but most cases imply addressing external dictionaries. In 1830–1860 TR did not provoke such issues because letter **я** was not used at every phonetically suitable situation. RC contains specific letters, for example, **ѣ→еа**; **ѥ→іа**.

4 Conclusion

The proposed technology simultaneously contributes to heritage preservation, simplifies considerably its usage, extends domains and possibilities for research including humanitarian domains and enriches international communication media.

Execution of the planned works will permit unification, homogenization, and integration of national and cultural media in the international information society, and will confirm status of the Romanian language as language of communication in the European continent.

The proposed technology could be used for completion of reusable linguistic resources with new words extracted from digitized texts and attested by linguists-experts. It could also be used in creation of e-learning platforms using these texts as didactic material at learning.

It is possible to apply the described technology for another language.

The technology will automate processing of texts printed in different variants of the Romanian Cyrillic script used in 17th–20th centuries, and give unlimited access to them.

References

[1] http://www.digitisation.eu/community/map-of-the-digitisation-landscape/

[2] Bărbulescu, I. Fonetica alfabetului cirilic în textele române din vécul XVI şi XVII. Bucureşti, 1904.

[3] Картя Молдовей. Сек. XVII – ынч. сек. XX. Едиций векь. Сек. XVII – ынч. сек. XIX. Каталог женерал. Кишинэу: «Штиинца», 1990.

[4] Cartea Moldovei (sec. XVII – înc. sec. XX), Ediţii cu caractere chirilice (sec. XIX – înc. sec. XX). Catalog general. Chisinău: «Ştiinţa», 1992.

[5] Cazimir, Ş. Alfabetul de tranziţie. – Bucureşti: Humanitas, 2006.

[6] Tufiş, D.; Diaconu, L.; Barbu, A.M.; Diaconu, C., Morfologia limbii române, o resursă lingvistică reversibilă şi reutilizabilă. Limbaj şi Tehnologie, Editura Academiei Române, Bucureşti, 1996, pp. 59–65.

[7] Moruz, M.; Iftene, A.; Moruz, A.; Cristea, D. Semi-automatic alignment of old Romanian words using lexicons. In: Proceedings of the 8th International Conference „Linguistic resources and tools for processing ofthe Romanian language", Iași, Editura Universității „A.I. Cuza", 2012, p. 119–125.

[8] Karlsruher Virtueller Katalog. http://www.ubka.unikarlsruhe.de/kvk.html

[9] Corpus Cyrillo-Methodianum Helsingiense. http://www.helsinki.fi/slaavilaiset/ccmh/

[10] Vitas, D.; Krstev, C.; Obradović, I.; Popović, L.; Pavlović-Lažetić, G. An Overview of Resources and Basic Tools for the Processing of Serbian Written Texts. http://poincare.matf.bg.ac.rs/~cvetana/biblio/Solun03MATF.pdf

[11] Pavlov, R.; Bogdanova, G.; Paneva-Marinova, D.; Todorov, T.; Rangochev, K. Digital archive and multimedia library for bulgarian traditional culture and folklore. International Journal "Information Theories and Applications", Vol. **18**, Number 3, 2011, pp. 276–288.

[12] Indermühle, E.; Liwicki, M.; Bunke, H. Recognition of Handwritten Historical Documents: HMM-Adaptation vs. Writer Specific Training. www.dfki.de/~liwicki/pdf/InLiBu08-01.pdf

[13] Корниенко, С.; Айдаров, Ю.; Гагарина, Д.; Черепанов, Ф.; Ясницкий, Л. Программный комплекс для распознавания рукописных и старопечатных текстов. «Информационные Ресурсы России» №1, 2011.

[14] Resurse lingvistice reutilizabile pentru limba română. http://www.math.md/elrr/

[15] Colesnicov, A. The Romanian spelling checker ROMSP: the project overview. Computer Science Journal of Moldova, vol. **3**, nr. 1(7), 1995, pp.40–54.

[16] Boian, E.; Ciubotaru, C.; Cojocaru, S.; Colesnicov, A; Malahov, L; Petic, M. Electronic linguistic resources for historical standard Romanian. The Proceedings of the conference "Linguistic resources and tools for processing the Romanian language", 16–17 May, 2013, Iasi, pp. 35–50.

[17] Boian, E.; Ciubotaru, C.; Cojocaru, S.; Colesnicov, A; Malahov, L. Digitizarea, recunoașterea si conservarea patrimoniului cultural-istoric. Akademos, Nr. 1(32), 2014, pp. 61–68.

[18] Cojocaru, S.; Colesnicov, A.; Malahov, L.; Bumbu, T. Optical Character Recognition Applied to Romanian Printed Texts of the 18th–20th Century. Computer Science Journal of Moldova, v. 24, Nr. 1(70), 2016, p. 106–117. ISSN 1561–4042.

[19] Erjavec, T. MULTEXT-East Version 3: Multilingual Morphosyntactic Specifications, Lexicons and Corpora.In Proc. of the Fourth Intl. Conf. on Language Resources and Evaluation, LREC'2004, ELRA, 2004. http://nl.ijs.si/ME/Vault/CD/docs/mte-d11f/

[20] Simionescu, R. Hybrid POS Tagger. In: Proceedings of "Language Resources and Tools with Industrial Applications", Workshop Eurolan 2011 summerschool, 2011.

S. Cojocaru[1,2], L. Burtseva[1,3], C. Ciubotaru[1,4], A. Colesnicov[1,5], V. Demidova[1,6], L. Malahov[1,7], M. Petic[1,8], T. Bumbu[1,9], Ş. Ungur[1,10]

[1]Institute of Mathematics and Computer Science
5 Academiei str., MD-2028, Chişinău
Republic of Moldova
[2]E-mail: svetlana.cojocaru@math.md
[3]E-mail: luburtseva@gmail.com
[4]E-mail: constantin.ciubotaru@math.md
[5]E-mail: acolesnicov@gmx.com
[6]E-mail: valentina.demidova@math.md
[7]E-mail: lmalahov@gmail.com
[8]E-mail: petic.mircea@gmail.com
[9]E-mail: bumbutudor10@gmail.com
[10]E-mail: ungur.stefan41@gmail.com

Re-engineering of SonaRes Knowledge Base for On-Site Triage Task in Mass Casualty Situations

Svetlana Cojocaru, Constantin Gaindric, Iulian Secrieru, Sergiu Puiu, Olga Popcova

Abstract

Ultrasound-competent physicians are able to use portable ultrasound in mass casualty situations. The obtained information helps emergency crews to make decisions regarding on-site triage, helping in determination of adequate diagnostics in proper time for saving lives of patients.

In the article an approach for re-engineering into 3 levels of knowledge base in domain of the abdominal region ultrasound examination for the case of mass casualty situations is described. Levels 1-2 correspond to casualty's severity state, and level 3 – to pathologies when the free fluid is present in the abdominal cavity, that is not the consequence of an abdominal injury.

Keywords: knowledge base re-engineering, on-site triage, mass casualty, medical ultrasound, hepato-pancreato-biliary region.

1 Introduction

The existence of people and society is increasingly being subjected to serious challenges caused by catastrophes, disasters or natural emergency situations (earthquakes, landslides, floods, etc.), technogeneous, biological social and premeditated (terrorism).

A *disaster* is a natural or man-made event that suddenly or significantly disrupts normal community function and causes concern for

the safety, property and lives of the citizens. Thus, disaster is an event that exceeds the capabilities of the response.

Since 20th century due to technological progress the number of disasters considerable increased, as well as their magnitude. In disaster focus an enormous number of victims died before hospitalization, having injuries compatible with life, because healthcare services did not provide a full qualitative rapid aid.

A *mass casualty incident* is an event that exceeds the health care capabilities of the response, when health care needs additional large resources.

The problem of providing medical aid in the case of a large number of victims was understood in 1881. As a consequence of a fire at the Ring Theater in Vienna more than 400 persons with trauma and burns did not obtained medical aid up in the morning because of lack of overnight medical emergency service.

In case of mass casualty situations, the frequency of which is growing, the number of victims usually exceeds local medical resources. Terrorist attacks of great resonance – attacks on the World Trade Center in New York, bombings in Madrid and London – have resulted in large numbers of victims, comparable to those in military conflicts.

As a result of disasters about 2 million people die annually in the world, more than 200 million suffer trauma of diverse severity, consequently about 10 million people remain disabled. 75-85 % of fatalities occur within first 20 minutes. According to some studies, the number of deaths would be reduced by 30 % if victims are provided medical care timely in an hour after catastrophe [1].

These figures demonstrate the importance of providing in proper time medical assistance on the disaster site.

In the case of emergency situation or disaster, resulting in a significant number of victims in a short period of time many people simultaneously require urgent medical assistance and evacuation from the impact zone. Inevitably, for a period of time there is a strain that medical assistance exceeds currently available medical capabilities and resources. Obviously, in such circumstances medical aiding to the full extent for all affected people is practically impossible, that highlights the importance of emergency diagnostics, triage and setting a schedule for evacuation.

2 Triage in the Disaster Scenarios

Emergency situation is characterized by the complexity of decisions to be made.

Medical triage in case of mass casualty accidents or disasters [2, 3] is a complex process of identification and differentiation of victims in homogeneous groups according to the severity and nature of injuries and the degree of emergency medical assistance. It determines sequence, mode and evacuation destination depending on available medical capabilities and resources, as well as specific circumstances imposed by the impact.

The basic aim of medical triage is ensuring the provision of medical assistance in optimum time and in maximum possible volume to the largest number (ideally – to all) of victims of the disaster. In extreme cases diagnostics requires from physician a strategy that reduces to sorting victims into several groups in order to ensure targeted aid, taking into account the severity of the case.

Priority 1 – *Absolute emergency*. Victims with serious and very serious injuries, illnesses, intoxication or contamination, compromising vital functions, which require immediate stabilization measures, as well as priority evacuation in assisted medical transport conditions.

Priority 2 – *Relative emergency*. Victims with serious or moderate injuries, illnesses, intoxication or contamination, with retained vital functions, but with the risk of developing life-threatening complications immediately ahead. They require urgent medical assistance, but not the immediate one.

Priority 3 – *Low urgency*. Victims with minor injuries, illnesses, intoxication or contamination, no life-threatening, which can be treated later, usually in outpatient conditions. They can be evacuated in non-specialized transport or independently.

In case of mass casualty situations examination should be made on the site, in reduced time in order to determine the right strategy for saving. This specific character requires an approach, different from the traditional clinical examination. Structure of trauma and injuries varies essentially depending on the disaster's nature.

Ultrasound diagnostics at the disaster site is aimed at determining the level of urgency to save lives and to prevent any complications for people at risk.

Portable ultrasound has clear advantages over other imaging equipment (particularly, computed tomography) to be applied in remote places. Since ultrasound is painless and safe technique that captures images in real-time (showing the structure and movement of the body's internal organs and blood vessels) and portable light-weight ultrasound scanners can be used easily at the accident site, this method has been accepted as an important initial screening tool in disaster medicine.

FAST (Focused Assessment with Sonography for Trauma) examination, a well-known rapid bedside ultrasound examination used in emergency medicine, was grown widespread in the early 1990s.

"The FAST examination is based on the assumption that the majority of clinically significant abdominal injuries result in hemoperitoneum. The standard FAST protocol is directed to detection of fluid in the pericardial and peritoneal spaces. With regard to the CAVEAT protocol… is limited to… intraperitoneal hemorrhage" [4]. CAVEAT is the concept of a comprehensive sonographic examination in the evaluation of chest, abdomen, vena cava, and extremities in acute triage.

As discovery of free fluid in the abdomen can lead to appropriate and timely diagnostics, FAST can be used to guide clinical decision-making, e.g. as a quick method for triaging patients.

Rozycki et al. [5] have found that ultrasound was the most sensitive and specific in patients with penetrating chest wounds or in hypotensive blunt abdominal trauma patients (sensitivity and specificity nearly 100 %).

In [6] ultrasound was performed by relief teams after the 1988 Armenian earthquake as a primary screening procedure in 400 of 750 injured multiple casualty incident (MCI) patients admitted to a large hospital within 72 h of the event. The average time spent on evaluation of a single patient was approximately 4 min. Traumatic injuries of the abdomen were detected in 12.8 % of the patients.

In [7] sonography was used after an MCI in Guatemala in which the dead far outnumbered the injured. In that setting, ultrasound was useful for excluding internal trauma.

Ultrasound has been utilized in military deployments in Kosovo, Afghanistan and Iraq [8]. The British Air Assault Surgical Groups deployed to Kuwait during 2003 included the use of a hand-held ultrasound scanner by the forward medical units FAST examinations performed by trained emergency physicians using portable equipment at a large military combat hospital in Iraq. It had very high sensitivity as confirmed by subsequent operative reports and computed tomography imaging. In that particular experience ultrasound was performed in patients who sustained blunt, blast and penetrating trauma [8].

Statistics shows that in cases of natural disasters, catastrophes and accidents about 70 % of affected persons need specific healthcare approach limited in time.

So, it is important to offer recommendations on the evacuation priority and creation groups for evacuation from the disaster focus, according to destination (specialized centers or general profile medical institutions) based both on triage results and limited possibilities for transportation in case of a large number of victims.

The main limitation of the FAST examination is that the operator must be knowledgeable in its clinical use and be aware that *it does not exclude all injuries* [9].

The limitations of FAST (as a task-focused approach) are caused by the narrow view in emergency situation formalization. In fact, the plausible reason of patient critical state can lie outside the emergency. For instance, the potential false-positive diagnosis of free traumatic fluid in the peritoneum may be due to fluid present in patients for physiologic reasons, including ovarian cyst rupture, as well as pathologic reasons, such as patients with ascites or inflammatory processes in the abdomen or pelvis [9].

In this paper we describe the algorithm that would allow physician rescue team in reduced time to appreciate diagnosis, severity, the life-threatening in order to make appropriate decisions about how to help, treat and evacuate from the disaster site.

In distinction from FAST, in our approach we consider emergency situation as a particular case of ultrasound examination domain formalization and take into account the injury severity.

3 Approach Based on SonaRes Knowledge Base

Abdomen region is the important one, as the most difficult cases to diagnose with extremely dangerous consequences are lesions of the abdominal cavity organs (liver, spleen, kidneys, large vessels of the abdomen, gallbladder and pancreas).

The paper authors over several years elaborated SonaRes technological platform, designed for development of medical informatics applications to support diagnostic process based on ultrasound examination method [10-11].

SonaRes technological platform consists of two main parts – SonaRes methodology and SonaRes technology. SonaRes methodology consists of knowledge acquisition strategy, knowledge representation and storage form, inference mechanism. SonaRes technology offers knowledge base editor and tools that allows creation of different destination information systems.

The main part of SonaRes technological platform represents SonaRes knowledge base, which includes the following formalized expert knowledge:

- 335 facts and 54 decision rules for gallbladder,
- 231 facts and 52 decision rules for pancreas,
- 167 facts and 31 decision rules for liver,
- 257 facts and 15 decision rules for bile ducts.

Based on facts, the decisions rules describe organs pathologies and anomalies.

Our approach presumes to re-engineer SonaRes knowledge base for on-site triage task in mass casualty situations, performing the following steps:

1. to add to the knowledge base information (facts and decision rules) that describes blood vessels;
2. to identify conclusions (decision rules) that describe fluid presence, obtaining *knowledge base – critical level 1*;
3. to localize the obtained conclusions into 4 areas of the FAST examination;
4. to identify information (facts and decision rules) that allows to make severity assessment (fluid volume and patient state severity), obtaining *knowledge base – emergency level 2*;

5. to identify conclusions (pathologies and anomalies) that describe presence of free fluid in the abdominal cavity, which is not the consequence of an abdominal injury, obtaining *knowledge base – non-injury level 3*;
6. to validate completeness of all 3 levels of the knowledge base for emergency situation;
7. to reorder the set of facts according to their information value in order to minimize the number of inference steps;
8. to classify conclusions from levels 1-2 in priority groups (absolute emergency, relative emergency, low urgency).

In this way we re-engineer SonaRes knowledge base for on-site triage task and overcome limitations of the FAST examination, taking into account physiologic and pathologic reasons.

4 Conclusion and Future Work

The current consensus supports ultrasound screening of mass casualties for evaluating trauma patients. In particular, FAST examination is used to identify presence of intraperitoneal or pericardial free fluid, presumed to be consequences of bleeding. It is important to note, however, that the FAST examination is a screening test, and false-negative conclusions do occur.

We propose a methodology of re-engineering of SonaRes knowledge base for on-site triage task in mass casualty situations.

We obtained a modified knowledge base oriented for emergency (mass casualty) situations, when injuries need immediate surgical intervention:

- *critical level 1* – corresponds to fluid presence
- *emergency level 2* – corresponds to severity assessment
- *non-injury level 3* – corresponds to presence of free fluid due to physiologic and pathologic reasons.

Our approach allows to differentiate process of on-site triage depending on time available for decision-making.

In cases when there is a need and the conditions allow (for instance, during transportation) to repeat examination, our approach, in distinction from FAST, provides more competent assistance, evaluating state severity assessment.

The following work could be done in the future:

- a scoring system to be added in priority groups (absolute emergency, relative emergency, low urgency), as it is usual for physicians;
- based on the re-engineered knowledge base, an algorithm to be created and validated on virtual scenarios.

In addition, there is a well suited provision of emergency physicians and rescue teams with a decision support system to assist emergency examination, helping in establishing the correct diagnostics in opportune terms. For example, it is possible to use SonaRes technological platform [10] that already exists and was tested in creation of a system that uses portable scanners and is aimed for diagnostics under field conditions, accessible through easy of use under mass casualty conditions, lack of time and qualified medical personnel.

References

[1] *Медицина неотложных состояний. Избранные клинические лекции.* Том 1. Под редакцией проф. В. В. Никонова, доц. А. Э. Феськова, Изд. 3-е. – Донецк: Издатель Заславский А. Ю., 2008.

[2] J. L. Jenkins, M. L. McCarthy, L. M. Sauer, G. B. Green, S. Stuart, T. L. Thomas, E. B. Hsu. *Mass-casualty triage: Time for an evidence-based approach.* Prehospital Disaster Medicine, 2008.

[3] *Disaster Medicine.* Second edition, Editors D. E. Hogan, J. L. Burstein, Lippincott Williams&Wilkins, a Wolters Kluwer business, 2007.

[4] S. P. Stawicki, J. M. Howard, J. P. Pryor, D. P. Bahner, M. L. Whitmill, A. J. Dean. *Portable ultrasonography in mass casualty incidents: The CAVEAT examination.* World J Orthop. 2010 Nov 18; 1(1): 10-19. Published online 2010 Nov 18. doi: 10.5312/wjo.v1.i1.10 PMCID: PMC3302028.

[5] G. S. Rozycki, R. B. Ballard, D. V. Feliciano, J. A. Schmidt, S. D. Pennington. *Surgeon-performed ultrasound for the assessment of truncal injuries: lessons learned from 1540 patients*. Ann Surg. (1998) 228: 557-567.

[6] A. E. Sarkisian, R. A. Khondkarian, N. M. Amirbekian, N. B. Bagdasarian, R. L. Khojayan, Y. T. Oganesian. *Sonographic screening of mass casualties for abdominal and renal injuries following the 1988 Armenian earthquake*. J Trauma (1991) 31: 247-250.

[7] A. J. Dean, B. S. Ku, E. M. Zeserson. *The utility of handheld ultrasound in an austere medical setting in Guatemala after a natural disaster.* Am J Disaster Med. (2007) 2: 249-256.

[8] T. E. Kolkebeck, S. Mehta. *The focused assessment of sonography for trauma (FAST) exam in a forward-deployed combat emergency department: a prospective observational study*. Ann Emerg Med. (2006) 48: S87.

[9] AIUM Practice Parameter for the Performance of the Focused Assessment With Sonography for Trauma (FAST) Examination, American Institute of Ultrasound in Medicine, 2014. www.aium.org/resources/guidelines/fast.pdf

[10] L. Burtseva, S. Cojocaru, C. Gaindric, O. Popcova, Iu. Secrieru. *Ultrasound diagnostics system SonaRes: structure and investigation process.* Second International Conference "Modelling and Development of Intelligent Systems", September 29 – October 02, 2011, Sibiu, Romania, Lucian Blaga University Press (2011), pp. 28-35.

[11] S. Cojocaru, C. Gaindric, O. Popcova, Iu. Secrieru. *SonaRes Platform for Development of Medical Informatics Applications.* Proceedings of the 3rd International Conference on Nanotechnologies and Biomedical Engineering ICNBME-2015, September 23-26, 2015, Chisinau, Moldova, Springer, IFMBE Proceedings, vol. 55, pp. 450-453.

Svetlana Cojocaru[1], Constantin Gaindric[2], Iulian Secrieru[3], Sergiu Puiu[4], Olga Popcova[5]

[1]Affiliation/Institution: Institute of Mathematics and Computer Science of the Academy of Sciences of Moldova
E-mail: svetlana.cojocaru@math.md

[2]Affiliation/Institution: Institute of Mathematics and Computer Science of the Academy of Sciences of Moldova
E-mail: gaindric@math.md

[3]Affiliation/Institution: Institute of Mathematics and Computer Science of the Academy of Sciences of Moldova
E-mail: secrieru@math.md

[4]Affiliation/Institution: Women's Health Center "Dalila"
E-mail: puiusv@yahoo.com

[5]Affiliation/Institution: Institute of Mathematics and Computer Science of the Academy of Sciences of Moldova
E-mail: oleapopcova@yahoo.com

Proceedings of the Conference on Mathematical Foundations of Informatics
MFOI2016, July 25-29, 2016, Chisinau, Republic of Moldova

Reliability of Information-Computer Systems with Hierarchical Structure

Andrei Corlat

Abstract

A mathematical model of information-computer systems with hierarchical structure is built in general assumptions concerning the distributions of failure and repair times of the systems unit.

Keywords: systems with hierarchical structure, reliability, semi-Markov processes.

1 Introduction

The Information-Computer Systems have in main a specific structure: the information from some principal control unit is forthcoming to several next units, each of that in its turn communicates the information to the following units, etc. This structure takes place in systems in which the circular transmission of information signals (or management signals) or collection of information from downstairs elements are carried out.

2 Systems with Hierarchical Structure

We will consider the system S with hierarchical structure: the principal unit a_0 is connected with a_1 units of the first level, each of that is connected with a_2 units of the second level, etc. The units of the last level n-level are called extreme elements, their number is $N = a_1 \cdot a_2 \cdot ... \cdot a_n$ and they form $K = a_1 \cdot a_2 \cdot ... \cdot a_{n-1}$ groups.

The system's unit may be unable to operate either it is failed, or it is disconnected as a result of failure of some unit. The failure of (i, j) unit (i indicates the level, $i = \overline{0, n}$, j – the number of the unit of i-level, $j = \overline{1, N_i}$, $N_i = a_1 \cdot a_2 \cdot ... \cdot a_i$) leads to the disconnection of all units that are connected with this unit and are controlled by it and of all preceding units connected with it and that do not belong to any efficient way. We will understand here under an efficient way a chain of functional connected operating units from the principal $(0, 1)$ to one of the extreme.

The restored unit is included in system simultaneously with all previously disconnected operative units (with that level of efficiency at the moment when these units are disconnected) that form an efficient way with the restored unit. Moreover, the disconnected early units under repair continue (but not start again) their repair, if these units are functionally connected with the restored unit.

The system is considered in failure (total failure) if the number of efficient ways is less than R $(1 \leq R < N)$ and at this moment all operative units are disconnected.

It is assumed that

- the failure times $\alpha_1^{(ij)}$ and the repair times $\alpha_0^{(ij)}$ are independent in totality random variables with limited mean $0 < E\alpha_k^{(ij)} = T_k^{(ij)} < \infty$, $i = \overline{0, n}$, $j = \overline{1, N_i}$, $k = \overline{0, 1}$,

- the distribution functions of failure and repair times are considered absolutly continuous with respect to Lebesgue measure, $\overline{F}_k^{(ij)}(t) = 1 - F_k^{(ij)}(t) > 0$, $t > 0$,

- the restored unit is as good as new,

- there is no queue to repair,

- the disconnection and including of the units in system, as well as failure, are taking place instantaneously.

3 Semi-Markov Model

The functioning of such system is described (see [1]) by the semi-Markov processes $\xi(t) = \{\xi_{01}(t), \xi_{11}(t), \xi_{12}(t), ..., \xi_{21}(t), ..., \xi_{ij}(t), ..., \xi_{nN}(t); v_{01}(t), v_{11}(t), v_{12}(t), ..., v_{21}(t), ..., v_{ij}(t), ..., v_{nN}(t)\}$, where

$$\xi_{ij}(t) = \begin{cases} 1, & \text{if the } (i,j) \text{ unit is operative at the moment } t, \\ 0, & \text{if the } (i,j) \text{ unit is under repair at the moment } t. \end{cases}$$

$v_{ij}(t)$ – is (if $\xi_{ij}(t) = 0$) the repair time of (i,j) unit from its last failure and (if $\xi_{ij}(t) = 1$) the lifetime of (i,j) unit from its last join in the system (without taking in consideration the possible time of disconnection).

The phase space of system's states is $(Z, \mathcal{Z})$, where $Z = \{(d; x^{(ij)}) : d \in D, x^{(ij)} = (x_{01}, x_{11}, ..., x_{1N_1}, x_{21}, ..., x_{2N_2}, x_{i1}, ..., x_{ij-1}, 0, x_{ij+1}, ..., x_{nN}), x_{km} > 0, k = \overline{0,n}, m = \overline{1,N_k}, (k,m) \neq (i,j)\}$;
$D = \{d : d = (d_{01}, d_{11}, ..., d_{1N_1}, d_{21}, ..., d_{2N_2}, ..., d_{km}, ..., d_{nN}), d_{km} = \overline{0,1}, k = \overline{0,n}, m = \overline{1,N_k}\}$
x_{km} – points the time passed by the last change of "physical" state of the (k,m) unit, d_{km} – describes the "physical" state of the (k,m) unit:

$$d_{km} = \begin{cases} 1, & \text{if it is operative (or disconnected in an operative state),} \\ 0, & \text{if it is under repair (or disconnected in an failured state),} \end{cases}$$

$\mathcal{Z}$ – σ-algebra of Borel sets in Z.

We define the set of operative system's states Z_1 and the set of failure system's states Z_0 proceeding from the concept of total system's failure: $Z_1 = \{(d; x^{(ij)}) \in Z : d \in D_1\}$, $Z_0 = \{(d; x^{(ij)}) \in Z : d \in D_0\}$, where

$$D_1 = \{d \in D : \sum_{u=1}^{N} S_u \geq R\}, \; D_0 = \{d \in D : \sum_{u=1}^{N} S_u < R\},$$

$$S_u = d_{nu} \cdot d_{n-1,u-1} \cdot ... \cdot d_{i,u_1} \cdot ... \cdot d_{1,u_1} \cdot d_{0,1},$$

$$u_i = \begin{cases} \left[\frac{u}{a_n \cdot a_{n-1} \cdot \ldots \cdot a_{i+1}}\right] + 1, & \text{if } u \neq 0 (mod(a_n \cdot a_{n-1} \cdot \ldots \cdot a_i)), \\ \left[\frac{u}{a_n \cdot a_{n-1} \cdot \ldots \cdot a_{i+1}}\right], & \text{otherwise,} \end{cases}$$

where $[\cdot]$ denotes entire part of the number.

The mean life time T_1 of the system S is given by

$$T_1 = \left\{ \sum_{d \in D_1} \prod_{i=0}^{n} \prod_{j=1}^{N_i} T_{d_{ij}}^{(ij)} \right\} \cdot \left\{ \sum_{d \in D_0} \sum_{(i,j) \in I} \prod_{s=0}^{n} \prod_{\substack{v=1 \\ (s,v) \neq (i,j)}}^{N_s} T_{d_{sv}}^{(sv)} \right\}^{-1},$$

and the mean repair time T_0 of the system S is given by

$$T_0 = \left\{ \sum_{d \in D_0} \prod_{i=0}^{n} \prod_{j=1}^{N_i} T_{d_{ij}}^{(ij)} \right\} \cdot \left\{ \sum_{d \in D_0} \sum_{(i,j) \in I} \prod_{s=0}^{n} \prod_{\substack{v=1 \\ (s,v) \neq (i,j)}}^{N_s} T_{d_{sv}}^{(sv)} \right\}^{-1},$$

where I denotes the set of units under repair that are not disconnected at the state d.

4 Homogeneous Systems

Suppose now that the system S is homogeneous: the units of the i-level are of the same type $T_k^{(ij)} = T_k^{(i)}$, $k = \overline{0,1}$, $i = \overline{0,n}$. The system will be considered under total failure when the number of operative extreme groups is less the P $(1 \leq P < K)$. An extreme group is operative, if it contains Q or more operative units from a_n.

Then may be suggested an iterative algorithm for determining T_1 and T_0. For example, when $P = 1$

$$T_1 = \frac{T_1^{(0)} S_+(1)}{T_1^{(0)} S_*(1) + S_+(1)},$$

$$T_0 = \frac{T_0^{(0)} S_+(1) + T_1^{(0)} S_-(1)}{T_1^{(0)} S_*(1) + S_+(1)},$$

where

$$S_+(n-i) = \left[T_1^{(n-i)} S_+(n-i+1) + A_{n-i}\right]^{a_{n-i}} - S_-(n-i), i = \overline{1, n-1},$$

$$A_{n-i} = T_0^{(n-i)} S_+(n-i+1) + T_1^{(n-i)} S_-(n-i+1), i = \overline{1, n-1},$$

$$S_-(n-i) = [A_{n-1}]^{a_{n-i}}, i = \overline{1, n-1},$$

$$S_*(n-i) = a_{n-i}\left[S_+(n-i) + T_1^{(n-i)} S_*(n-i+1)\right] A_{n-i}^{a_{n-i}-1}, i = \overline{1, n-1},$$

$$S_+(n) = \left(T_1^{(n)} + T_0^{(n)}\right)^{a_n} - \sum_{k=a_n-Q+1}^{a_n} C_{a_n}^k \left(T_1^{(n)}\right)^{a_n-k} \left(T_0^{(n)}\right)^k;$$

$$S_-(n) = C_{a_n}^{a_n-Q+1} \left(T_1^{(n)}\right)^{Q-1} \left(T_0^{(n)}\right)^{a_n-Q+1};$$

$$S_*(n) = C_{a_n}^{a_n-Q+1} \left(T_1^{(n)}\right)^{Q-1} \left(T_0^{(n)}\right)^{a_n-Q}.$$

5 Conclusion

The main characteristics of the reliability of information-computer systems with hierarchical structure are obtained.

It should be mentioned that the results are obtained in terms of structure and means of failure and repair times and in a suitable for coding form.

References

[1] A. Corlat, V. Kuznetsov, M. Novicov, A. Turbin. *Semi-Markov models of systems with repair and queueing systems.* Shtiintsa, Kishinev, 1991 (in Russian).

Andrei Corlat,

Affiliation/Institution: Institute of Mathematics and Computer Science of the Academy of Sciences of Moldova
Email: andrei.corlat@math.md

Proceedings of the Conference on Mathematical Foundations of Informatics
MFOI2016, July 25-29, 2016, Chisinau, Republic of Moldova

An Extensional Model of Natural Languages

Ioachim Drugus

Abstract

Based on an algebraic set theory called "aggregation theory" developed by the author, a set-theoretic, i.e. an "extensional", model of natural languages is here presented. The syntax is immersed into the semantics by treating the syntactic objects as carriers of "formal sense" – the kind of meaning, which these objects have before they are interpreted to obtain a meaning. Due to this, this model is expected to reflect both the syntax and the semantics of natural languages. Associated with this model is a "nominalistic interpretation" of set theory, where the sets are treated as common names of their elements and, in particular, the singletons are treated as proper names.

Keywords: algebraic set theory, metalingua, semantics.

1 Introduction

To regard everything as populating the universe of discourse of set theory with atoms, is an approach often referenced as "universalist view". Since an "atom" is treated as a "non-set", anything is either an atom or a set. Therefore, the universalist view considering the universe of discourse as comprising everything does not lack logical grounds. This view can raise foundational concerns due to its "maximalism", but such concerns can be taken care of by an appropriate axiomatics and by considering many smaller universes of discourse rather than one universe. A discourse in natural language is also limited to a certain domain and such a domain changes with changing the object of discussion. Thus, this view has a value for the philosophy of mathematics. This value is

transposed here into a value for the philosophy of language. Namely, the universalist view is regarded here as grounds for the belief that there exists an axiomatic set theory with atoms, which can serve as a framework for modeling natural languages. Another value of this view is that it brings to foreground the set theory with atoms, which is useful for applications of mathematics in practice, versus the "pure" set theory, which treats any objects as sets built from the empty set, which is useful for mathematics.

To regard *anything* as residing in the universe of discourse of set theory with atoms presupposes that this universe comprises entirely the physical World. In this universe of discourse, the material things are set-theoretic atoms, and the sets are making up a "superstructure" over the World (see [11] for a mereological set theory where the universe of discourse of a "pure" set theory, i.e. one without atoms, is treated as a superstructure over the "World" treated mereologically). Thus, it appears that both the natural language and the language of set theory are used to discuss about the same thing – the World. Therefore, one should look for an appropriate language of set theory with atoms which, together with its interpretation, would serve as a model of natural languages, and the features of such a set theory are described next.

Since in a discussion one limits to a certain domain to be treated as a universe of discourse, the set theory looked for must be extremely weak to be applicable to all domains. It must only make distinction between a set and an atom and be able to express the essence of what is a set. Here, this essence is expressed like this: "A set is a *minimal structure*, built by application of operations". Due to the minimality, no other means of building a set are presupposed, and a set, which is not built in this manner, but is defined by comprehension principle (the principle called "separation axiom" in ZF, according to which, the set denoted as $\{x\ \epsilon X \mid \phi(x)\}$ exists), might not be a "proper set"; in mainstream set theories, this can be a "proper class". Since such a set theory is based on operations rather than on a relation (the relation "ϵ"), the set theory one would look for must be *algebraic*. Because no comprehension principle in any form is presupposed, there should exist a means to build

arbitrary sets, and here this is a "generalized induction" principle. The "induction" is treated here as an explication in set-theoretic terms of the intuitive notions of "building" or "construction".

The word "aggregation" is used here as a "generic term" for operations used in building sets. This term was chosen, because it is a cognate of the term "aggregate", used both in the first translation to English of the Cantor's "Mengenlehre" ("Set Theory"), and in Principia Mathematicae – as a generic term for any explication of the conception "set". But here, the term "aggregate" cannot be used, because the aggregation theory is also about atoms. Even when an object is known to be a "set", it makes little sense to call it "aggregate" to comply with the name of the theory. The term "multitude" of [1] is also not the best due to atoms. The term used here for the entities populating the universe of discourse is "object". But, the term "aggregation" sounds appropriate for any operation used to build a set. In development of the set theory needed for modeling natural languages, one should start from looking for a minimal number of aggregation operations.

An algebraic set theory called "aggregation theory" was presented in [4] – a theory which, in opinion of the author, can play the role of a framework for modeling natural languages. The aggregation theory is pivoted around a set-theoretic operation called "adduction" (whereas, the operation symmetrical to it is called "adjunction"). This operation is regarded as playing the same role in semantics as that of the space between words in natural languages. Thus, a text in a natural language is treated here as an expression in the language of aggregation theory, which is built by application of the adduction or adjunction operations. Both these operations are needed for a natural language using linear texts, where one cannot avoid both "direct" and "inverse" operations.

The idea of this extensional model of natural languages was first presented in [5] in connection with a symbolism called "Punctuation Markup Language" (PML) proposed for disambiguation of texts in natural language. With a larger focus on its use in natural languages, this idea was presented in [6], to appear. In current paper, this model

is presented with the axiomatics of aggregation theory of [4] to parallel the language of aggregation theory and natural languages. Also, the "nominalistic interpretation" of set theory, developed in some detail in [5] and [6], is presented here – an interpretation, according to which a set is a common name of its elements, with singletons playing the role of proper names – phenomena specific to natural languages.

The aggregation theory presented next is an algebraic counterpart of a set theory weaker than the one presented in [1], since various existence axioms (like the infinity axiom) of [1] are not postulated here.

2 A short introduction to aggregation theory

The binary operation which, applied to two sets x and y, produces the set $\{x\}\cup y$, is called here "adduction" and is denoted as "$x : y$". Notice, that the meaning in Latin of "adduction" is "to bring (in)to", and this complies with the intuitive treatment of "$x : y$" as "bring x (in)to y". This denotation was selected to be suggestive of the historical denotation $\{x : \phi(x)\}$ of a class defined via a property $\phi(x)$ of its elements. In modern use, the colon is customarily replaced by a vertical bar "|", which is a stylized colon. The languages of mainstream set theories use exclusively the symbol "ϵ" of membership, and this denotation of a class is a *metamathematical* symbol for a *complex* aggregation operation. The following fact is regarded here as enabling the adduction to serve as a fundamental operation for an algebraic set theory:

$$x : y = y \leftrightarrow \ x \in y. \quad (*)$$

The notation $\{x : \phi(x)\}$ of a class, which appears due to comprehension principle in class theory, is treated here as reflecting, even though in a latent manner, the application of two operations – the adduction operation applied first, and the unary operation $\{.\}$ denoted by the two braces applied second (and last); this operation is called here "individuation" for reasons to be discussed later. Thus, the notation $\{x : \phi(x)\}$ is treated here as denotation of a complex aggregation operation, an operation for building sets, which is a superposition of two

operations. More details on this will be given later, when this will be treated below from semantic viewpoint. Here, it is worthwhile noticing that $\{x : \phi(x)\}$ is a metalinguistic notation, and thus, the other two notations should be also treated as metalinguistic. Therefore, the language of such metalinguistic notations is called "metalingua" and is denoted as μL. Metalingua is the language of the aggregation theory. The notations of the two aggregation operations, adduction and individuation, identified above, are part of the metalingua.

The empty set denoted here as "0" can be treated as obtained by application of a 0-ary aggregation operation and the symbol "0" can be considered as a symbol of metalingua. In this case, the operation of formation of a singleton $\{x\}$ is definable through the other operations like this: $\{x\} = x : 0$. But a set theory with atoms and without an empty set may be also interesting, and such a theory makes a lot of sense for semantics. Therefore, one can alternatively axiomatize the aggregation theory with or without an empty set. Also, metalingua should have a predicate symbol in order to formulate statements, and here this is the symbol "=" of equality to be interpreted as an equivalence relation.

The axioms of the aggregation theory in the language of the symbols ":" and "=" are the universal closures of the formulas below:

$$(x : x) : z = x : z, \tag{1}$$

$$(x : y) : z = (y : x) : z, \tag{2}$$

$$x \in (y : z) \leftrightarrow x = y \vee x \in z \tag{3}$$

These identities can be referenced as "left idempotency", "left commutativity" and "adduction axiom", respectively. The (3) can be also called "Bernays law", since as early as in 1909, in another form, it was stated by Bernays in [2]. The expressions of the form "$X \in Y$" used in (3) are treated as short for longer expressions like this: "$X : Y = Y$".

This theory is considered to be the "*minimal* aggregation theory", because these axioms are true assertions about individuals of any mainstream set theory (see also [10] where these axioms in another form, among others, are used for axiomatizing hereditarily finite set theory). When this theory is extended, the objects described by the "larger"

theories satisfy these axioms. In other words, these three axioms are considered to "express the essence" of what is a set.

The axiom of empty set is not postulated in the minimal aggregation theory. This axiom is the universal closure of the formula $x : 0 \neq 0$, and cannot be regarded as expressing the essence of sets, because it also holds for urelements: if u is a urelement, then $x : u \neq u$. In this context, recall that a "Quine atom" is an object, which satisfies the condition $x = \{x\}$. This is an object for which the membership is "reflexive". Notice, that in aggregation theory, a Quine atom is an object which satisfies the condition $x : x = x$, i.e. this is "idempotent" in the algebra of aggregation theory.

The fact that a set S has the elements $s_1, ..., s_n$ can be expressed by the following correlation (**) generalizing the correlation (*):

$$(s_1 : (s_2 : ... : (s_n : S)...)) = S. \quad (**)$$

This correlation suggests introducing the "right association rule", i.e. the rule which allows to drop the parentheses in expressions like the left-hand expression of (**). Also, due to axiom (2), all the colons in such an expression except the last one can be replaced by commas, so the expression would look like this: "$s_1,\ s_2, ..., s_n : S$". At first sight, one would consider "$s_1,\ s_2, ..., s_n : S$" as another denotation of a finite set $S = \{s_1,\ s_2, ..., s_n\}$, but this is correct only if the correlation (**) holds, and when it does, S can be treated as the *name* of the set $\{s_1,\ s_2, ..., s_n\}$. Thus, in this theory, the *names of sets reside in the universe of discourse*, and this is the first manifestation of the "nominalistic interpretation" of set theory, which will be discussed later.

Notice, that the "left association rule" also makes sense. In mainstream set theories, natural numbers are represented by finite ordinals, but Zermelo defined them differently – he treated a natural number n as the set $\{...\{0\}...\}$, where the number of nested parentheses is n. In notation of aggregation theory, the expression "$0 : 0 : ... : 0$" with $n + 1$ of 0-s separated by colon, where the parentheses associated to the left are dropped, is exactly the Zermelo's number n. One can say that the "left associated" expressions reflect the "size", and the "right

associated" ones reflect the cardinality. True, this intuition works only for the finite sets (actually, for hereditarily finite sets). Probably, a work on extending this idea to the "transfinite" can be done similarly to how this was done by von Neumann for ordinals.

In minimal aggregation theory, the extensionality axiom is not postulated, but it can be added to obtain an extended theory. In minimal aggregation theory, the following definition comes up as natural:

$$\varepsilon(x) = \{u \in U \mid u \in x\}, \tag{4}$$

where U is the universe of discourse. This is a meta-definition, i.e. a definition in the metalanguage, of a meta-term denoting a class called "*extension* of x". Using this terminology, one can formulate the regular extensionality axiom like this: any two objects x and y coincide, iff their extensions coincide. The following statement expresses the extensionality axiom in metalanguage:

$$\varepsilon(x) = \varepsilon(y) \to x = y \qquad (***)$$

In the minimal aggregation theory, neither the extensionality axiom, nor this statement holds, and the symbol $\varepsilon(x)$ has an intuitive meaning – it is a meta-name of a collection, also "named" as x. Here, x is an object described by the theory, i.e. one residing in its universe of discourse, and we will refer to such names as "identities", or "IDs" (from the word "identifier"). Notice, that here the word "identity" is used with the same meaning as in acronym "URI" ("Universal Resource Identifiers) and not like in the expression "algebraic identity". Thus, one can say, that the minimal elementary aggregation theory can be regarded as a "multi-identity set theory". Should such a theory be regarded as a "multi-set theory" can be a matter of large polemics. Notice, that in current presentation the notion of "copy" of set is not invoked, and the notion of "identity" is employed. Thus, the objects described by the theory introduced here are *not* regarded as having "copies" – they are considered as existing in one "copy", but to have many "identities". Here the word "identity" is treated as a "unique name", i.e. "a name selected from many synonyms".

The focus on "identities" or "IDs" suggests using the denotation "ID" of the minimal aggregation theory. Another reason for this denotation is that in this theory, induction and deduction, with initials suggesting the acronym "ID", are treated as two key dual notions, but this topic is not covered in current paper. An extension of ID theory can be denoted as "ID+X", which is similar to the denotations like "ZF+CH", where "CH" stands for "Continuum Hypothesis axiom". Accordingly, the extension of ID to a theory with empty set is denoted as "ID + 0", and the extension of ID with the extensionality axiom is denoted as "ID + Eq". Thus, similar to how the ZF theory is actually a framework of theories based on one explication of the conception "set", the ID can be treated as a framework of theories based on an explication of this conception via the axioms (1), (2), (3).

The extension of ID with a generalized form of transfinite induction called "reduction principle" is presented in due detail in [4] to make aggregation theory "competitive in strength" with ZF, even though weaker. In that theory, various usual set-theoretic operations are defined by the "reduction principle". The reduction principle is important for foundations, but it is not same important for the topic discussed here, and it will not be also presented here. Thus, going forward, the term "aggregation theory" is treated as the "minimal aggregation theory" with the axioms (1), (2) and (3).

3 The nominalistic interpretation of set theory

The objects in the universe of discourse of the aggregation theory can be treated as *names* as this was mentioned in previous section, and this leads towards a "nominalistic interpretation" of set theory. To go into further detail, the sets about which is the discourse of any set theory are treated as *common names* – any set is regarded as a name of all its elements, and as a particular case, a singleton is treated as a *proper name* of its single element. Also, a Quine atom is regarded here as a name of itself, and a urelement as a name which "names" nothing, a name like "unicorn". Thus, this interpretation in a natural manner

generates an ontology of objects-names.

The nominalistic interpretation comes along prominently in aggregation theory due to multiple identities, but it also makes sense in any set theory with extensionality axiom, i.e. in set theories wider than ID+Eq, like the ZF theory. In such "mainstream" set theories, a set is to be regarded as a *one-identity aggregate*.

The main intuition against nominalistic interpretation, one due to which this treatment was not considered earlier, probably, is that admitting a set to have an identity is not compatible with extensionality axiom. To change this intuition one should proceed from a mereological view on set theory, according to which the identity of a set is in relation "*part of*", and not in relation "element of", with a set. This intuition is compatible with extensionality, but the acceptance that there exists an "extra" entity called "identity" does not comply with Occam's parsimony law for a theory. On the other hand, considering such entities, one can develop a "structuralist" set theory.

4 Metalingua - a formal language for extensional model and nominalistic interpretation

Metalingua was first informally introduced in [7] and a presentation in detail of it as a formal language denoted as "μL" will come in [6]. Here, the essentials of μL relevant to the ideas of current paper are presented following the presentation of [6], but also new details which appeared due to the development of aggregation theory in [4] are added. What is essentially new is the "nominalistic interpretation" provided by aggregation theory. Next, several basic notions related to languages are presented and their treatment in terms of aggregation theory is indicated in *italic* font.

4.1 Basic notions related to languages

In [8], Frege introduced two attributes of a sign, which he called "sense" and "reference", and preferred the term "name" for a sign used in

language. Frege discussed about "names" as "proper names" – things, which are said to "refer to" (or to "denote") an existing or non-existing thing (like unicorn), frequently also called "denotatum". Frege treated names as having "sense" or "meaning", treated by him as synonyms, as depending on the language in which a name is used. In [13], Russell used the term "denoting phrase" for both common and proper names, but he rejected the attribute "sense", regarding it as useless and even creating contradictions. Carnap replaced the notions introduced by Frege and Russell with the notions "intension" and "extension", which he regarded as fundamental for semantics [3]. "Extension of a name" is a notion going back to middle age philosophers, but "intension" is a term which does not seem to have obtained a rigorous treatment, even though lately it has found good use in informatics [12].

To properly subsume the terms used by the three founders of semantics, one has first to notice that the terms "sense" and "meaning" are not exact synonyms. Notice, that an expression is said "to make sense", but "to have a meaning". Accordingly, when precision in using the two terms is important, here "to make sense" is treated as "to be interpretable", but "to have a meaning" as "to have a meaning according an interpretation". Therefore, a syntactic object is regarded as also "making sense", a "formal sense", and when it is interpreted, it obtains a meaning per the interpretation. Due to this treatment, the syntax is immersed into the semantics, and this permits applying semantic approaches also to syntax.

The term "name" is used as short for "denoting phrase" and, same as "denoting phrases", a name can be a "proper name" or a "common name". Also, very long "denoting phrases" are regarded as "names". For example, the declarative sentences are treated as names of their truth values. This treatment can be extended from assertions to questions and other types of sentences by using the approach of [9]. Any text is treated here as a name.

A name M is said to have an "intension", which is denoted as "$\iota(M)$" or, when there is no risk of confusion, as "ιM". An intension should be intuitively treated as an abstraction from synonymous names,

a special kind of abstract name to be used in metalanguage rather than in the language of discourse. In terms of the aggregation theory, *the intensions of names are treated here as the identities (IDs).*

A name M has "denotata", and the collection of all its denotata is called "extension of M", and is denoted as "$\varepsilon(M)$" or "εM". In terms of aggregation theory, *the extension "$\varepsilon(M)$" of a name M coincides with what is called extension of an object in aggregation theory and is denoted by the same symbol.*

Whereas ιM and εM denote two objects, "ι" and "ε" denote two functions, which are called here "attributive functions". This term sounds appropriate, since a name is represented as a formal object (a string of characters or sounds) devoid of "intension" or "extension" and these latter ones are "attributed" to it. This wording sounds correct for anything which has attributes, not only for names.

The term "formal application" of an operation is customarily used to refer to the formation of a "functional notation", which in mathematics usually looks like this: "$f(x_1, ..., x_n)$", whereas in natural languages this is a sequence of words separated by punctuation marks, a grammatically written text. In this paper, a text is regarded as "composed", or longer, "built by repetitive formal application of *composition operations*". The composition operations act on both the syntactic and the semantic level.

A fundamental unary composition operation is called "formal sense operator" for its role in semantics, and "atomification" for its role in the nominalistic interpretation of set theory. The result of application of this operation to a name X is denoted as "$[X]$" for both its roles for reasons of "notational economy". On its role in semantics, the formal sense operator prescribes ignoring the meaning of a meaningful name, i.e. considering it as a formal expression, a syntactic object, and when applied to a formal expression, this operation results in this formal expression. On its role in the nominalist interpretation of set theory, the atomification operation prescribes to treat a name as an atom (not a set), and one can say that atomification operation maps the language into the universe of discourse of aggregation theory.

4.2 The approach to the semantics of metalingua

There are different methods of specifying the semantics of concrete languages, and before the semantics of μL is presented, some general explanations about our approach are in place. In particular, one would expect an answer to the question "Why the semantics of μL, presented here, is 'extensional', and how 'complete' is such a semantics, given that there might exist also an 'intensional semantics'?". In this section the explanation is given why an "intensional semantics" is not needed.

According to general practices, a language is regarded as specified, if two inventories are given: (1) a "vocabulary", which is a set of atomary expressions called here "vocabulas", and (2) a "grammar", which is a set of composition operations' notations, with rules for markup of their application, as well as a mechanism to specify notations for new composition operations. To specify the semantics of a language is to show how its composition operations "interact" with attributive functions – here, with intension and extension attributive functions. In other words, if "$C(X_1, , X_n)$" is a composition operation, then one would need to specify how $\iota C(X_1, , X_n)$ can be defined through the intensions $\iota X_1, ..., \iota X_n$, and how $\varepsilon C(X_1, , X_n)$ can be defined through the extensions $\varepsilon X_1, ..., \varepsilon X_n$.

Next, an explanation follows why one can limit to specifying the semantics by showing the above "interaction" of composition operations *only* with the extension attributive function (not also with the intention). Notice, that Frege did not mention about any concrete correlation between the two attributes of a name, but a correlation is rather evident – namely, *to know the meaning of a name is also to know what it denotes*, or in other terms, "the meaning of a name determines its extension". In precise terms, this is formulated here like this: there exists a one-to-one map $\tau : \iota M \mapsto \varepsilon M$, such that $\tau(\iota M) = \varepsilon M$. Here "$\tau$" comes from "tension" (a term, which in folklore is customarily used to generalize both terms, "intension" and "extension"). Proceeding from the above, one can consider the intension of a name as a special identity (ID) of its extension, a multi-identity set, and τ as a one-to-one correspondence between extensions and the selected intensions. Struc-

turally, the operator denoted as τ is an isomorphism. Thus, one can indicate the semantics of a composition operation by showing only how it "interacts" with extension of the name. Therefore, the function playing the role of "interpretation", which defines semantics of expressions is denoted here as "ε".

4.3 The syntax of metalingua

A more complete specification of μL is given in [6]. Here, only a subset of μL is presented to showcase the approach. The names and notations of the composition operations of μL follow below:

- Two symmetric binary composition operations – "direct modification" denoted as ": " (colon followed by space), and "inverse modification" denoted as " :" (infix space followed by colon);
- One binary composition operation called "union" denoted as ", " (comma followed by a space);
- Three unary composition operations called and denoted in this manner: "atomification" denoted by square brackets "[...]", "individuation" denoted by angular brackets "$\langle ... \rangle$", "association" denoted by round brackets "(...)".

Some symbols of μL indicated above can be defined through others, and are considered as part of the μL for convenience.

4.4 The semantics of metalingua

A text is a sequence of words separated by punctuation marks. The space between words is treated here also as a punctuation mark – one which denotes the "modification operation". In natural languages, the meaning of words is context-sensitive and the minimal context of a word is the adjacent word. By default, two adjacent words represent a context modifying each other, a default overriden by inserting between them a "visible" punctuation mark. The modification operation is

interpreted as adduction or as adjunction, depending on whether its "direct" or "inverse" presentation is used.

Consider the expression "white house", where the word "white" is a modifier (and where the word "house" is "the modified", a term used here to avoid the scientific jargon term "modificand"). Here, the expression "white house" is treated as the result of application of the direct modification operation, which can be denoted as "white: house". Also, consider the Romanian translation "casa alba" of the expression mentioned above, where the modifier is "alba", and the application of inverse modification operation should be denoted as "casa :alba". Notice, that the colon needs to be used near the modifier. Also, notice that the appropriate manner in which to name the two presentations, the "direct" and the "indirect" ones, depends on the concrete language and on consensus. Thus, the interpretation in aggregation theory of the modification, in its inverse presentation, can be defined like this:

$$\varepsilon(x : y) = \varepsilon(x) : \varepsilon(y), \tag{5}$$

where the colon in the two places is accompanied by spaces differently and it has different meanings.

The union operation, denoted as comma followed by space, is interpreted as the set-theoretic union operation, which is defined here also for Quine atoms and urelements. It is justified to denote by comma both the operation composing a list like "John, Peter, Amy" and a union of sets for this reason: for two Quine atoms x and y, it is true that $\varepsilon(x, y) = \{\varepsilon(x),\ \varepsilon(y)\}$. Also, this interpretation allows for the "strange" expressions like "horses and unicorns" to be treated on common grounds with expressions commonly accepted as making sense.

To clarify what are atomification and individuation operations, one needs to take into account the "use-mention distinction" ([14], p. 23). An expression can be "used" to refer to something, or it can be "mentioned" ignoring that it can refer to anything. The atomification operation of μL serves for mentioning an expression X and its application is marked up as $[X]$. In natural languages one mentions an expression by enclosing it between quotation marks. Obviously, a "mentioned"

expression is used in "formal sense". In contrast with atomification, the individuation operation is applied to indicate that an expression X should be treated as "used", and namely, used as an "individual". Its application is marked up as "$\langle X \rangle$". The interpretation in aggregation theory of these two operations looks like this;

$$\varepsilon([x]) = \{x\},\ \varepsilon(\langle x \rangle) = \{\varepsilon(x)\}.$$

The term "individuation" comes from the practice to use the term "individual" as a "generic term" for atoms and sets, but *not* also for classes. Thus, the result of application of "individuation operation" is an "individual". The fact that the symbol of this operation is a pair of angular brackets, and not the braces which serve as a symbol of a set defined by the list of its elements, is due to the treatment of individuation as a unary operation. The braces $\{.\}$ used in denotation of a finite set of n elements, are commonly treated as denoting a n-ary operation and cannot be used as a symbol for individuation operation.

Union operation denoted as comma followed by space coincides with the union operation on sets and, obviously, is defined also for Quine atoms, since these are a special kind of singletons. The union operation is also defined for urelements, which are treated as the empty set with many identities (IDs). Notice, that the modification operation can be expressed through union and individuation like this:

$$(x : y) = (< x >,\ y).$$

Notice, that the parentheses are used on the right side to avoid confusions, which can appear with expressions containing punctuation marks, and are also used on the left side for the sake of symmetry.

The association operation application is marked up by parentheses and is treated as acting on the level of syntax to disambiguate the application of operations. Its name was derived from the practice to refer as "left associated" or "right associated" to mathematical expressions with dropped brackets. The treatment of parentheses as denoting an operation might sound as a surprise, but notice that the syntax of μL does not require parentheses for denoting the application of operations,

as this is usually done. The reason for this, is that ambiguous expressions are commonly used in natural languages and the syntax of μL must comply with this.

5 Conclusions

Two ideas have been developed in this paper – the extensional model of languages, and the nominalistic interpretation of set theory – as well as a formal language to serve both for modeling languages and as a language appropriate for the nominalistic interpretation of set theories.

The approach with these three components leads towards a structural theory of "aggregates", where an aggregate can be defined as a structure consisting of a set and an ID which plays the role of an "intension". Based on this idea, the approach presented here can also serve as a *data model* for the "intensionalized data" introduced in [12] for use in informatics, treated here as the science of *data structures*.

References

[1] J. Bell. Sets and Classes as Many. In: Journal of Philosophical Logic. Vol. 29, No.6, pp. 585-601. (2000).

[2] P. Bernays. *A System of Axiomatic Set Theory.* (Part I). The Journal of Symbolic Logic (Association for Symbolic Logic) 2 (1). pp. 65-77, (1937).

[3] R. Carnap. *Logical Syntax of Language*. Rutledge. (2000).

[4] I. Drugus. *An Algebraic Set Theory Based on Induction.* 7th European Congress of Mathematics (2016) (to appear).

[5] I. Drugus. *PML: A Punctuation Symbolism for Semantic Markup.* 11th International Conference Linguistic Resources and Tools for Processing The Romanian Language, ConsILR-2015, 26-27 Nov. 2015, Iasi, Romania. pp. (2015), pp. 79-92.

[6] I. Drugus. *On Set-theoretic Semantics of Metalingua*. Proceedings of 8th International Conference on Computational Collective Intelligence Technologies and Applications, 28 - 30 September, 2016, Halkidiki, Greece (2016) (to appear).

[7] I. Drugus. *Metalingua: A Language to Mediate Communication with Semantic Web in Natural Languages. In: Advanced Information Technology in Education*, AISC 126. K.S. Thaung (Ed.), Springer-Verlag Berlin, Heidelberg. pp. 109-115. (2012).

[8] G. Frege. *On Sense and Reference.* Translations from the Philosophical Writings of Gottlob Frege. Blackwells, London (1892/1966). P. Geach, M. Black(Eds.) pp. 56-78. (1892).

[9] J. Groenendijk, F. Roelofsen. *Inquisitive Semantics and Pragmatics*. The Stanford workshop on Language, Communication and Rational Agency, May 30-31, (2009).

[10] L. Kirby, *Finitary Set theory*. Notre Dame Journal of Formal Logic, Volume 50, Number 3, pp. 227-244, (2009).

[11] D. Lewis. *Parts of Classes*. Basil Blackwell. p. 9. (1991).

[12] M. Nikitchenko, A. Chentsov. *Basics of Intensionalized Data: Presets, Sets, and Nominats*. Computer Science Journal of Moldova, vol.20, no.3(60) (2012).

[13] B. Russell. *On Denoting*. Mind, New Series, Vol. 14, No. 56. pp. 479–493. (1905).

[14] W. V. Quine. Mathematical Logic. Cambridge, MA. (1981).

Ioachim Drugus[1]

[1] Institute of Mathematics and Computer Science, Academy of Sciences of Moldova
Email: ioachim.drugus@math.md

Diachronic Analysis Using a Statistical Model

Daniela Gîfu

Abstract: This paper describes a statistical methodology for a diachronic study on a large corpus (a collection of publications, written from the second decade of the 19th century in two countries, Romania and Republic of Moldova, known as Bessarabia). The aim of this work is to analyse the lexical evolution of words in all four regions using a machine learning approach to identify the patterns that govern language changes. Basically, it was developed a mechanism for automatic correlation of different forms of the same words in order to train a statistical model on a list of known word-to-word correlations between lexicons.

Keywords: statistical model, diachronic study, corpora, lexical evolution, writing press.

1 Introduction

This study is based on diachronic exploration of Romanian and Bessarabian texts in order to implement a methodology for detecting automatically the language variation from the second decade of the 19th century to nowadays using a probability distribution estimation model, called MaxEnt (*Maximum Entropy*). Actually, this work is a continuation of a previous one (Gîfu & Simionescu, 2016), here a priori division of the temporal axis was excluded.

The present research is based on the question *how Romanian language has evolved at a particular period in different historical places?*

The language variation is often narrowed to consideration of change in one aspect of language: lexis, morphology, phonology, syntax, and semantics. A language variation in fact occurred also at the levels of discourse and pragmatics (Gass et al., 1989). The diachronically contrastive studies of the Romance languages (e.g. Romanian, Spanish,

French, Italian, and Portuguese) expose the presence of many similarities. (Densuianu, 1902). The Romanian language with approximately 24 million speakers has an important particularity. It is still spoken in Eastern Europe, with official status in Romania, Moldova, and parts of Serbia and Greece. Moreover, Romanian is recognized in Hungary (historical reasons) as a minority language and spoken in Ukraine, Albania, and Macedonia. (Miller-Broomfield, 2015).

For instance: the noun *prieten* → (EN. *friend*) in Romanian to five Romance languages has the same origin, Latin (*amicus*): Romanian – *amic* is synonym with prieten[1], Spanish – *amigo*, Catalan – *amic* is a dialect of Spanish, French – *ami*, Italian – *amico*, *and* Portuguese – *amigo*.

The paper is structured as follows: Section 2 presents briefly relevant literature that reveals a large interest for diachronic studies. Section 3 describes a methodology for research of language using a statistical model, on a list of known word-to-word correlations between lexicons, Section 4 presents the statistics results using machine learning model. Finally, the survey conclusions and future work are given in Section 5.

2 Related Work

Until now, the Romanian diachronic phenomenon was analysed using various methods. One of them relies on reconstructing a diachronic morphology for Romanian (Cristea et al., 2012), based on the digital version of the Romanian Language Thesaurus Dictionary (eDTLR) (Cristea et al, 2007). The authors detected the old form words occurring in the citations. For the Bessarabia, a group at the Institute of Mathematics and Computer Science, Academy of Sciences of Moldova proposed a technology based on transliteration and parallel texts alignment for creation of linguistic lexicon for Bessarabian corpus in Cyrillic script between 1967–1989 starting from actual Romanian lexicon. (Boian et al., 2014).

[1] In this study we used Romanian WordNet (http://dcl.bas.bg/bulnet/) the largest lexical ontology available today with a large collection of synsets. A synset is the wordnet's basic unit, being a set of synonyms which defines a specific meaning, common to the members of the synset. (Tufiș & Cristea, 2002; Tufiș et al., 2004, Ștefănescu, 2015).

Also, a study of language as an evolutionary phenomenon is included in (Mihalcea and Năstase, 2012). Their task was word epoch disambiguation, using text classification according to a specific epoch, knowing that the language is a dynamic phenomenon over time, being dependent on context. Actually, we found useful to statistical tests presented for epoch detection in (Popescu & Strapparava, 2013/2014), also, called temporal dynamics in (Wang & McCallum, 2006; Wang et al., 2008; Gerrish and Blei, 2010). Moreover, the diachronic text evaluation requires the automatic system in order to identify the epoch when the newspaper article was written (Gîfu, 2016; Popescu and Strapparava, 2015).

In order to evaluate the writing styles more researchers considered various indices: text features (Dascălu & Gîfu, 2015), textual formality (Eggins and Martin, 1997), and textual styles (Biber, 1987).

The development and use of software for natural language processing (NLP) highlight the defining aspects of two journalistic languages (Romania and Bessarabia) that have many similarities on the time axis that we have chosen. (Gîfu, 2014/2015). Furthermore, the diachronic study continues with exploring the patterns that govern the lexical differences between two lexicons, based on machine learning approach (Gîfu & Simionescu, 2016). This paper considers the study of the evolution of Romanian language focused on the lexical similarity based on statistical model.

3 Work Methodology

This section describes a language study based on a historical corpus used for investigating the evolution of words over time. This work is based on the Maximum Entropy (MaxEnt) text classifier being commonly used in Natural Language Processing (NLP) tasks, introduced first by Berger [Berger, et al, 1996] and Della Pietra [Della Pietra et al., 1997] in statistical estimation and pattern recognition. Noteworthy is that MaxEnt classifier has great results when the training corpus is limited, as in this case. Actually, the differences between the „source" lexicons (Moldavian, Transylvanian, Wallachian, and Bessarabian) and the „destination"

lexicon (DEX-online[2]) from the perspective of transformation patterns, were analyzed using this model. All the substring replacement operations (referred as "REP") are classified and extracted from the known correlations list, based on the character-level context in which they are applied in the source word. Based on these REPs, a set of fictive/candidate words are generated, each having a trust score attached. If a candidate word is found in the destination lexicon, the two words are marked as a corresponding pair.

Basically, all unknown words are extracted in order to find them the current correspondent. By applying these REPs operations on the first word of the pair (the old word), the present word[3] is obtained.

For instance the vowel [u] at the end of words that only had a phonetic significance.

Example: `tot`**`u`** `= totul` (Transylvania, 1881)

The consonant [s] (deaf) intervocalic is vocalized; thus it became the consonant [z].

For instance: `mu`**`s`**`ician = musician` (Wallachia, 1919)

The vowel [i] becomes in some situations [î].

An example: **`i`**`n =` **`î`**`n` (Moldavia, 1869)

The inflexion of words is often different: [ei] is transformed in [ii].

An example: `reclădir`**`ei`** `= reclădir`**`ii`** (Bessarabia, 1918)

Of course, these are the simplest situations, when we talk just one REP operation. But, in the present corpus, we have complex cases, when several REPs operations have intervened. To increase the accuracy of statistical data in identifying automatically the correlations, we decided to focus just three operations REPs for each words pair (old - new).

[2] www.dexonline.ro

[3] There was used the morphologic dictionary there (e.g. DEX on-line – www.dexonline.ro)

To illustrate this option, below is one example for each geographical area that includes 3 REPs operations:

Transylvania: a noun `serbâtoria` (En: *celebration*) in the direct case:

`serbâtoria = **sărbătoarea:** î->ă ria->area e->ă`

where `serbâtoria` is the "source" word, and `sărbătoarea` is the "destination" word.

Wallachia: a noun `esposiţiunea` (En: *exhibition*) in the direct case:

`esposiţiunea – **expoziţia:** s->z unea->a s->x`

Moldavia: a predicative verb `măngăemu` (En: *cosset*) in the indicative moode:

`măngăemu = **mîngîiem:** ă->îi ă->î u->`

Bessarabia: a noun `iantămplari` (En: *events*) in the direct case:

`iantămplari = **întîmplări:** ia->î ă->î a->ă`

As it was mentioned, this model is trained on a list of known word to word correlations between two lexicons (*source* and *destination*). For this study the size of the training data was not too big (40% from the current corpus), but we tried to cover a wide variety of lexical evolution phenomena. Basically, the trained model is used for predicting REPs which can be applied on a previously unknown word from the source lexicon.

3.1 Corpus

The corpus includes articles (over 3 million lexical tokens), chronologically ordered, from the most important Romanian and Bessarabian publications since 1917 until nowadays (Table 1). Moreover, the corpus was developed and structured in four independent collections of publications corresponding to Moldavia (*Albina românească*; *Convorbiri literare*; *Curierul*. *Foaia intereselor generale*; *Constitutionalul*; *Moldova Socialistă*; *Scânteia*; *Noutatea*; *Deşteptarea*; *Bună ziua, Iaşi*; *Ziarul de Vrancea*; *Monitorul de Vaslui*; *Evenimentul regional al Moldovei*; *Imparţial*), Wallachia (*Curier românesc*; *Buletin. Gazeta oficială*; *România*; *Curierul românesc*; *Pressa*, *România liberă*; *Românulu*; *Timpul*; *Literatorul*; *Albina*; *Deşteptarea*. *Foaie pentru*

popor; *Adeverul*; *Curierul artelor*; *Dimineața*; *Universul*; *Viitorul*; *Curentul*; *Universul literar*; *Adevărul*; *Adevărul literar și artistic*; *Scânteia*; *Romania literară*; *Dimineața copiilor*; *Evenimentul zilei*; *Gândul*; *Ziua*; *Ziua news*; *Ziua veche*), Transylvania (*Organulu Luminarei*; *Gazeta de Transilvania*; *Gazeta Transilvaniei*; *Telegrafulu Românu / Telegraful român*; *Foaia pentru Minte Anima și Literatură*; *Transilvania*; *Federațiunea*; *Gura Satului*; *Albina*; *Telegraful Românu*; *Familia*; *Aradu*; *Patria*; *Chemarea tinerimei române*; *Dreptatea*; *Aradul*; *Curierul creștin*; *Vatra românească*; *Echinox*; *Adevărul de Cluj*; *Făclia*; *Monitorul de Cluj*; *Bihoreanul*), and Bessarabia (*Basarabia reînoită*; *Curierul*; *Candela*; *Deșteptarea*; *Viața economică din Bălți*; *Solidaritatea*; *Ehos*; *Buletinul Arhiepiscopiei Chișinăului*; *Cuvânt moldovenesc*; *Basarabia*; *România nouă*; *Sfatul țării*; *Democratul Basarabiei*; *Glasul Basarabiei*; *Luminătorul*; *Dreptatea*; *Basarabia Chișinăului*; *Literatura și artă*; *Moldova Socialistă*; *Jurnal*; *Contrafort*; *Jurnal de Chișinău*; *Moldova suverană*; *Ziarul de gardă*) that was a part of old Moldavia until 1812, and then between 1918-1941, becoming an independent state since 1991.

3.2 Preprocessing chain

The automatic preprocessing chain applied on this corpus consists of the following sequences: segmentation, tokenization, part-of-speech tagging, lemmatization, using the Romanian POS-tagger (Simionescu, 2011). The final XML includes an extra markup attribute, `NotInDict`. Each NotInDict is a token which is not recognized by DEX-online.

Below is a segmentation annotation in XML standoff format from Albina (Transylvania), 1884:

Trăim în nisce...

where *nisce* is an old form (marked with NotInDict) of the indefinite article, *niște.*

```
  <?xml          version="1.0"          encoding="UTF-8"
standalone="no"?>
<POS_Output>
<S id="4" offset="186">
```

```
    <W   EXTRA="tranzitiv"   LEMMA="trăi"   MSD="Vmip1p"
Mood="indicative"       Number="plural"       POS="VERB"
Person="first"    Tense="present"    Type="predicative"
id="4.1" offset="0">Trăim</W>
    <W  LEMMA="în"  MSD="Sp"  POS="ADPOSITION"  id="4.2"
offset="6">în</W>
    <W          Case="direct"          Definiteness="no"
EXTRA="NotInDict"    Gender="feminine"    LEMMA="nisce"
MSD="Ncfsrn"         Number="singular"         POS="NOUN"
Type="common" id="4.3" offset="9">nisce</W>
…
</S>
```

The global situation is related in Table 1 and represented graphically in Figure 1.

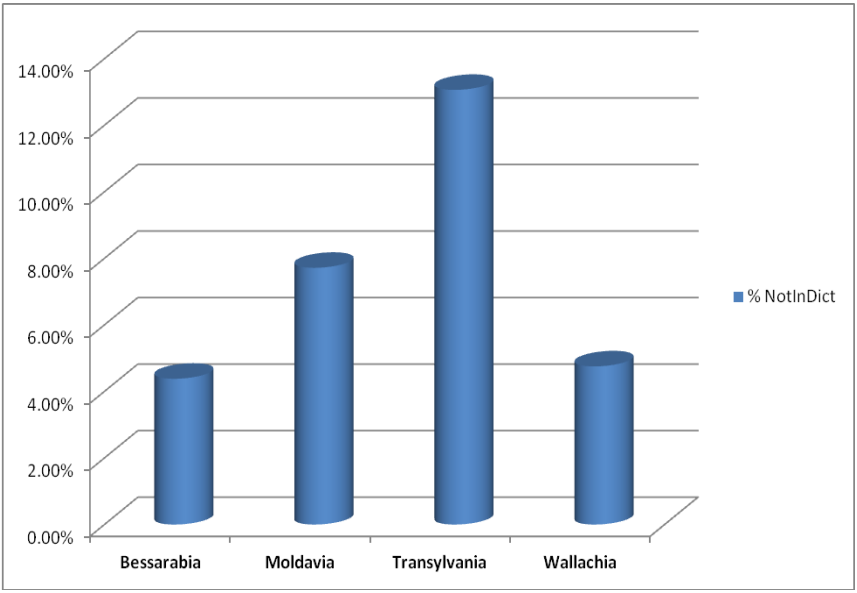


Figure 1. Percentage of NotInDict Words - 1817-2015

Table 1. General corpus statistics

Region	Time	Total considered words[4]	Total unknown Words	Total unique unknown words	%(total unknown words/total words)
Moldavia	1829-2015	65901	5085	2979	7.72
Wallachia	1829-2015	137261	6525	4105	4.75
Transylvania	1837-2015	160923	21023	8518	13.06
Bessarabia	1817-2015	107324	4703	2891	4.38

Although the four corpora are slightly disproportionate as number, the Transylvania case is different from the other three. The language in Transylvania is marked by historical waves: 1849-1860, the official language is German, including in administration; 1860-1866, the Romanian language returned as the official language, following the Romanian claims; 1867-1914, the Dual Monarchy is installed, which grants visible linguistic concessions by nationalities law.

[4] From the total tokens the punctuation, the numbers and the words with less than two characters were removed.

4 Statistics and interpretation

In this section the statistical results for the 4 collections of journalistic texts that correspond to Moldova, Transylvania, Wallachia and Bessarabia will be highlighted. There was used a mechanism of automatic correlation of unknown words with the new ones, presented above.

In Table 2, we consider the most common REPs (12) for the present corpus for a common period (1840-2015).

Table 2. The percentage of REPs

REP	1840 - 2015 Wallachia	1840 - 2015 Transylvania	1840 - 2015 Moldavia	1840 - 2015 Bessarabia
u ->	2.61%	**18.26%**	**11.55%**	5.79%
s -> z	**12.89%**	5.19%	9.06%	4.06%
e -> ă	5.91%	3.83%	6.22%	1.66%
ei -> ii	6.27%	2.56%	3.58%	**7.86%**
e -> i	2.61%	2.10%	3.09%	2.07%
i -> î	0.75%	1.99%	5.82%	1.82%
i -> e	3.22%	1.23%	3.04%	3.23%
ĭ -> i	2.04%	1.77%	1.14%	1.99%
a -> ă	1.22%	1.97%	1.19%	1.16%
s -> x	3.58%	1.48%	0.70%	0.08%
a ->	1.97%	1.30%	2.39%	2.73%
e -> î	1.72%	1.38%	2.34%	0.91%

It can be seen that in Transylvania and Moldavia similarities appear in writing rules, if we look at the hierarchy of these REPs (first 5). There was a special situation, Bessarabia between 1945-1989, a period when nothing was published anymore in Latin script (except the war years). This period has not been considered in this study. The vales from the Table 2 are represented in Figure 2.

All these REPs will become an important rules-based system very useful to develop a diachronic POS tagger for Romanian, another future work direction. We believe this MaxEnt model could be used to add

enhanced support for unknown words in order to develop the POS-tagger[5] for contemporary Romanian used in this paper (a free online service). This collection of publications can be considered a start for developing a Gold Corpus required for training such a diachronic POS tagging model.

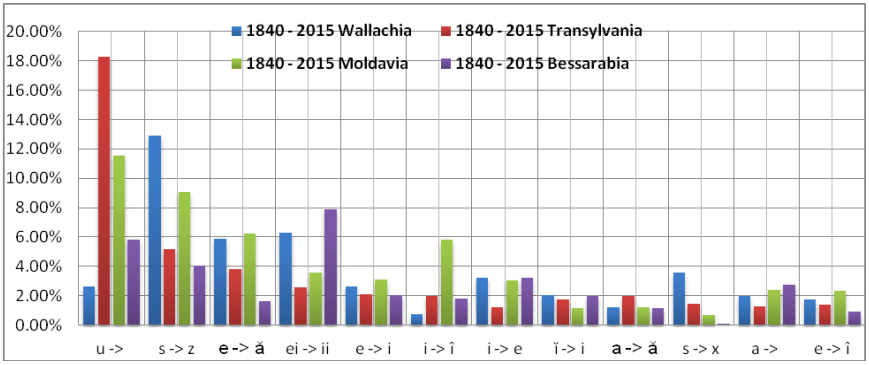


Figure 2. The most frequent REPs

The results from Table 3 are very promising.

Table 3. Known words vs. identifiable words comparison

Region/ statistical parameter	Transylvania	Wallachia	Moldavia	Bessarabia
Known words	85.96%	94.98%	91.82%	95.91%
Identifiable words	97.40%	98.11%	96.55%	97.80%
MaxEnt model - Precision	**81,39**	**82,92**	**77,05**	**86,81**

The "Identifiable words" represent the percent of words from the texts with known words or which can be correlated automatically with a known form. As we can see, for all geographical areas this indicator is over 95%. In this case, over 96% of the words can be recovered, but only

[5] http://nlptools.info.uaic.ro/WebPosRo/

76% of these are automatically correlated with a precision of 82%. Regarding the indicator "Known Words", only for Transylvania the result is smaller, because the corpus was bigger.

5 Conclusions and discussions

The methodology presented is language independent and it offers a basis for future large-scale studies, having a large impact on reducing the amount of human effort required by linguistic analysis of language variants.

This work presents a language variation over time in order to compare the journalistic language changes in four regions, Moldavia, Wallachia and Transylvania (Romania) and Bessarabia (a historical part of Romania). This survey investigates the problem of journalistic language similarity between cognate languages. The statistical results show the fact that there exists a high level of similarity between the lexicons of those four historical Romanian regions analyzed, at least in the newspapers.

In the future, an interesting experiment could be focused on the transliteration differences from Cyrillic to Latin both in Romania and Bessarabia until 1862, when in Romania the texts were published in both alphabets. Moreover, given that in the period 1944-1989 (excluding the war years) in Bessarabia the writing in Latin alphabet was prohibited, the process of collecting and transliterating publications of those times - with the support of the Academy of Sciences of Chisinau – should continue.

Acknowledgments. I would like to thank my colleagues, Radu Simionescu and Augusto Perez from the Faculty of Computer Science, "Alexandru Ioan Cuza" University of Iaşi for all support to finish this work.

References

[1] A. L. Berger, S. A. Della Pietra, and V. J. Della Pietra. (1996). *A Maximum Entropy Approach to Natural Language Processing*. In: Computational Linguistics, 22(1), pp. 39-71.

[2] D. Biber. *A textual comparison of British and American Writing*. American Speech, (62) (1987), pp. 99–119.

[3] E. Boian, C. Ciubotaru, S. Cojocaru, A. Colesnicov, L. Malahov. *Cultural and Historical Heritage Digitization, Recognition and Conservation*. In: Akademos: Revista de Ştiinţă, Inovare, Cultură şi Artă, nr. 1 (32) (2014), pp. 61-68.

[4] D. Cristea, M. Răschip, C. Forăscu, G. Haja, C. Florescu, B. Aldea, E. Dănilă, *The Digital Form of the Thesaurus Dictionary of the Romanian Language*. In: Proceedings of SpeD 2007 Speech Technology and Human - Computer Dialogue, Iași, May 10-12 (2007).

[5] D. Cristea, R. Simionescu, G. Haja. *Reconstructing the Diachronic Morphology of Romanian from Dictionary Citations*. In: Proceedings of LREC-2012, Istanbul, 21-25 May (2012).

[6] M. Dascălu, and D. Gîfu. *Evaluating the Complexity of Online Romanian Press*. In: Proceedings of the 11th International Conference Linguistic Resources and Tools for Processing The Romanian Language, ConsILR-2015, 26-27 Nov. 2015, Iaşi, Romania, "Alexandru Ioan Cuza" University Publishing House, Iaşi, Romania (2015), pp. 149-162.

[7] S. A. Della Pietra, V. J. Della Pietra, and J. Lafferty, J. (1997). *Inducing features of random fields*. IEEE Transactions on Pattern Analysis and Machine Intelligence, 19(4):380–393.

[8] A. Delmestri, and N. Cristianini, N. *String Similarity Measures and PAM-like Matrices for Cognate Identification*. Bucharest Working Papers in Linguistics, 12(2) (2010), pp. 71-82.

[9] O. Densusianu. *Filologia Romanică în universitatea noastră*, Bucuresci, J. V. Socecu Editeur (1902), p. 23.

[10] Ș. D. Dumitrescu. *Rowordnetlib – the First API for the Romanian WordNet*. In: Proceedings of the Romanian Academy, Series A, vol. 16: 1 (2015), pp. 87-94.

[11] S. Eggins, J. R. Martin. *Genres and Register of Discourse*. In: Dijk, T.A.v. (ed.) Discourse as Structure and Process (Discourse Studies – A Multidisciplinary Introduction), Vol. 1, Sage Publications, London, UK (1997), pp. 231–232.

[12] S. M. Gass, C. Madden, D. Preston, and L. Selinker (eds.) *Variation in second language acquisition*, vol. 1: Discourse and pragmatics, Clevedon, England (1989).

[13] S. M. Gerrish and D. M. Blei. *A language-based approach to measuring scholarly impact*. In Proceedings of International Conference of Machine Learning (2010).

[14] D. Gîfu. Diachronic Evaluation of Newspapers Language between Different Idioms at the first *Workshop on Natural Language Meets Journalism*, NLPMJ-2016, held at the *International Joint Conference on Artificial Intelligence*, IJCAI-16, 9-15 July 2016, New York, USA.

[15] D. Gîfu. *The Emotional Orientation*. In: Proceedings of the third Conference of Mathematical Society of the Republic of Moldova, "IMCS-50", 9-23 August 2014, Chișinău, Republic of Moldova (2014), pp. 511-516.

[16] D. Gîfu. *Contrastive Diachronic Study on Romanian Language*. In: Proceedings FOI-2015, S. Cojocaru, C. Gaindric (eds.), Institute of Mathematics and Computer Science, Academy of Sciences of Moldova (2015), pp. 296-310.

[17] D. Gîfu and R. Simionescu. *Tracing Language Variation for Romanian* at the 17th International Conference on Intelligent Text Processing and Computational Linguistics, CICLing 2016, Konya, Turkey (2016).

[18] R. Mihalcea and V. Năstase, V. *Word epoch disambiguation: Finding how words change over time*. In: Proceedings of ACL 2012 (2012).

[19] C. Miller-Broomfield. *Romanian: The Forgotten Romance Language*. In: Unravel: The Accesible Linguistics Magazine (2015), http://unravellingmag.com/articles/romanian-the-forgotten-romance-language/.

[20] O. Popescu and C. Strapparava. *Behind the Times: Detecting Epoch Changes using Large Corpora*. In: International Joint Conference on Natural Language Processing, Nagoya, Japan, 14-18 October 2013 (2013), pp. 347-355.

[21] O. Popescu and C. Strapparava. *Time corpora: Epochs, opinions and changes*. Knowledge-Based Systems (2014).

[22] O. Popescu and C. Strapparava. Semeval-2015 task 7: *Diachronic text evaluation*. In: Proceedings of SemEval (2015).

[23] R. Simionescu. *UAIC Romanian Part of Speech Tagger*, resource on nlptools.info.uaic.ro, "Alexandru Ioan Cuza" University of Iași (2011).

[24] D. Tufiș, E. Barbu, V. Barbu Mititelu, R. Ion, L. Bozianu. *The Romanian WordNet*, Romanian Journal of Information Science and Technology, 7, 1–2 (2004), pp. 107–124.

[25] D. Tufiş, D. Cristea. *Ro-BALKANET – ontologie lexicalizată în context multilingv pentru limba română*. In: Limba Română în Societatea Informațională – Societatea Cunoașterii, Ed. Expert, Tufiş, D., Filip, F. Gh. (coord.) (2002), pp. 139-166.

[26] X. Wang and A. McCallum. *Topics over Time: A Non-Markov Continuous-Time Model of Topical Trends*. In: KDD 2006, USA (2006).

[27] X. Wang, M. S. Gerber, and D. E. Brown. *Automatic Crime Prediction using Events Extracted from Twitter Posts*. SBP, LNCS 7227:231-238 (2012).

Daniela Gîfu

Faculty of Computer Science, "Alexandru Ioan Cuza" University of Iași
e-mail: daniela.gifu@info.uaic.ro

Proceedings of the Conference on Mathematical Foundations of Informatics
MFOI2016, July 25-29, 2016, Chisinau, Republic of Moldova

Properties of Nominative Programs Specified by Effective Definitional Schemes *

Ievgen Ivanov, Mykola Nikitchenko,
Volodymyr G. Skobelev

Abstract

In the paper we investigate the following problem: characterize the class of programs over complex data structures stable under natural data structures transformations. We propose a solution based on the composition-nominative approach. According to this approach data structures are considered as nominative data structures and semantics of programs are presented using special program algebras with operations called compositions. We define various data transformations and specify a very general class of compositions based on effective definitional schemes by H. Friedman. We show that this class preserves program stability under natural data structures transformations. This result can be useful in software development and verification.

Keywords: semantics, computability, algorithmic algebras, nominative data, composition, effective definitional scheme.

1 Introduction

The generalized recursion theory as proposed by H. Friedman [1] and subsequently developed in [2] investigates generalized notions of computability on objects of algebraic structures. In this context in [1] H. Friedman defined the notion of a generalized Turing algorithm and the

*This work was supported in part by the project "Development of logic-algorithmic methods for investigation of formal models of natural languages" of Taras Shevchenko National University of Kyiv, Ukraine, Ref. Nr. 0116U004780

equivalent notion of an effective definitional scheme (eds) [1]. Basically, eds are definitions by infinite cases which have a recursive enumerable structure. They can be used to give a very general definition of a computable function; in fact it was argued [2] that for reasonable definitions of computable functions over algebraic structures computable functions need to be eds definable.

Such a definition of a computable function can be described as follows [2]. Consider a language L with finitely many constant, relation, operation symbols interpreted in an algebraic structure M with some domain, constants, relations and operations. Then a function f (on the domain of M) is eds definable, if there is a set S of conditions of the form $\varphi_i(v, v_1, ..., v_n) \to t_i(v, v_1, ..., v_n)$ (eds) consisting of terms $t_i(v, v_1, ..., v_n)$ in L and basic semialgebraic conditions (i.e. finite conjunctions of atomic formulas and their negations) $\varphi_i(v, v_1, ..., v_n)$ in L, where $v, v_1, ..., v_n$ are formal variable names, such that S is effective (recursively enumerable as a set of strings) and there exist elements $a_1, ..., a_n$ of the domain of M (parameters of the definition) such that for each i: $f(x) = t_i(x, a_1, ..., a_n)$, if $\varphi_i(x, a_1, ..., a_n)$.

It is known [2, Theorem 2] that in a suitable formalization, programs expressible in imperative programming languages with variables ranging over M and assignments, stacks of values from the domain of M with the operations *Push* and *Pop,* conditional operators (*If-Then-Else*), and jumps (*Goto*) define eds definable functions.

However, as it is, the notion of eds definability has a limited applicability to semantics of programming languages, since it tells only what are computable functions from M (or more generally, M^n) to M or M^m, the elements of which can be considered as data which a program processes, i.e. it deals with computability of a program viewed as an input-output relation.

In contrast, in the context of semantics of programming languages [3, 4, 5], especially denotational and big-step operational semantics, it is more important to describe computability (over M) of the steps taken by the program during execution, which are usually transformations of program states to program states.

In this paper we propose a solution to this issue and generalize eds definability of functions on a structure M to eds definability of transformations of program execution states for programs operating on complex data structures (e.g. multidimensional arrays, lists, trees and tree-like structures, etc.) over M. We also list programming language constructs (compositions) which preserve eds definability and show that eds definability can be used to demonstrate stability of program semantics when the data structures used in the program are changed to equivalent in the sense of information content and supported operations, e.g. if jagged arrays in the program are replaced with 2D arrays.

2 Notation

We will use the following notation:

- $A \tilde{\to} B$ is the set of all partial functions from A to B;
- $f(x)\downarrow$ (or $f(x)\uparrow$) means that a partial function f is defined (or respectively, undefined) on an argument x. The notation $f(x)\downarrow R\,y$, where R is a binary relation e.g. equality ($=$), membership ($\in$), etc. means that $f(x)$ is defined and the relation $f(x)R\,y$ holds;
- $dom(f)$ denotes the set of arguments on which f is defined (domain of definedness), i.e. $dom(f) = \{x \mid f(x)\downarrow\}$;
- T, F denote the logical values (true and false);
- $f(x) \cong g(y)$ means the strong equality: if at least one of $f(x)$ and $g(y)$ is defined, then the another is defined and they are equal.

Let V be a fixed nonempty set of basic names and A be a fixed set of atomic values. For each $u, v \in V^+$ we will use the following notation:

- $|v|$ is the length of v;
- uv is the concatenation of the words u and v;

- $u \leq v$ means that u is a prefix of v, i.e. either $u = v$, or there exists $w \in V^+$ such that $v = uw$.

3 Program states and nominative data

Program states describe the contents of the execution environment and in the simplest case can be conveniently modeled as name-value mappings $[name_1 \mapsto value_1, name_2 \mapsto value_2, ...]$ (this approach is described in detail in [3]). Here the individual $name \mapsto value$ assignments are unordered, names correspond to program variable names, and values belong to the value ranges of the variables, i.e. to the domain of the structure M (we assume here a programming language with just one data type; generalizations for languages with complex type systems are possible as well).

Then, the semantics of built-in basic programming language operations on data and compound data-processing operations expressible in the language can be represented by functions mapping program states to program states [3], i.e. functions mapping name-value mappings to name-value mappings. The semantics of constructs (compositions) such as branching (*If-Then-Else*), cycle (*While*), etc. that allow expressing compound data-processing operations using simpler operations are mappings between functions on program states.

These ideas are further generalized e.g. in the composition-nominative approach to program formalization [6, 7, 8] which aims to build a mathematical basis for development of formal methods of analysis and synthesis of software systems. According to this approach, program models are specified as *composition-nominative systems* (CNS) which consist of simpler systems: composition, description, and denotation systems. Composition system defines semantic aspects of programs, description system defines syntactical aspects, and denotation system specifies meanings of descriptions. Semantics of programs are defined as partial functions over a class of data processed by programs. The means of construction of complex programs from simpler programs (e.g. branching, cycle, etc.) are defined as n-ary operations over func-

tions over data which are called compositions. A composition system can be specified as two algebras: a data algebra and a function algebra. Syntactically programs are represented as terms in the function algebra. The corresponding term algebra defines a descriptive system and the ordinary procedure of term interpretation gives a denotation system.

In the composition-nominative approach both the data on which programs operate and program states are modeled *in a unified way* as *nominative data* [6]. There are several types of nominative data [9, 10, 11, 12, 13, 14], but the common idea behind them is the name-value mapping mentioned at the beginning of this section.

The simplest kind of nominative data are nominative sets [6, 9] which are defined as partial functions that map names to values.

In the general case, nominative data are classified in accordance with the following parameters:

– *values* can be simple (unstructured) or complex (structured),

– *names* can be simple (unstructured) or complex (structured).

Complex values mean that values corresponding to names in a nominative data can, in particular, be nominative data themselves. Complex (structured) names are understood as strings consisting of simple (unstructured) names. The mentioned parameters give 4 types of nominative data denoted as follows:

TND_{SS}: nominative data with simple names and simple values,

TND_{SC}: nominative data with simple names and complex values,

TND_{CS}: nominative data with complex names and simple values,

TND_{CC}: nominative data with complex names and complex values.

The formal definitions of nominative data of different types and the corresponding examples are given below.

- For any fixed sets of names V and values A, the class of data of the type TND_{SS} over V and A is defined as $D_0(V, A) = V \xrightarrow{n} A$, where $V \xrightarrow{n} A$ denotes the set of partial functions from V to A which have a finite graph. The elements of this class are denoted using notation $[v_1 \mapsto d_1, ..., v_n \mapsto d_n]$, where $v_i \in V$ are names and d_i are the corresponding values. For example, $[u \mapsto 1, v \mapsto 2] \in$

$D_0(V,A)$, where $u, v \in V$ are distinct elements and $\{1,2\} \subseteq A$, d is a data $d \in D_0(V,A)$, such that $dom(d) = \{u,v\}$ and $d(u) = 1$, $d(v) = 2$.

- For any fixed sets of names V and values A, the class of data of the type TND_{SC} over V and A is $D_1(V,A) = ND(V,A)$, where
  - $ND(V,A) = \bigcup_{k\geq 0} ND_k(V,A)$,
  - $ND_0(V,A) = A \cup \{\emptyset\}$,
  - $ND_{k+1}(V,A) = A \cup \left(V \xrightarrow{n} ND_k(V,A)\right), \quad k \geq 0.$

  Here, we denote by $\emptyset$ the empty nominative data, i.e. a function with an empty graph (this notation is also used for the empty set). For the empty nominative data we will also use the notation $[]$.

  Data of type TND_{SC} are hierarchically constructed. An example of such data is $[u \mapsto 1, v \mapsto [w \mapsto 2]]$, where $u, v, w \in V$, $1, 2 \in A$. Such data can be represented by oriented trees (of varying arity) with arcs labelled by names and with leafs labelled by atoms.

  A *path* is a nonempty finite sequence $(v_1, v_2, ..., v_k)$, $v_1, ..., v_k \in V$.

  For a given data d, a *value of a path* $(v_1, v_2, ..., v_k)$ in d is defined by the expression $d(v_1, v_2, ..., v_k) \cong (...((d(v_1))(v_2))...(v_k))$.

  We say that a path $(v_1, v_2, ..., v_k)$ is *a path in a data* $d \in ND(V,A)$, if a value of $(v_1, v_2, ..., v_k)$ in d is defined, i.e. $d(v_1, v_2, ..., v_k)\downarrow$ (a path in data corresponds to a path from the root to a node in an oriented tree). A *terminal path* in a data $d \in ND(V,A)$ is a path in d such that its value belongs to $A \cup \{\emptyset\}$. The least k such that $d \in ND_k(V,A)$ is the *rank* of a data d.

- For any fixed sets of names V and values A, the class of data of the type TND_{CS} over V and A is defined as $D_2(V,A) = NDVS(V,A)$, where $NDVS(V,A)$ is the set of all elements of $A \cup (V^+ \xrightarrow{n} A)$ such that either $d \in A$, or $d \in V^+ \xrightarrow{n} A$ and all strings from $dom(d)$ are pairwise incomparable in the sense of the prefix relation (*principle of unambiguous associative naming*). An

example of such data is $[uv \mapsto 1, uw \mapsto 2, w \mapsto 3]$, $u, v, w \in V$. Such data have *complex names* i.e. names that are strings.

- For any fixed sets of names V and values A, the class of data of the type TND_{CC} over V and A is defined as $D_3(V, A) = NDVC(V, A)$, where $NDVC(V, A)$ is the class of all data $d \in ND(V^+, A)$ such that for any two paths $(u_1, u_2, ..., u_k)$ and $(v_1, v_2, ..., v_l)$ in d, neither of which is a prefix of another, the words $u_1u_2...u_k$ and $v_1v_2...v_l$ are incomparable in the sense of the prefix relation (*principle of unambiguous associative naming*). Such data is also called *complex-named data* [11]. An example of such data is $[uv \mapsto 1, w \mapsto [uw \mapsto \emptyset]]$, $u, v, w \in V$.

In [12] it was demonstrated that nominative data with complex names and/or values can be used to adequately represent many data structures used in programming practice. This gives a reason to model programs as partial functions that map nominative data to nominative data and to model means of program construction as compositions (n-ary operations on partial functions over nominative data).

In the previous work [12] the authors of this paper proposed to generalize Glushkov Algorithmic Algebras [15] to algebras of functions and predicates over nominative data of the type TND_{CC} (i.e. data with complex names and complex values) to obtain a rich, but tractable formalism language for specification and reasoning about programs. This generalization was called an Associative Nominative Glushkov Algorithmic Algebra (ANGAA). In this paper we will show that the compositions of ANGAA preserve eds definability.

In [12] it was proved that the programs expressible in ANGAA have an attractive property called *nominative stability* [12, 9, 10, 11]. This property is a formalization of the idea of stability of program semantics when the data structures used in the program are changed to equivalent in the sense of information content and supported operations.

It can be illustrated by the following feature of the Pascal programming language: the two-dimensional array definitions `var A: array [1..n, 1..m] of real` and `var A:array [1..n] of array [1..m]`

`of real` are equivalent and both the `A[i,j]` and `A[i][j]` syntax can be used to access the array elements regardless of the form of its definition (it should be noted that the languages like C++ and Java do not have this feature). This implies that one can safely swap two-dimensional array definitions in a program without changing the rest of the text of the program while preserving program semantics. This fact can be formalized in terms of nominative stability.

In more detail, nominative stability allows a programmer to construct a program oriented on a certain hierarchical naming structure of input data, but this program would give equivalent results, if input data were changed to equivalent data. Such stability simplifies programming with complex data making it "softer" because the programmer should not remember the current structure of data.

Formally, nominative stability is defined using the *nominative equivalence* relation on nominative data of the type TND_{CC} [12]. This relation is a formalization of the idea that data are equivalent, if they have essentially the same information content, but may have different hierarchical naming structure. For example, the following data are nominatively equivalent: $[v_1 \mapsto [v_2 \mapsto [v_3 \mapsto 1]]]$ and $[v_1 v_2 v_3 \mapsto 1]$, as they differ only in the naming hierarchy, but contain the same basic names and values. A function on nominative data is nominative stable, if on nominative equivalent data it gives nominative equivalent results.

Nominative equivalence is just one relation on data which is preserved by programs (functions) expressible in ANGAA. This gives a rise to a question of what other relations on data are preserved by programs expressible in ANGAA. In this paper we will give an answer to this question and using eds definability show that if the basic operations on data of ANGAA are monotone with respect to some partial order on data, then the programs expressible in ANGAA are monotone programs with respect to this partial order. This result substantially generalizes the fact that the programs expressible in ANGAA are nominative stable.

4 Algebra of Nominative Data

The main operations over nominative data are the operations of *denaming* (taking the value of a name), *naming* (assigning a new value to a name), and *overlapping*. In this section we will define these operations for data of the type TND_{CC}.

Let V and A be fixed sets of names and values.

Definition 1 (Denaming)**.** *The (associative) denaming is an operation $v \Rightarrow_a$ with a parameter $v \in V^+$ defined by induction on the length of v:*

- *if $|v| = 1$, then $v \Rightarrow_a (d) \cong \begin{cases} d(v), & \text{if } d(v)\downarrow; \\ d/v, & \text{if } d(v)\uparrow \text{ and } d/v \neq \emptyset; \\ \text{undefined}, & \text{if } d(v)\uparrow \text{ and } d/v = \emptyset, \end{cases}$ where $d/u = [v_1 \mapsto d(v) \mid d(v)\downarrow, v = uv_1, v_1 \in V^+]$;*
- *if $|v| = n > 1$, then $v \Rightarrow_a (d) \cong v_1 \Rightarrow_a (x \Rightarrow_a (d))$, where $v = xv_1$, $x \in V$, $v_1 \in V^+$, $|v_1| = n - 1$.*

The following examples illustrate this definition:
$u \Rightarrow_a ([u \mapsto 1, v \mapsto 2]) = 1$;
$(uv) \Rightarrow_a ([u \mapsto [vw \mapsto 1, u \mapsto 2]]) = [w \mapsto 1]$.

The name of this operation originates from the following property (*associativity*) [11]: $u \Rightarrow_a (d) \cong u_n \Rightarrow_a (u_{n-1} \Rightarrow_a (\ldots u_1 \Rightarrow_a (d)\ldots))$ for all complex names $u, u_1, u_2, \ldots, u_n \in V^+$ such that $u = u_1u_2\ldots u_n$.

Definition 2 (Naming)**.** *Naming is an unary operation $\Rightarrow v$ with a parameter $v \in V^+$ such that $\Rightarrow v(d) = [v \mapsto d]$.*

Overlapping can be intuitively considered as an updating operation which updates values in the first argument with the values of the second argument taking into account their names. For the types of nominative data with complex names and/or values different overlapping operations can be considered. We will define two kinds of overlapping: global and local overlapping. Global (associative or structural) overlapping ∇_a updates several values while the local one ∇_a^v (with a parameter name v) updates only one value with complex name v.

The global overlapping can be used for formalization of procedures calls and the local operation formalizes the assignment operator in programming languages. Intuitively, this operation joins two data and resolves name conflicts in favour of its *second* argument.

Definition 3 (Global overlapping)**.** *For nominative data of the type TND_{CC}, (global) overlapping is a binary operation ∇_a defined inductively by the rank of the first argument as follows.*

Let $NDVC_k(V,A) = NDVC(V,A) \cap ND_k(V^+,A)$ be the data from the set $NDVC(V,A)$, the rank of which is $\leq k$.

Induction base of the definition. If $d_1 \in NDVC_0(V,A)$, then

$$d_1 \nabla_a d_2 \cong \begin{cases} d_2, & \text{if } d_1 = \emptyset \text{ and } d_2 \in NDVC(V,A) \backslash A; \\ \textit{undefined}, & \text{if } d_1 \in A \text{ or } d_2 \in A. \end{cases}$$

Induction step of the definition. Assume that the value $d_1 \nabla_a d_2$ is already defined for all d_1, d_2 such that $d_1 \in NDVC_k(V,A)$. Let

$$d_1 \in NDVC_{k+1}(V,A) \backslash NDVC_k(V,A).$$

Then $d_1 \nabla_a d_2 = d$, where d is defined for each name $u \in V^+$ as follows:

1) $d(u) = d_2(u)$, if $u \in dom(d_2)$ and u does not have a proper prefix which belongs to $dom(d_1)$;

2) $d(u) = d_1(u) \nabla_a (d_2/u)$, if $d_1(u)$ is defined and does not belong to A and u is a proper prefix of some element of $dom(d_2)$, where $d_2/u = [v_1 \mapsto d_2(v) \mid d_2(v)\downarrow, v = uv_1, v_1 \in V^+]$;

3) $d(u) = d_2/u$, if $d_1(u)$ is defined and belongs to A and u is a proper prefix of some element of $dom(d_2)$;

4) $d(u) = d_1(u)$, if $d_1(u)$ is defined and u is not comparable (in the sense of the prefix relation) with any element of $dom(d_2)$;

5) $d(u)\uparrow$, otherwise.

The global overlapping has the following properties [11]:

- $[u \mapsto d_1] \nabla_a [v \mapsto d_2] = [u \mapsto d_1, v \mapsto d_2], \quad u, v \in V, \quad u \neq v;$

- $[uv \mapsto d_1]\nabla_a[u \mapsto d_2] = [u \mapsto d_2]$, $u, v \in V^+$, i.e. the value under a name u in second argument overwrites the value under names in first argument, which are extensions of u;
- $[u \mapsto d_1]\nabla_a[uv \mapsto d_2] = [u \mapsto (d_1\nabla_a[v \mapsto d_2])]$, if $u, v \in V^+$, $d_1 \notin A$, i.e. the value under a name uv in second argument modifies values under prefixes of uv in first argument;

Definition 4 (Local overlapping)**.** *For nominative data of the type TND_{CC} local overlapping is a binary operation ∇_a^v with a parameter $v \in V^+$ defined as follows: $d_1\nabla_a^v d_2 \cong d_1\nabla_a(\Rightarrow v(d_2))$.*

Besides operations, there are important predicates on nominative data:

Name checking predicate $u!$ on $NDVC(V, A)$ with a parameter $u \in V^+$: $u!(d) = T$, if $u \Rightarrow_a (d)\downarrow$; $u!(d) = F$, if $u \Rightarrow_a (d)\uparrow$.

Emptiness checking predicate $IsEmpty$ on $NDVC(V, A)$: $IsEmpty(d) = T$, if $d = \emptyset$; $IsEmpty(d) = F$, if $d \neq \emptyset$.

Definition 5. *An algebraic structure of nominative data of the type TND_{CC} is $NDAS_{CC}(V, A) = (NDVC(V, A); \emptyset, \{v \Rightarrow_a\}_{v\in V^+}, \{\Rightarrow v\}_{v\in V^+}, \{\nabla_a^v\}_{v\in V^+}, \{v!\}_{v\in V^+}, IsEmpty)$. Here $\emptyset$ is a constant – the empty nominative data (note that this is a structure without equality).*

Definition 6. *1) A path in $d \in NDVC(V, A)$ is a nonempty sequence $(v_1, v_2, ..., v_n)$ of words from V^+ such that the value $((d(v_1))(v_2)...)(v_n)$ is defined. The value $((d(v_1))(v_2)...)(v_n)$ is the value of the path $(v_1, v_2, ..., v_n)$ in d.*

2) A path in a complex-named data $d \in NDVC(V, A)$ is called a terminal path, if its value in d belongs to $A \cup \{\emptyset\}$.

Definition 7. *1) $d_1 \in NDVC(V, A)$ is nominatively included in $d_2 \in NDVC(V, A)$, if either $d_1, d_2 \in A$ and $d_1 = d_2$, or $d_1, d_2 \notin A$ and for each terminal path $(v_1, v_2, ..., v_n)$ in d_1 there is a terminal path $(v'_1, v'_2, ..., v'_m)$ in d_2 such that $v_1v_2...v_n = v'_1v'_2...v'_m$ and the values of $(v_1, v_2, ..., v_n)$ in d_1 and $(v'_1, v'_2, ..., v'_m)$ in d_2 coincide.*

2) d_1, d_2 are nominative equivalent ($d_1 \approx d_2$), if d_1 is nominatively included in d_2 and d_2 is nominatively included in d_1.

5 Associative Nominative Glushkov Algorithmic Algebra

Let V and A be fixed sets of basic names and values. Denote

$Pr_{CC}(V,A) = NDVC(V,A) \tilde{\to} \{T,F\}$,

$Fn_{CC}(V,A) = NDVC(V,A) \tilde{\to} NDVC(V,A)$.

We will assume that T and F do not belong to $NDVC(V,A)$.

We will call the elements of $Pr_{CC}(V,A)$ (*partial nominative*) *predicates* and the elements of $Fn_{CC}(V,A)$ *(partial binominative) functions.*

Let us denote by $\bar{U}$ the set of all tuples $(u_1,u_2,...,u_n)$, $n \geq 1$ of complex names from V^+ such that whenever $i \neq j$, u_i and u_j are incomparable in the sense of the prefix relation.

- Sequential composition of functions (denoted using the infix notation) $\bullet : Fn(V,A) \times Fn(V,A) \to Fn(V,A)$ is defined as follows: for all $f,g \in Fn(V,A)$ and data d: $(f \bullet g)(d) \cong g(f(d))$.

- Prediction composition [15] $\cdot : Fn(V,A) \times Pr(V,A) \to Pr(V,A)$ is defined as follows: for all $f \in Fn(V,A)$, $p \in Pr(V,A)$, and data d: $(f \cdot p)(d) \cong p(f(d))$.

- Assignment composition $Asg^u : Fn(V,A) \to Fn(V,A)$ with a parameter $u \in V^+$ is defined as follows: for each $f \in Fn(V,A)$ and data d, $(As^u(f))(d) \cong d\nabla_a^u f(d)$.

- The composition of superposition into a function

$$S_F^{u_1,u_2,...,u_n} : Fn(V,A) \times (Fn(V,A))^n \to Fn(V,A)$$

with parameters $n \geq 1$ and $u_1,...,u_n \in V^+$ such that $(u_1,...,u_n) \in \bar{U}$ is defined as follows:

$$S_F^{u_1,...,u_n}(f,f_1,...,f_n)(d) \cong f(...(d\nabla_a^{u_1} f_1(d))...\nabla_a^{u_n} f_n(d))...).$$

We will also use the following notation for this composition: for each tuple $\bar{u} = (u_1,u_2,...,u_n) \in \bar{U}$, $S_F^{\bar{u}}$ denotes $S_F^{u_1,u_2,...,u_n}$.

- The composition of superposition into a predicate

$$S_P^{u_1,u_2,...,u_n} : Pr(V,A) \times (Fn(V,A))^n \to Pr(V,A)$$

with parameters $n \geq 1$ and $u_1, ..., u_n \in V^+$ such that $(u_1, ..., u_n) \in \bar{U}$ is defined as follows:

$$S_P^{u_1,...,u_n}(p, f_1, ..., f_n)(d) \cong p(...(d\nabla_a^{u_1} f_1(d))...\nabla_a^{u_n} f_n(d))...).$$

We will also use the following notation for this composition: for each tuple $\bar{u} = (u_1, u_2, ..., u_n) \in \bar{U}$, $S_P^{\bar{u}}$ denotes $S_P^{u_1,u_2,...,u_n}$.

- Branching composition $IF : Pr(V,A) \times Fn(V,A) \times Fn(V,A) \to Fn(V,A)$ is defined: for each $p \in Pr(V,A)$, $f, g \in Fn(V,A)$:

$IF(p,f,g)(d) \cong f(d)$, if $p(d) \downarrow = T$.

$IF(p,f,g)(d) \cong g(d)$, if $p(d) \downarrow = F$.

$IF(p,f,g)(d)$ undefined, if $p(d) \uparrow$.

- Cycle composition $WH : Pr(V,A) \times Fn(V,A) \to Fn(V,A)$ is defined as follows: for each $p \in Pr(V,A)$, $f \in Fn(V,A)$, and d:

$WH(p,f)(d) \downarrow = f^{(n)}(d)$, if there exists $n \geq 0$ such that $(f^{(i)} \cdot p)(d) \downarrow = T$ for all $i \in \{0, 1, ..., n-1\}$ and $(f^{(n)} \cdot p)(d) \downarrow = F$, where $f^{(n)}$ is a n-times sequential composition of f with itself ($f^{(0)}$ is the identity function), and $WH(p,f)(d)$ is undefined otherwise.

- Negation $\neg : Pr(V,A) \to Pr(V,A)$ is a composition such that for each $p \in Pr(V,A)$ and data d: $(\neg p)(d) \cong T$, if $p(d) \downarrow = F$; $(\neg p)(d) \cong F$, if $p(d) \downarrow = T$; $(\neg p)(d)$ is undefined, if $p(d) \uparrow$.

- Disjunction $\vee : Pr(V,A) \times Pr(V,A) \to Pr(V,A)$ is a composition defined as follows: for each $p_1, p_2 \in Pr(V,A)$ and data d:

$$(p_1 \vee p_2)(d) \cong \begin{cases} T, & \text{if } p_1(d) \downarrow = T \text{ or } p_2(d) \downarrow = T; \\ F, & \text{if } p_1(d) \downarrow = F \text{ and } p_2(d) \downarrow = F; \\ \text{undefined}, & \text{otherwise.} \end{cases}$$

- Identity composition $Id : Fn(V,A) \to Fn(V,A)$ is defined as follows: $Id(f) = f$ for all $f \in Fn(V,A)$.
- True constant predicate (null-ary composition) $True \in Pr(V,A)$ is defined as follows: $True(d) \downarrow = T$ for all data d.
- Bottom function (null-ary composition) $\bot_F \in Fn(V,A)$ is defined as follows: $\bot_F\ (d) \uparrow$ for all data d.
- Bottom predicate (null-ary composition) $\bot_P \in Pr(V,A)$ is defined as follows: $\bot_P\ (d) \uparrow$ for all data d.
- Name checking predicate (null-ary composition) with a parameter $u \in V^+$: $u!(d) = T$, if $u \Rightarrow_a (d) \downarrow$; $u!(d) = F$, if $u \Rightarrow_a (d) \uparrow$.
- Empty constant function (null-ary composition): $Empty(d) = \emptyset$.
- Emptiness checking predicate (null-ary composition): $IsEmpty(d) = T$, if $d = \emptyset$; $IsEmpty(d) = F$, if $d \neq \emptyset$.

These compositions allow us to specify a rather expressive program language – a generalization of Glushkov Algorithmic Algebras.

Definition 8. *An Associative Nominative Glushkov Algorithmic Algebra (ANGAAA) is a two-sorted algebra*

$$NGA^a_{CC}(V,A) = (Pr_{CC}(V,A), Fn_{CC}(V,A); \bullet, IF, WH, \cdot, \{Asg^u\}_{u\in V^+},$$

$$\{S^{\bar{u}}_F\}_{\bar{u}\in\bar{U}}, \{S^{\bar{u}}_P\}_{\bar{u}\in\bar{U}}, \vee, \neg, Id, True, \bot_F, \bot_P, \{u!\}_{u\in V^+}, Empty, IsEmpty)$$

6 Generalization of eds definability

Let $V = \{\bar{v}_1, \bar{v}_2, ..., \bar{v}_m\}$ be a fixed finite set of basic names and A be a fixed set of basic values. If $x_1, ..., x_n$ are variable names, denote by $T_{x_1,x_2,...,x_n}(V)$ the set of all terms in $NDAS_{CC}(V,A)$ in $x_1, ..., x_n$, and by $\Phi_{x_1,...,x_n}(V)$ the set of all basic semalgebraic conditions, i.e. formulas which have a form of a finite conjunction of atomic formulas in $NDAS_{CC}(V,A)$ or their negations (note that equality is not allowed).

For each term t in $T_{x_1,x_2,...,x_n}(V)$ or formula φ in $\Phi_{x_1,...,x_n}(V)$, denote by $[t]$ and $[\varphi]$ their standard interpretations (i.e. the corresponding partial function and predicate on tuples of elements of $NDVC(V,A)$).

Definition 9. *A function $f \in Fn_{CC}(V,A)$ is eds definable, if there exists a natural number n, data $d_1, d_2, ..., d_n \in NDVC(V,A)$, and a finite or countable set S of pairs of the form*

$\{(\varphi_i(x, x_1, x_2, ..., x_n), t_i(x, x_1, ..., x_n)) \mid i \in I\}$

(I is a set of indices $I = \mathbb{N}$ or $I = \{1, 2, ..., k\}$ for some natural k), where $\varphi(x, x_1, ..., x_n) \in \Phi_{x,x_1,...,x_n}(V)$ and $t(x, x_1, ..., x_n) \in T_{x,x_1,...,x_n}(V)$ and $x, x_1, ..., x_n$ are different variable names, such that

1) the set S' of all strings of the form $\varphi(x, x_1, ..., x_n) \to t(x, x_1, ..., x_n)$ for $(\varphi(x, x_1, ..., x_n), t(x, x_1, ..., x_n)) \in S$ in the alphabet $\{\bar{v}_1, \bar{v}_2, ..., \bar{v}_m, ,, (,), x, x_1, ..., x_n, \emptyset, \Rightarrow, a, !, IsEmpty, \neg, \wedge\}$ is recursively enumerable (it is assumed that symbols with sub/superscripts ∇_a^v in terms are represented as ∇av in the elements of S').

2) For each $i \in I$ and $d \in NDVC(V,A)$, if $[\varphi_i](d, d_1, ..., d_n) \downarrow = T$, then $f(d) \cong [t_i](d, d_1, ..., d_n)$.

3) if $[\varphi_i](d, d_1, ..., d_n) \uparrow$ for all $i \in I$, then $f(d) \uparrow$.

Definition 10. *A predicate $p \in Pr_{CC}(V,A)$ is eds definable, if there exists a natural number n, data $d_1, d_2, ..., d_n \in NDVC(V,A)$, and a finite or countable set S of pairs of the form*

$\{(\varphi_i(x, x_1, x_2, ..., x_n), b_i) \mid i \in I\}$

(I is a set of indices $I = \mathbb{N}$ or $I = \{1, 2, ..., k\}$ for some natural k), where $\varphi(x, x_1, ..., x_n) \in \Phi_{x,x_1,...,x_n}(V)$ and $b_i \in \{T, F\}$ and $x, x_1, ..., x_n$ are different variable names, such that

1) the set S' of all strings of the form $\varphi(x, x_1, ..., x_n) \to b$ for $(\varphi(x, x_1, ..., x_n), b) \in S$ in the alphabet $\{\bar{v}_1, \bar{v}_2, ..., \bar{v}_m, ,, (,), x, x_1, ..., x_n, \emptyset, \Rightarrow, a, !, IsEmpty, \neg, \wedge\}$ is recursively enumerable (it is assumed that symbols with superscripts ∇_a^v in terms are represented as ∇av in S').

2) For each $i \in I$ and d, if $[\varphi_i](d, d_1, ..., d_n) \downarrow = T$, then $f(d) \cong b$.

3) if $[\varphi_i](d, d_1, ..., d_n) \uparrow$ for all $i \in I$, then $f(d) \uparrow$.

7 Main results

Let us introduce the following notation.

- $PrEds_{CC}(V, A)$ is the set of eds definable predicates in $Pr_{CC}(V, A)$
- $FnEds_{CC}(V, A)$ is the set of eds definable functions in $Fn_{CC}(V, A)$

Theorem 1 (eds definability of programs of ANGAA)**.** *$PrEds_{CC}(V, A)$ and $FnEds_{CC}(V, A)$ form a subalgebra of $NGA^a_{CC}(V, A)$.*

For any preorder $\leqslant$ on $NDVC(V, A)$ let us denote:

- $PrM_{CC}(V, A, \leqslant)$ is the set of all $p \in Pr_{CC}(V, A)$ such that for all d_1, d_2, if $p(d_1)\downarrow$ and $d_1 \leqslant d_2$, then $p(d_2)\downarrow$ and $p(d_1) = p(d_2)$.
- $FnM_{CC}(V, A, \leqslant)$ is the set of all $f \in Fn_{CC}(V, A)$ such that for each d_1, if $f(d_1)\downarrow$ and $d_1 \leqslant d_2$, then $f(d_2)\downarrow$ and $f(d_1) \leqslant f(d_2)$.
- $PrM^n_{CC}(V, A, \leqslant) = PrM_{CC}(V, A, \leqslant)$ for each $n \in \mathbb{N}$, where $\leqslant^n$ is the product order on $(NDVC(V, A))^n$ induced by $\leqslant$.
- $FnM^n_{CC}(V, A, \leqslant) = FnM^n_{CC}(V, A, \leqslant)$ for each $n \in \mathbb{N}$, where $\leqslant^n$ is the product order on $(NDVC(V, A))^n$ induced by $\leqslant$.

Definition 11. *Elements of $Fn_{CC}(V, A, \approx)$ (where $\approx$ is nominative stability on $NDVC(V, A)$) are called nominative stable functions.*

Theorem 2. *Let $\leqslant$ be a preorder on $NDVC(V, A)$. Assume that:*
(1) $\Rightarrow u \in FnM_{CC}(V, A, \leqslant)$ for each $u \in V^+$;
(2) $u \Rightarrow_a \in FnM_{CC}(V, A, \leqslant)$ for each $u \in V^+$;
(3) $\nabla^u_a \in FnM^2_{CC}(V, A, \leqslant)$ for each $u \in V^+$;
(4) $u! \in PrM_{CC}(V, A, \leqslant)$ for each $u \in V^+$;
(5) $IsEmpty \in PrM_{CC}(V, A, \leqslant)$.
Then $PrEds_{CC}(V, A) \subseteq PrM_{CC}(V, A, \leqslant)$ and $FnEds_{CC}(V, A) \subseteq FnM_{CC}(V, A, \leqslant)$.

Corollary 1. *Under the conditions of the theorem, all functions (programs) expressible in $NGA^a_{CC}(V, A)$ belong to $PrM_{CC}(V, A, \leqslant)$.*

Corollary 2. *All functions (programs) expressible in* $NGA^a_{CC}(V, A)$ *are nominative stable (since nominative equivalence is an equivalence on* $NDVC(V, A)$ *and the conditions of the theorem hold for it).*

8 Conclusions

The process of software system development is related with such transformations of programs that preserve the main requirements to software systems. In this paper we have investigated the following problem: characterize the class of programs stable under natural data structures transformations. To do this we have presented the formal program models using composition-nominative approach. According to this approach data structures were considered as nominative data structures and semantics of programs was presented by special program algebras with operations called compositions. We have defined various data transformations and specified a very general class of compositions based on H. Friedman effective definitional scheme. We have proved that this class preserves program stability under natural data structures transformations. The obtained results can be useful in software development and verification.

References

[1] H. Friedman. *Algorithmic Procedures, Generalized Turing Algorithms, and Elementary Recursion Theory.* Studies in Logic and the Foundations of Mathematics, Volume 61, 1971, pp. 361-389.

[2] H. Friedman, R. Mansfield. *Algorithmic Procedures.* Transactions of the AMS, Volume 332, Number 1, 1992, pp. 297–312.

[3] H. R. Nielson, F. Nielson. *Semantics with Applications: A Formal Introduction.* John Wiley & Sons, Inc., New York, NY, USA, 1992.

[4] R.W. Floyd. Assigning meanings to programs. In *Proceedings of the American Mathematical Society Symposia on Applied Mathematics*, volume 19, pp. 19–31, 1967.

[5] C.A.R. Hoare. An axiomatic basis for computer programming. *Communications of the ACM*, (12):576–580, 583, 1969.

[6] N.S. Nikitchenko. A composition-nominative approach to program semantics. Technical report, IT-TR 1998-020, Technical University of Denmark, 1998.

[7] M.S. Nikitchenko, S.S. Shkilniak. *Mathematical logic and theory of algorithms.* Publishing house of Taras Shevchenko National University of Kyiv, in Ukrainian, 2008.

[8] A. Kryvolap, M. Nikitchenko, W. Schreiner. Extending Floyd-Hoare logic for partial pre- and postconditions. *In: V. Ermolayev et al. (eds.): ICTERI 2013, CCIS*, 412:355–378, 2013.

[9] M. Nikitchenko, Ie. Ivanov. Programming with nominative data. In *Proceedings of CSE'2010 International Scientific Conference on Computer Science and Engineering, September 20-22, 2010, Kosice, Slovakia*, pp. 30–39, 2010.

[10] M.S. Nikitchenko, Ie. Ivanov. Stability and monotonicity of programs with respect to structure transformations of data. *Problems in programming*, No. 2-3:58–67, 2010.

[11] M.S. Nikitchenko, Ie. Ivanov. Composition-nominative languages of programs with associative denaming. *Visnyk (Bulletin) of the Lviv University. Ser. Appl. Math. Inform.*, 16:124–139, 2010.

[12] V.G. Skobelev, M. Nikitchenko, Ie. Ivanov. *On algebraic properties of nominative data and functions.* Communications in Computer and Information Science (CCIS), Springer International Publishing, Volume 469, pp. 117-138, 2014.

[13] M.S. Nikitchenko, V.G. Tymofieiev. Satisfiability in composition-nominative logics. *Central European Journal of Computer Science*, 2:194–213, 2012.

[14] V.G. Skobelev, Ie. Ivanov, M. Nikitchenko. Set-theoretic Analysis of Nominative Data. *Computer Science Journal of Moldova*, No. 3 (69), Vol. 23, 2015, pp. 270-288.

[15] V.M. Glushkov. Automata theory and formal transformations of microprograms. *Cybernetics (in Russian)*, 5:3–10, 1965.

Ivanov Ievgen[1], Mykola Nikitchenko[2], Volodymyr G. Skobelev[3]

[1]Taras Shevchenko National University of Kyiv, Ukraine
Email: ivanov.eugen@gmail.com

[2]Taras Shevchenko National University of Kyiv, Ukraine
Email: nikitchenko@unicyb.kiev.ua

[3]V.M. Glushkov Institute of Cybernetics of NAS of Ukraine
Email: skobelevvg@mail.ru

Expanding a gold collection of images using the Flickr network

Andreea-Alice Laic, Lavinia-Maria Gherasim, Adrian Iftene

Abstract

In last years, multimedia content has grown increasingly over the Internet, especially in social networks, where users often post images using their mobile devices. Currently searching into these networks is primarily made using the title and the keywords associated to resources added by users that have posted the images. The problem we face comes from the fact that many times, title or related keywords are not relevant to the image content. The project presented in this article, has expanded a gold collection of annotated images that will be further processed in order to find similar images based on image content.

Keywords: image retrieval, social networks, human annotations, Flickr network.

1 Introduction

Image retrieval domain is dedicated to systems which deal with browsing, indexing and searching of images in a large context [1]. Typically, this search is done by keywords, metadata and descriptions of images. The volume of data has increased significantly in the latest years, which has led to the development of algorithms performing image processing, the Image Retrieval domain being in a continuous expansion. Big companies like Google, Bing, Yahoo have developed over the time tools and optimized algorithms to be efficient in image search, as proof is the option "Image Search" that they offer.

Content-Based Image Retrieval is preferable because usual keywords search depends on the quality and accuracy of annotations [2]. Also,

methods based on feature selection were used for automatic image annotation [3].

Google proposes a new type of image search, the one based on similar images[1] (images that have similar content, both in color and texture, and the components of the image) of user data. This option is available only in the browser, allowing the user to drag-and-drop an image, enter the URL of the image or make a simple image upload. The advantage of this option (to what is now on the market) comes from the fact that the image database from Google is almost 100,000,000 gigabytes of indexed pages. The disadvantage of this option regarding the programmers is that Google still do not provide an API for application developers.

Similar to what Google offers, TinEye[2] developed a framework that allows the user to perform a reverse search by image. There is a Web application where the user can enter an URL, drag-and-drop or upload an image and get similar results with the image inserted by him. Unlike Google, TinEye offers an API for application developers, but the process of integration into an application development is chargeable.

RevIMG[3] is an image search engine that provides a library for JavaScript and one for Android mobile applications. This engine is intended only to certain image categories like pictures, monuments, famous people, flags, etc.

Besides these applications, there are a series of platforms (Lire[4], pHash[5]) which are able to extract the content items (color, texture, etc.) of the image.

Our project aims to create an expanded collection of annotated images, which will be used to find images similar to a given image and then to use user profiling and image search diversification like in [4, 5].

[1] https://support.google.com/websearch/answer/1325808?hl=en
[2] https://www.tineye.com/
[3] http://www.revimg.net/
[4] http://www.semanticmetadata.net/lire/
[5] http://www.phash.org/

2 System architecture

To develop the proposed project, we have started from a collection of 100 images human annotated with list of keywords [6, 7]. The collection was filtered afterwards by a system created by us in order to keep only the most defining words for each image.

Based on this application, we expanded the collection to 13,000 annotated images using external services, by performing keywords search. The next paragraphs are describing in more detail the developed system.

2.1 Creation of gold collection with annotated images

The initial collection of images consisted of 100 images, from different areas [6, 7]. Images were categorized in the following: 30% images with people, 15% images from nature, 20% images with animals and the remaining images were from various categories (art, furniture, sport, other, etc.).

The images were selected by six human experts and then were manually annotated by human annotators. Some of the images have in their visual content words to see how this can influence the process of annotation. Figure 1 displays how a logged user can annotate an image.

Figure 1. Application interface where users can annotate images

In the section "Ce părere ai despre imagine?" (English: *What is your opinion about image*?), the user can select how much he liked the image shown. We record these opinions in our database, and this allowed us to build profiles for users who have annotated images and to build a recommendation system for them.

In the section "Ce etichete ai asocia imaginii?" (English: *What keywords do you associate to the image*?), the user can indicate a series of English keywords, keywords that he considers suitable for the image. Besides the simple words, they can write also expressions which they consider appropriate for the image.

2.1.1 Experiments

In the process of annotating, there were 28 volunteers in third-year and master students of the Faculty of Computer Science from Iasi. They had to annotate 100 images; the only criterion was to write keywords in English language, criterion that was established from the beginning.

Comparing keywords entered by users for the same picture, we observed that there were small differences between words entered, most of the words were in the same lexical family or were synonymous. Each user was able to annotate as many pictures as he wanted, but in the analysis only keywords entered by 21 users who have annotated all 100 images were selected.

Doing an analysis on what users have annotated on a period of two weeks, we noticed that their tendency was to introduce, on average, 3.41 keywords per image, with a minimum of 2 keywords for an image and a maximum of 12 keywords for an image. Looking further into the keywords that they have entered, we noticed that most users have preferred simple words and not phrases. As a general rule, they have chosen to annotate the content of the image that quickly appears in sight. In the end, the 21 users have entered for the 100 images a total of 1,514 keywords.

For example, for the Figure 2, users have chosen keywords such as "dog, puppy, baby, bed, muster sheet, purity", elements that can be seen easily in the image, and not for keywords like "wood", which can be hardly seen in the background.

Figure 2. One of the images annotated by users

Further, we have implemented an algorithm which, for each image, counts the frequency of lemmas of the keywords associated by users and keeps those with a frequency of at least 4. This frequency was chosen by analyzing the results of the algorithm (each word with its score) and we found that the words with this minimum frequency are the most suitable tags for the image. Besides frequency, we considered the relation of synonymy using WordNet. From all the synonyms, we kept the keyword which appears more often to users who have annotated the image.

For expressions, we divided them into component words, and then calculated the frequency of word components based on lemma and synonymy. If the components of the word had a frequency of occurrence over 4, we decided to keep the expression and give up the words which appeared in the expression. In the end, we considered for every image a list of keywords in descending order of frequency (of course, for frequencies over 4).

In addition to the score calculated for each keyword based on frequency, we decided to calculate a score (in our project is given by formula (1)) for each user who annotated all images in order to see check the credibility of users. Furthermore, towards the way we calculated the score for the keywords, for the user's score we took into account the order of the entered keywords. For example, the user's score was calculated as a product between the number of users who entered that keyword and its quota, given by the formula (1). Thus we could identify the reliable and the less reliable annotators.

$$UserScore = \frac{1}{abs(keywordIndexFinalList - keywordIndexUserList)} \quad (1)$$

Each image contained initially around 30-40 different keywords from all users, and after we applied the algorithm explained above, the number of keywords was reduced to around 3-4 keywords per image. The average remained 3.32 keywords per image, with a minimum of 1 and a maximum of 7. It can be seen that the filtering was done quite rigorous.

After we perform the steps explained above for the image from Figure 2, we left with the following keywords: "dog, child, bed".

2.2 Expanding gold collection of images

2.2.1 Growing from 100 to 13,000 images

This step was about getting the list of keywords for each image and combining them two by two (per image) to get relevant content based on the API provided by Flickr[6]. Our choice to get images only by two keywords has revealed that the images retrieved by the Flickr API were more accurate for the query provided than searching for more or less keywords.

For example, for the image in Figure 3, one of the requests to Flickr will be for the mountain and nature tags. The response is a JSON object, which is parsed, then another request is built to find the address of the image in order to save it into the database.

[6] Flick API: https://www.flickr.com/services/api/flickr.photos.search.html

Figure 3. Image with its corresponding tags

2.2.2 Filtering the newly created collection

To stabilize the collection, it was necessarily to remove the duplicate images resulted from the first step. This was made by trusted human annotators' beings by crossing the images through an interface and removing the duplicates. As it can be seen in Figure 4, every image has a delete button, so the person who is in charge with this operation can remove one of the images when two similar are found.

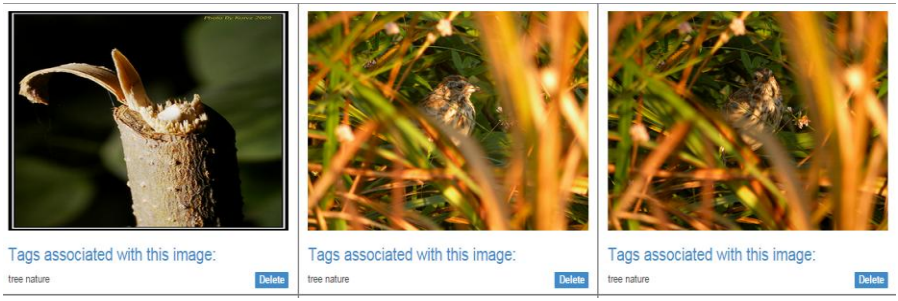


Figure 4. The interface for removing the duplicates

After performing this operation, the collection counts around 13,000 images.

2.2.3 Improving the description of each image

The last step was to increase the list of keywords of each image retrieved by the Flickr API. This was made by human annotators through an interface (see Figure 5) which gave them the possibility to remove the existing keywords and/or add new ones.

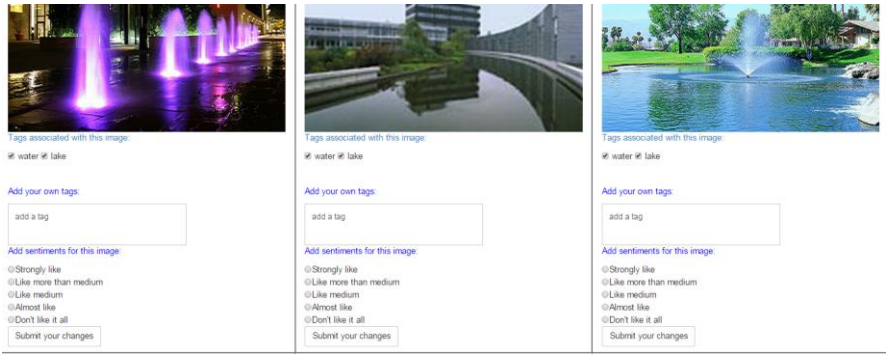


Figure 5. The interface for improving the description of the images

The images are organized in pages, the user gets 100 images per page. He can make the changes right there and then press submit, so the interface is very user-friendly in order to make the annotation process very easy.

2.3 Evaluation

To see how accurate the above system is, we conducted a series of experiments based on human annotators. The metrics we took into consideration were quantity and quality.

2.3.1 Quantitative evaluation

This kind of evaluation refers to the number of keywords added by each annotator. We have observed that a user adds, on average, 3-4 words to describe an image. The keywords are related to the content of the image, but also to the feelings expressed by it.

The number of keywords introduced by the user increases as the image is more complex. This number may come up to 6-7 words in this scenario.

This kind of evaluation is not too relevant for the content of the image because it takes into consideration only the amount of keywords associated, and not the semantic meaning of the image.

2.3.2 Qualitative evaluation

We have seen that the users use similar words for describing the same idea. They tend to focus only on the prominent part.

We have considered that a keyword is relevant for an image when more users used it (or something similar) to annotate the image.

For example, for the image in Figure 2, the common keywords are: "dog, child, bed". As it can be seen, the trend was to use the abstract instead of specific: *child* instead of girl/boy, *dog* instead of boxer (breed of dog).

3 Conclusions

The application presented in this article can be very useful when you need an expanded collection of annotated images to use it in future projects like reverse image search (you have an image and you want to search similar images).

From the evaluation of the created system, we can say that the system works efficiently as long as the users are not influenced one by each other when they perform the annotation process.

Future directions for improving this application are related to the quality of the selected keywords in the filtering process after the annotation.

Acknowledgments. The research presented in this paper was funded by the project MUCKE (Multimedia and User Credibility Knowledge Extraction), number 2, CHIST-ERA/01.10.2012.

References

[1] R. Datta, D. Joshi, J. Li, J. Z. Wang. *Image Retrieval: Ideas, Influences, and Trends of the New Age*. ACM Computing Surveys (CSUR), vol. 40, issue 2, article 5 (2008) pp. 1-60.

[2] J. Eakins, M. Graham. *Content-based Image Retrieval*. Journal Library and Information Briefings, vol. 85 (1999), pp. 1-15.

[3] S. Zhu. *A feature selection method for automatic image annotation*. International Journal of Mathematics and Statistics Invention (IJMSI). E-ISSN: 2321 – 4767 P-ISSN: 2321 – 4759, www.Ijmsi.org, Volume 3, Issue 2, February (2015) pp. 17-22.

[4] C. Şerban, L. Alboaie, A. Iftene. *Image and user profile-based recommendation system*. Workshop on Social Media and the Web of Linked Data (RUMOUR 2015) at EUROLAN 2015 Summer School on Linguistic Linked Open Data. 18 July 2015, Sibiu, Romania. Springer International Publishing Switzerland. D. Trandabăț and D. Gîfu (Eds.): EUROLAN 2015, CCIS 588 (2016) pp. 1-16.

[5] A. Iftene, L. Alboaie. *Diversification in an image retrieval system based on text and image processing*. In Computer Science Journal of Moldova, vol.22, no.3 (66) (2014) pp. 339-348.

[6] A. Laic, A. Iftene. *Automatic Image Annotation*. Proceedings of the 10th International Conference "Linguistic Resources and Tools for Processing the Romanian Language", Craiova, 18-19 September 2014, ISSN 1843-911X (2014), pp. 143-152.

[7] A. Laic, A. Iftene. *Automatic Image Annotation with Romanian Keywords*. BringITon! 2014 Catalogue. ISSN 2285-0929, 14-15 November, Iasi (2014), pp. 46-47.

Andreea-Alice Laic, Lavinia-Maria Gherasim, Adrian Iftene

Institution: "Alexandru Ioan Cuza" University

Address: General Berthelot, No. 16, Iasi, Romania

Phone: 004-0232-2011549

E-mail: andreea.laic@info.uaic.ro, lavinia.gherasim@info.uaic.ro, adiftene@info.uaic.ro

Proceedings of the Conference on Mathematical Foundations of Informatics
MFOI2016, July 25-29, 2016, Chisinau, Republic of Moldova

On linear formats of resolution and paramodulation over ordered clauses

Alexander Lyaletski and Alexandre Lyaletsky

Abstract

Classical first-order logic without and with equality is considered. Certain linear strategies for resolution-type methods over ordered clauses are given. For logic without equality, their soundness and completeness are based on the soundness and completeness of a certain, so-called literal tree calculus. For logic with equality, the strategies admit sound and complete paramodulation extensions when using functional reflexivity axioms.

Keywords: Classical first-order logic, clause, resolution, paramodulation, factorization, strategy, linear format, soundness, completeness, unsatisfiability.

1 Introduction

Modern intelligent systems require the use of methods of efficient inference search in classical first-order logic. As a rule, a preference is given to the resolution approach, first proposed in [1]. In this regard, the resolution methods have been sufficiently studied from the point of view of the construction of their various strategies, which are usually treated as different constraints that should be satisfied in constructing inferences. In particular, the resolution methods over ordered clauses as well-formed expressions has been fairly well studied for example in [2], where the proof of their completeness is based on inferences in the form of a linear sequence of clauses. But in [3], it was shown that the use of the "projections" of tree-like structures in sequential calculi can lead to new resolution strategies.

This paper is devoted to the description of certain linear resolution strategies over ordered clauses and based on the results of [3] concerning literal trees used for refutation search in classical logic without equality. A special feature of the approach proposed here is that the soundness and completeness of these strategies are a consequence of the results obtained in [3] for literal trees.

As for logic with equality, the incorporation of the usual paramodulation in the suggested strategies is made with preserving soundness and completeness if functional reflexivity axioms are used.

2 Preliminaries

In what follows, we give only those notions and definitions that are necessary for an understanding of the paper's content. At that, the notions of a *term, atomic formula, literal, formula*, and *variant* of a formula are presupposed to be known. (The peculiarity of our approach is that *all formulas are quantifier-free in the assumption that all their free variables are universally bound.*)

An *inference* is a sequence of formulas that pairwise have no common variables and each of which is a variant of either a formula belonging to an original set of formulas or a formula obtained from previous formulas by one of given inference rules.

If L is a literal, then $\overline{L}$ denotes its *complementary.*

Usually, a clause is defined as a set of literals and an ordered clause as an ordered set of clauses [2]. Since we focus our attention only on ordered clauses, the following definition of a clause is convenient.

A formula of the form $L_1 \vee \ldots \vee L_n$, where $L_1, \ldots, L_n$ are literals, is called an *ordered clause.*

That is, in the terms of [2] (see also [4]), an ordered clause is an ordered multiset of literals. The *empty clause* (that is a clause containing no literals) is denoted by $\square$.

In what follows, we consider calculi containing certain inference rules that can be applied to ordered clauses in attempt to deduce $\square$ from a given *original se*t IS of *input clauses.*

Let IS be an *original set* of (input) clauses checking on unsatisfiability. If $C \in$ IS and C is an ordered clause $L_1 \vee \ldots \vee L_n$, then the pair $\langle C, i\rangle$ $(1 \leq i \leq n)$ is called *index of* L_i *in* C *w.r.t. IS*.

Note that any applications of any inference rules to clauses of IS, then to their successors and clauses from IS, and so on *preserve indexes of literals belonging to clauses from IS*.

A *substitution, unifier* and *most general unifier (mgu)* are understood in the sense of [1] (see also [2, 4]). If σ is a substitution and Ex an expression, then the result of applying σ to Ex is denoted by $Ex \cdot \sigma$.

As in [3], the resolution strategies under consideration are based on the binary resolution [2] and weak factorization rules [3].

Weak factorization. Let $C_1 \vee L_1 \vee C_2 \vee \ldots \vee C_{n-1} \vee L_n \vee C_n$ be an ordered clause, in which $C_1, \ldots, C_n$ are clauses and $L_1, \ldots, L_n$ literals with the same index. Suppose that there exists the mgu σ of the set $\{L_1, \ldots, L_n\}$. Then the ordered clause $(C_1 \vee L_1 \vee C_2 \vee \ldots \vee C_n) \cdot \sigma$ is deducible from $C_1 \vee L_1 \vee C_2 \vee \ldots C_{n-1} \vee L_n \vee C_n$ according to the *weak factorization rule* that is denoted by WF.

Remark 1. In the case, when the requirement that $L_1, \ldots, L_n$ have the same index is omitted in WF, we obtain the usual factorization rule (see, for example, [2]).

Binary resolution. (This rule has two forms depending on whether there is an input clause among its premises or not.) Let ordered clauses C_1 and C_2 are of the forms $D_1 \vee L$ and $D_2 \vee E$ ($D_2 \vee E \vee D_3$ in the case of the belonging of it to an initial set IS), where D_1, D_2, and D_3 are clauses (possibly, empty) and L and E literals. Suppose that there exists the mgu σ of the set $\{L, \overline{E}\}$. Then we say that the clause $(D_1 \vee D_2) \cdot \sigma$ $((D_1 \vee D_2 \vee D_3) \cdot \sigma)$ is deducible from C_1 and C_2 according to the binary resolution rule that is denoted by RR.

3 Linear resolution with weak factorization

This strategy is a modification of the ordered linear resolution from [2], which uses the RR and WF rules. That is, it is considered that

an inference $C_1, \ldots, C_m (m \geq 1)$ satisfies the *linear resolution strategy with weak factorization* w.r.t. an ordered clause C belonging to an original set IS of clauses if, and only if, C_1 is a variant of C and for each $i = 2, \ldots, m$ one of the following conditions is satisfied:

– C_i is a variant of an input clause from IS,

– C_i is the result of the application of RR to C_{i-1} and D, where D is a variant of an input clause from IS or a variant of a previously deduced clause C_j $(1 \leq j \leq i-1)$,

– C_i is the result of the application of WF to C_{i-1}.

Theorem 1. *(Soundness and completeness of the linear resolution strategy with the weak factorization). Suppose IS is an original set of ordered clauses, $C \in$ IS, and the set IS $\setminus \{C\}$ is satisfiable in classical first-order logic without equality. The set IS is unsatisfiable in classical first-order logic without equality if, and only if, there exists an inference of $\square$ w.r.t. C from IS satisfying the linear resolution strategy with the weak factorization.*

Proof. This theorem is an obvious corollary of Prop. 3 from [3]. $\square$

Example 1. Suppose IS $= \{A^{(1,1)} \vee A^{(1,2)}, \neg A^{(2,1)} \vee \neg A^{(2,2)}\}$, where A is an atomic formula, $\neg$ is the negation symbol, and the pair (i, j) indicates that j is an index of the jth occurrence of a literal (A or $\neg A$ in our case) in the ith clause from IS. (Note that this linear resolution strategy presupposes that the same literal with the different upper pairs presents different literals.) Then we can construct the following linear inference of $\square$ with the help of RR and WF:

(1) $A^{(1,1)} \vee A^{(1,2)}$ $(\in$ IS$)$
(2) $\neg A^{(2,1)} \vee \neg A^{(2,2)}$ $(\in$ IS$)$
(3) $A^{(1,1)} \vee \neg A^{(2,2)}$ (from (2) and (1) by RR)
(4) $A^{(1,1)} \vee A^{(1,1)}$ (from (3) and (1) by RR)
(5) $A^{(1,1)}$ (from (4) by WF)
(6) $\neg A^{(2,1)}$ (from (5) and (2) by RR)
(7) $\square$ (from (6) and (5) by RR)

Therefore, IS is an unsatisfiable set of ordered clauses.

Remark 2. The just-given example shows that in the case of the replacement of the usual factorization by the weak factorization rule, in some cases it is impossible to avoid the appearance of tautologies (i.e. clauses containing a literal and its complementary, such as, for example, the clause (3)) in attempting to construct an inference of $\square$ even in the case of applying the binary resolution rule without any restrictions.

Corollary. The usual ordered linear resolution [2] is a sound and complete method.

Remark 3. In the case of the usual ordered linear resolution (see, for example, [2] or [4]) we can construct the following inference for the above-given example of IS:

(1) $A^{(1,1)} \vee A^{(1,2)}$ ($\in$ IS)
(2) $\neg A^{(2,1)} \vee \neg A^{(2,2)}$ ($\in$ IS)
(3) $A^{(1,1)}$ (from (1) by the usual factorization)
(4) $\neg A^{(2,1)}$ (from (2) by the usual factorization)
(5) $\square$ (from (4) and (3) by RR)

Draw your attention to the fact that the usual linear resolution avoids producing tautologies. That is why this example demonstrates that using the usual linear resolution, we should allow applying the factorization to any earlier deduced clause *not being mandatory* by an immediate predecessor to any factorization rule application as this requires our line format with the weak factorization.

4 Linear resolution with weak factorization and quasi-subsumption

Attempt to combine RR and WF rules in the form of one rule leads to a subsumption strategy.

A cause C is called an *initial subclause* of a clause D if, and only if, D is of the form $C \vee D'$, where D' is a clause. (This definition takes into account that *graphically equal literals with different indexes are*

considered different.)

Quasi-subsumption rule. Let clauses D_1 and D_2 be of the form $C_1 \vee L$ and $C_2 \vee E$, where L and E are literals and C_1 and C_2 are clauses, at that, there exists the mgu σ of the set $\{L, \overline{E}\}$. Suppose that there exists such a substitution θ that $(C_1' \vee C_2') \cdot \theta$ is derived from $(C_1 \vee C_2) \cdot \sigma$ by several applications of WF and that $C_1' \cdot \theta$ is an initial subclause of $C_2' \cdot \theta$. Then $C_1' \cdot \theta$ is deducible from D_1 and D_2 by the *quasi-subsumption rule* that is denoted by RF.

Remark 4. It is clear that θ is the simultaneous mgu of certain sets, every of which contains only literals with the same index. This feature permits to "rewrite" the RF rule only in terms of the resolution and unification (without involving the notion of weak factorization).

We say that an inference $C_1, \ldots, C_m$ $(m \geq 1)$ is constructed in accordance with *linear resolution strategy with weak factorization and quasi-subsumption* w.r.t. C if, and only if, C_1 is a variant of C and the following conditions are satisfied for each i $(2 \leq i \leq m)$:

– C_i is a variant of an input clause from IS,

– C_i is the result of the application of RR to C_{i-1} and D, where D mandatorily is a variant of an input clause from IS,

– C_i is deduced by the application of RF rule to C_{i-1} and D, where D is a variant of a previously deduced clause C_j distinquished from any variant of any input clause from IS $(1 \leq j \leq i-1)$,

– C_i is the result of the application of WF to C_{i-1}.

Theorem 2. *(Soundness and completeness of linear resolution with weak factorization and quasi-subsumption). Suppose IS is an original set of ordered clauses, $C \in$ IS, and the set IS $\setminus \{C\}$ is satisfiable in classical first-order logic without equality. The set IS is unsatisfiable in classical first-order logic without equality if, and only if, there exists an inference of $\square$ w.r.t. C from IS satisfying the linear resolution strategy with weak factorization and quasi-subsumption.*

Proof. If we turn to the literal tree calculation LC from [3], then using its properties as it was done in the proof of Proposition from [3], that

is, analyzing literal trees Tr in inferences of the "degenerative" tree $\triangle$ and then passing to the clause images $\lambda(Tr)$ of Tr, we reach the required result. $\square$

Remark 5. For propositional logic we have that $C_1' \cdot \theta$ from the definition of RW is an (initial) subclause [2] of D_1 and, therefore, D_1 can do not participate in the subsequent inference search of $\square$.

Example 2. Suppose IS = $\{A^{(1,1)} \vee A^{(1,2)}, \neg B^{(2,1)} \vee \neg A^{(2,2)}, B^{(3,1)} \vee \neg A^{(3,2)}\}$, where A and B are atomic formulas. Then we can construct the following linear inference of $\square$ in the linear resolution strategy with the weak factorization and quasi-subsumption:

(1) $A^{(1,1)} \vee A^{(1,2)}$ ($\in$ IS)
(2) $\neg B^{(2,1)} \vee \neg A^{(2,2)}$ ($\in$ IS)
(3) $\neg A^{(3,2)} \vee B^{(3,1)}$ ($\in$ IS)
(4) $\neg A^{(3,2)} \vee \neg A^{(2,2)}$ (from (3) and (2) by RR)
(5) $\neg A^{(3,2)} \vee A^{(1,1)}$ (from (4) and (1) by RR)
(6) $\neg A^{(3,2)}$ (from (5) and (4) by RF)
(7) $A^{(1,1)}$ (from (6) and (1) by RR)
(8) $\square$ (from (7) and (6) by RR)

Therefore, IS is an unsatisfiable set of ordered clauses.

As in the case with the linear resolution with weak unification, the application of this strategy in some cases *leads to the necessity of generating tautology* for providing completeness (the clause (5)). Also note that after the construction of $\neg A^{(3,2)}$ on the step (6) it is possible to block (due to subsumption) the using of clauses (3), (4), and (5) and in this case the clause (2) can do not be also taken into account because the complementary of the literal $\neg B^{(2,1)}$ will be absent.

5 Paramodulation extensions of linear strategies

For handling equality in the case of classical logic with equality, we use the paramodulation rule in the following form (cf [2]).

If an ordered clause C contains a term t, then $C[t]$ denotes a single (selected and fixed) occurrence of t in C.

Paramodulation rule. If a term t has an occurrence in an ordered clause C_1 (that is $C_1[t]$ is a clause with a selected and fixed occurrence of t), an ordered clause C_2 is of the form $C_2' \vee t = s \vee C_2''$ or $C_2' \vee s = t \vee C_2''$, and there exists the mgu σ of the set $\{s, t\}$, then the clause $(C_2' \cdot \sigma \vee C_2'' \cdot \sigma \vee C_1[s]) \cdot \sigma$ is deducible from C_1 and C_2 by the *paramodulation rule* (denoted by PP) in the direction from C_2 to C_1 ($C_1[s]$ is the result of the replacement of the selected and fixed term t by the term s).

We obtain two *paramodulation extensions* of the introduced linear strategies by simple adding the PP rule to them that satisfies the following restriction on its application: one of its premises should be a variant either of an input clause or earlier deduced clause and the other obligatorily should be a variant of an immediately deduced clause.

Denote the paramodulation extension of the linear resolution with weak factorization by RR+WF+PP and the paramodulation extension of the linear resolution with weak factorization and quasi-subsumption by RR+WF+RF+PP.

An expression of the form $f(x_1, \ldots, x_k) = f(x_1, \ldots, x_k)$, where f is a k-arity functional symbol and $x_1, \ldots, x_k$ are variables, is called a *functionally reflexive axiom.*

If IS is an original set of ordered clauses, then Rf(IS) denotes the set of functionally reflexive axioms for all the functional symbols from IS.

Theorem 3. *(Soundness and completeness of paramodulation extensions of linear strategies). Suppose IS is an original set of ordered clauses, $C \in$ IS, and the set IS $\setminus \{C\}$ is satisfiable in classical first-order logic with equality. The set IS is unsatisfiable in classical first-order logic with equality if, and only if, there exists an inference of $\square$ w.r.t. C from IS $\cup$ Rf(IS) $\cup \{x = x\}$ (x is a variable) satisfying the linear (RR+WF+PP)-strategy (the linear (RR+WF+RF+PP)-strategy).*

Proof. The proof of this theorem can be obtained in the same way that

was applied in [2] for proving the soundness and completeness of the (usual) linear paramodulation. □

To demonstrate some of the peculiarities of inference search satisfying both (RR+WF+PP)- and (RR+WF+RF+PP)-strategies, let us consider the original set IS = $\{\neg R_1^{(1,1)}(f(a), f(f(a))),\ R_2^{(2,1)}(a,a) \vee R_1^{(2,2)}(f(a), f(a)),\ \neg R_2^{(3,1)}(f(f(a)), a),\ \neg R_1^{(4,1)}(f(a), f(a)),\ R_1^{(5,1)}(f(a), f(f(y))) \vee y = f(y)\}$, where a is a constant, x and y variables, f a functional symbol, and R_1 and R_2 predicate symbols. Construct an inference of □ from this IS (w.r.t. $R_1(a, f(x)) \vee x = f(x)$) satisfying the (RR+WF+PP)-strategy.

(1) $\neg R_1^{(1,1)}(f(a), f(f(a)))$ $(\in \text{IS})$
(2) $R_2^{(2,1)}(a,a) \vee R_1^{(2,2)}(f(a), f(a))$ $(\in \text{IS})$
(3) $\neg R_2^{(3,1)}(f(f(a)), a)$ $(\in \text{IS})$
(4) $\neg R_1^{(4,1)}(f(a), f(a))$ $(\in \text{IS})$
(5) $R_1^{(5,1)}(f(a), f(f(y))) \vee y = f(y)$ $(\in \text{IS})$
(6) $R_1^{(5,1)}(f(a), f(f(a))) \vee R_2^{(2,1)}(f(a), a) \vee R_1^{(2,2)}(f(a), f(a))$ (from (5) to (2) by PP; here a is substituted for y)
(7) $R_1^{(5,1)}(f(a), f(f(a))) \vee R_1^{(5,1)}(f(a), f(f(a))) \vee R_2^{(2,1)}(f(f(a)), a) \vee R_1^{(2,2)}(f(a), f(a))$ (from (5) to (6) by PP; here a is substituted for y)
(8) $R_1^{(5,1)}(f(a), f(f(a))) \vee R_1^{(5,1)}(f(a), f(f(a))) \vee R_2^{(2,1)}(f(f(a)), a)$ (from (8) and (4) by RR)
(9) $R_1^{(5,1)}(f(a), f(f(a))) \vee R_2^{(2,1)}(f(f(a)), a)$ (from (7) by WF)
(10) $R_1^{(5,1)}(f(a), f(f(a)))$ (from (9) and (3) by RR)
(11) □ (from (10) and (1) by RR)

Therefore, IS is an unsatisfiable set in classical first-order logic with equality.

We can transform this inference into an inference satisfying the (RR+WF+RF+PP)-strategy by the replacement of the applications of RR at the step (8) and WF at the step (9) by the single application of RF producing the same clause that is at the step (10).

The the just-given inference does not contain any functional reflexivity axiom. But the presence of Rf(IS) in the wording of Theorem 3 is necessary for providing the completeness of the introduced linear paramodulation extensions (RR+WF+PP) and (RR+WF+RF+PP) in the general case.

In order to make sure of this, it is enough to consider the set $\{a = b, R_1(h_1(z,z)),\ R_2(h_2(u,u)),\ g_1(f(a),f(b)) = h_1(f(a),f(b)),$ $g_2(f(a),f(b)) = h_2(f(a),f(b)),\ \neg R_1(g_1(x,x)) \vee \neg R_2(g_2(y,y))\}$, where a and b are constant, x, y, z, u variables, R_1 and R_2 predicate symbols, and f, g_1, g_2, h_1, and h_2 functional symbols, the unsatisfiability of which in classical logic with equality was proved in [3]. Any attempt to construct an inference of $\Box$ from this set without functional reflexivity axioms will be unsuccessful (the proof of this is omitted here).

6 Conclusion

The proposed strategies emerged from an analysis of inference search in the literal trees calculi [3] with subsequent passing to the clause images of literal trees. In this regard, the soundness and completeness of the strategies for classical logic without equality are simple consequences of the soundness and completeness of a certain literal trees calculus while their direct proof would require considerable efforts, not to mention the wording of the strategies themselves. Apparently, this can be explained by the fact that the tree-like structures give more possibilities for organizing different ways for making inference searching than their linear analogues.

As to the paramodulation, unfortunately, its direct incorporation requires using functional reflexivity axioms for providing the completeness of the proposed paramodulation extensions for classical logic with equality. But it can be expected that further research in this direction will lead to such paramodulation extensions that will be complete without using functional reflexivity axioms. In order to reach this, one can try to use, for example, the ideas proposed in [5], transforming respectively the paramodulation extensions considered in the paper.

References

[1] J.A. Robinson. *A machine-oriented logic based on resolution principle*. Journal of the ACM, 12(1), 1965, pp. 23–41.

[2] Chang C. and Lee R. *Symbolic logic and mechanical theorem proving*. Academic Press Inc., Orlando, FL, USA, 1997, 332 pp.

[3] Lyaletski A., Letichevky A., and Kalinovskyy O. *Literal trees and resolution technique*. Intelligent Information Processing and Web Mining: Proceedings of the International IIS: IIPWM05 Conference (Gdansk, Poland, June 2005), Springer, 2005, pp. 97–106.

[4] Bachmair L. and Ganzinger H. *Resolution theorem proving*, Handbook of Automated Reasoning, Elsevier Science Pub, 2001, pp. 19–99.

[5] Paskevich A. *Connection tableaux with lazy paramodulation*, Journal of Automated Reasoning, Vol. 40, No. 2-3, 2008, pp. 179–194.

Alexander Lyaletski and Alexandre Lyaletsky Received May 28, 2016

Alexander Lyaletski, Alexandre Lyaletsky
Phone: (+38)(044)2293003
E–mails: `forlav@mail.ru`, `foraal@mail.ru`

Proceedings of the Conference on Mathematical Foundations of Informatics
MFOI2016, July 25-29, 2016, Chisinau, Republic of Moldova

Set-theoretic models of the untyped λ-calculus determined by new notions of continuity

Alexandre Lyaletsky

Abstract

The research is devoted to the construction of nontrivial models for untyped λ-calculus according to the Koymans method, which led to the introduction and study of special non-topological notions of a continuity of a function acting on partially ordered sets. The main results obtained in this direction are presented.

Keywords: Untyped λ-calculus, λ-model, Koymans method, continuity of a function.

1 Introduction

Recall that in early 80's, K. Koymans and others developed a general method for constructing λ-models. Without going into details, just remind the main Koymans result [1] stating that up to an isomorphism, each λ-model can be constructed by means of this method from an appropriate cartesian closed category with a so-called reflexive object. Also note the result of E. Engeler, D. Scott, F. Honsell, and others, that states that each groupoid can isomorphically be embedded into some (extensional) λ-model; it permits to see the class of all λ-models as very "wide" and very "various".

On the other hand, all known "continuous" λ-models were (or may be viewed as) constructed by means of the Koymans method on the basis of an appropriate notion of a continuity of a function, where the continuity is determined by the underlining topologies which do not satisfy even the T_1 separation axiom; hence, on the (extended) real

line $\mathbb{R}$, each of these topological notions of continuity is not equivalent to the usual notion of continuity.

In this connection, the following natural question arises: *whether a natural notion of a continuity of a function can be introduced in such a way that, first, it is equivalent to the usual notion of a continuity of a function on the (extended) real line $\mathbb{R}$, and, second, it admits applying the Koymans method and, as a result, determines nontrivial λ-models?*

This investigation gives a positive answer on the posed question and it is based on a certain notion of continuity that is equivalent to the usual notion of continuity for the unary real functions and is not equivalent to the one for n-ary real functions, when $n \geqslant 2$.

2 Remarks on λ-models

By construction, one can distinguish two sorts of set-theoretic models of (the untyped version of) the theory λ: free λ-models built from λ-terms and syntax-independent λ-models built with the help of usual set-theoretic constructions independent from syntax of λ. Since λ is an equational theory, it permits the construction of its free models in the usual way. However, early attempts to construct a nontrivial syntax-independent lambda-model ran into substantial difficulties, what was mainly caused by the following. The untyped theory λ does not discriminate between functions and (their) arguments; therefore, in order to construct a nontrivial syntax-independent λ-model, it is natural to seek for a nonsingleton set A equipollent to the set A^A of all the unary operations on A. But the famous G.Cantor powerset theorem implies that there does not exist such a set A.

In 1969, this obstacle was overcome by D. Scott [2]. His basic idea is to equip A with some appropriate topology T and to restrict A^A to the set $[A \to A]_T$ of all the unary operations on A continuous w.r.t. T. More precisely, the set A was presupposed to be equipped with the structure of a complete lattice, the topology T was determined by the corresponding partial order relation, and the central Scott result is that every lattice L can be extended to a complete lattice A such that A is

completely isomorphic to $[A \to A]_T$ (where $[A \to A]_T$ is also a lattice w.r.t. pointwise ordering). This complete lattice A is constructed from L as the limit of the direct exponential spectrum $L_0 \to L_1 \to ... \to L_i \to ...$, where $L_0 = L$, $L_{i+1} = [L_i \to L_i]_T$ and each embedding φ_i maps an element $x \in L_i$ to the constant function $x\varphi_i \in L_{i+1}$ such that $(x\varphi_i)(y) = x$ for every $y \in L_i$. At that, after identifying, in a usual way, each of the lattices L_i with the corresponding sublattice of A, the "application" operation $\cdot : A^2 \to A$ ($a_0 \cdot a_1 = (a_0\iota)(a_1)$, where ι sets the isomorphism $A \cong [A \to A]_T$) turns out to be an extension of the application operation $\alpha_i : [L_{i+1} \to L_{i+1}]_T \times L_i \to L_{i+1}$, for every $i \in \mathbb{N}$.

Since then, the original construction of D. Scott was generalized and modified; in particular, there has been constructed several other order-topological λ-models, but they also explore ideas and constructions proposed by D.Scott and incorporate many common features:

1. (a subclass of) the class of partially ordered sets serves as the universe of initial mathematical structures for constructing these order-topological λ-models;

2. they are introduced (or may be viewed) as the limit of either direct, or inverse spectrum of posets;

3. λ-terms are interpreted in these models as continuous operations on their carriers; at that

4. the continuity is introduced as continuity w.r.t. some special monotonic topologies (a topology T on a poset A is called *monotonic* if so is every T-continuous operation on A).

In 1982, K. Koymans developed his method for constructing syntax-free λ-models, which generalizes and unifies, from the point of view of the category theory, the methods of constructing "concrete" syntax-free λ-models. According to this method [1], every cartesian closed (small) category with the so-called reflexive object naturally determines some λ-algebra; moreover, Koymans has shown that up to isomorphism, every λ-algebra (and hence, every λ-model) can be obtained with the help of his method from an appropriate cartesian closed category with a reflexive object. However, for our purposes, another, set-theoretic version of the Koymans method seems to be more convenient. It explores a

semi-formal notion of a "set-theoretic structure" (with a "carrier"), but a reader may keep in mind that the below-given description can completely be formalized with the help of the notion of a strictly concrete category.

Given a class K of set-theoretic structures closed w.r.t. direct products and containing a one-element set and a notion P of P-continuous functions such that for each pair $\langle A_0, A_1\rangle$ of K-structures, the family $[A_0 \to A_1]_P \subseteq {A_1}^{A_0}$ of all P-continuous functions from A_0 into A_1 is a K-structure, at that the following conditions are satisfied:

(1) the composition of any two P-continuous functions is a P-continuous function;

(2) if a function $f : A_0 \times A_1 \to B$ is P-continuous in each of its arguments, then f is P-continuous in both arguments;

(3) the application "operation" $\alpha_A : [A \to A]_P \times A \to A$ is P-continuous for every K-structure A;

(4) K contains a reflexive structure M, i.e. there exist two P-continuous maps, namely $\varphi : [M \to M]_P \to M$ and $\psi : M \to [M \to M]_P$, such that $\varphi \cdot \psi = id_{[M\to M]_P}$, where $id_{[M\to M]_P}$ is the identity map on $[M \to M]_P$.

On a reflexive K-structure M, introduce a binary operation "$\cdot$": we set $m_0 \cdot m_1 = (m_0\varphi)(m_1)$, for every m_0, $m_1 \in M$. Define a λ-interpretation $I : T \times Val(M) \to M$ into groupoid $\langle M, \cdot\rangle$ by induction on the structure of λ-terms; for every valuation ρ into M, we set (where $[t]_\rho$ is the same as $I(t, \rho)$):

a) $[x]_\rho = \rho(x)$ (where x is a variable);

b) $[t_0t_1]_\rho = [t_0]_\rho \cdot [t_1]_\rho$;

c) $[\lambda x.t]_\rho = f_t\psi$, where $f_t \in [M \to M]_P$ is a P-continuous function such that $f_t(m) = [t]_{\rho[x:=m]}$ for every $m \in M$.

Then the following statements hold:

i) the grupoid $\langle M, \cdot\rangle$ is a λ-model;

ii) $\langle M, \cdot\rangle$ is extesional if, and only if, the equality $\psi \cdot \varphi = id_M$ holds.

We also note the earlier-mentioned result (E. Engeler, D. Scott, and some others) stating that each grouppoid can isomorphically be embedded into some (extensional) λ-model. It allows us to see the classes of

all λ-algebras and all λ-models as very "wide" and very "various"; in particular, it suggests that there should exist some notions of continuity, which do not obligatorily satisfy all the stated above conditions (1)-(4), but also lead to the construction of nontrivial λ-models with the help of (an appropriate modification of) the Koymans method.

3 Main results

The carried out research on the construction of nontrivial models for untyped λ-calculus according to the Koymans method required an additional study of the notion of continuity of a function because the one of the necessary conditions for applying the Koymans method is that continuity should satisfy the following property: if a binary function is continuous by each of its arguments, then it should be continuous by its both arguments.

The usual notion of continuity of a real function does not satisfy this requirement (for example, $f(x,y) = (x \cdot y)/(x+y)^2$ is such a function). Hence, every notion coinciding with the usual one for the real functions of the form $f : \mathbb{R}^n \to \mathbb{R}$, $n \geq 1$, does not lead to nontrivial λ-models; in particular, so is the notion of (o)-continuity of a function widely used in the theory of partially ordered linear spaces and measure theory. Therefore, the best one can expect in this situation is to find an appropriate notion of continuity equivalent to the usual notion of continuity for the unary real functions and not equivalent to the one for n-ary real functions for $n > 1$.

Moving in this direction, the corresponding research was performed and following results reflected in [3] were obtained:

• Special non-topological notions of a continuity of a function acting on partially ordered sets, which for the functions of the form $f\colon \overline{\mathbb{R}} \to \overline{\mathbb{R}}$ ($\overline{\mathbb{R}}$ is the extended real line) are equivalent to the usual notion of continuity (but for the functions of the form $f\colon \overline{\mathbb{R}}^n \to \overline{\mathbb{R}}$, $n \geqslant 2$, this is not true), were introduced and studied. An order-theoretic characterization of these notions of continuity was given.

• It was proved that for each of these notions of continuity, there

exists a partially ordered set A such that A is isomorphic, as a partially ordered set, to the pointwise ordered set of all continuous operations acting on A.

• With the help of this result and an appropriate order-theoretic characterization of one of these new notions of continuity, namely (θ)-continuity, it was shown that *the notion of a (θ)-continuous function leads to the construction, by means of the Koymans method, of some new nontrivial λ-models.*

4 Conclusion

The presented research and results give a way for the construction of new models for untyped λ-calculus on the basis of the notion of continuity and Koymans method. Additionally, the author hopes that the introduced notions of continuity of a function will take attention of specialists in functional analysis and other mathematical disciplines.

References

[1] K. Koymans. *Models of the lambda calculus.* Information and Control, vol. 52 (1982), pp. 306-332.

[2] D. Scott. *Models for the λ-calculus.* Manuscript, draft, Oxford, 1969.

[3] A. Lyaletsky. *Continuity of a function in intensional models of lambda-similar calculi.* PhD Thesis, Kiev, Ukraine, 2009, 110 pp. (in Ukrainian).

Alexandre Lyaletsky Received May 28, 2016

Alexandre Lyaletsky
Phone: (+38)(044)2293003
E–mail: foraal@mail.ru

RoDia – project of a regional and historical corpus for Romanian

Cătălina Mărănduc, Ludmila Malahov, Cenel-Augusto Perez, Alexandru Colesnicov

Abstract

The majority of big corpora are in contemporary journalistic style. Parsers work better in the standardized style. But recently the geographic and historic variation of natural languages become in the center of the interest of linguists and computer scientists. We have experienced the variety and creativity of Romanian studying the Social Media communication. The old Romanian has a bigger variety; because it is written before the rules were established, being also non-standardized. We will construct tools for the old Romanian and its south Danube dialects processing. We made a big lexicon of Old Romanian, having about 150,000 inflected forms.

Keywords: linguistic variation, diachronic corpora, non-standardized language, lexicon, inflected forms, parser training.

1 Introduction

In this paper we argued the necessity of the construction of RoDia, a balanced corpus, showing the geographic and the historical variation of the (un)standardized Romanian.

Recruitment Officers predicted that in 10 years the NLP will become a highly sought expertise, and theoretical linguistics without IT support will disappear. But the future began yesterday, when a great linguist argued ahead of computer scientists a wrong theory about the disappearance of supine mode, and they provided her 119,000 examples in Contemporary Romanian. They were unable to combat her theory, because she could argue that they could find 229,000 examples in texts

written 100 years before; they didn't possess a diachronic corpus, but only the one in contemporary Romanian.

A balanced corpus for Romanian, with all the styles and with all the geographic and historical variants is required. The RACAI (Research Academic Institute of Artificial Intelligence) from Bucharest is interested only in Contemporary standardized language. But UAIC-NLP (Natural Language Processing group of Al. I. Cuza University) has a balanced treebank, called UAIC-RoDepTb, having 11,183 sentences and over 205,000 automatically annotated and manually supervised tokens. The treebank contains standardized and non-standardized language. It was a difficult task; we needed over 2,500 sentences for the training of the UAIC POS-tagger and the syntactic parser on Social Media non-standardized texts. This treebank will become the nucleus of the diachronic corpus of Romanian.

The sub-corpora illustrating the regional and the old Romanian texts are at the beginning. We intend to process and supervise sub-corpora from all the regions of the Romanian: Oltenia, Montenia, Moldova on the right and Moldova on the left of the Prut River, Dobrudgea, Transylvania, also from the sixteenth, seventeenth, eighteenth and nineteenth century. The south Danube dialects processing is more difficult, due to the bigger difference toward Romanian standard, but we do not give up, because the Istroromanian dialect is ongoing disappearing and the culture of these populations must be conserved, i. e. preserved for the future.

2 Related Work

Searching for diachronic Corpora, a lot of papers, books [12], projects are found, beginning since the year 2005. There are also two recent conferences: "International Conference on Practical Applications of Language" PALC 23-24 October 2015, Lodz[1], and "Diachronic Corpora, Genre, and Language Change", 8-9 April 2016, Nottingham[2].

[1] http://palc.uni.lodz.pl/

[2] https://www.nottingham.ac.uk/conference/fac-arts/clas/dcglc/home.aspx/

To sum up, there are more diachronic corpora for the Romanic languages, Spanish, Portuguese, Italian, but also for German, English, Japanese, and Polish. In Proceedings of the 12th International Pragmatics Conference in Manchester in 2011 there are chapters based on diachronic pragmatic developments in English, Dutch, Swedish, Italian, Spanish, Finnish, Estonian and Japanese [1].

There exist also some corpora for the dead languages: "The Diachronic Corpus of Sumerian Literature" (DCSL) project seeks to establish a web-based corpus of Sumerian literature spanning the entire history of Mesopotamian civilization, over a range of 2500 years[3].

Diachronic corpus of historical Spanish contains 86 Spanish texts first printed between 1482 and 1647; it covers a representative variety of authors and genres distributed under an open license[4].

Another diachronic corpus of Spanish allows making searches in more than 100 million words in more than 20,000 Spanish texts from the 1200s to the 1900s[5]. Approximately 7% of the words in the corpus have been annotated with their lemma, part of speech, and modern equivalent.

The following site allows making searches in more than 45 million words in almost 57,000 Portuguese texts from the 1300s to the 1900s[6].

The diachronic corpus of Italian is described in [8]. It aims at the construction of a diachronic corpus comprising written Italian texts produced between 1861 and 1945.

Historical English corpora compares poorly with other languages with large, annotated corpora (45-100 million words) that are available, and which have been used to study diachronic syntax in some detail[7].

In Japanese, as an initial step in the development of the Diachronic Corpus at NINJAL, this project carries out basic research on the design of

[3] http://dcsl.orinst.ox.ac.uk/
[4] http://bvmcresearch.cervantesvirtual.com/diasearchtool/
[5] http://www.corpusdelespanol.org/x.asp/
[6] http://www.corpusdoportugues.org/x.asp/
[7] http://corpus.byu.edu/historical-syntax.asp/

a corpus of pre-modern Japanese. Based on representative texts from several periods ranging from ancient times to early modern times, an experimental model of the Diachronic Corpus will be created[8].

Ta-D/DC [5] is a diachronic corpus for German. It uses selected materials from the German Gutenberg Project and enriches them with different linguistic annotation layers, including part-of-speech, lemma, and constituent structure. Linguistic annotation is performed automatically by using statistical tools. In [1], another diachronic German corpus is described. A relational database supplements the XML representation to support sophisticated search and presentation facilities.

Finally, we decided to adhere to PROIEL[9] (Pragmatic Resources in Old Indo-European Languages) that studies the Greek text of the New Testament as well as its translations into the old Indo-European languages Latin, Gothic, Armenian and Old Church Slavonic, having pragmatic objectives as: word order, discourse particles, pronominal reference and the use of null pronouns, expressions of definiteness and the use of participles to refer to background events, taking part at UD (Universal Dependencies)[10]. After its affiliation at UD, the project must continue by adding new texts. They accepted that we add a New Testament in old Romanian, printed in the seventeenth century, and then, probably, other old Romanian texts, printed in the sixteenth century.

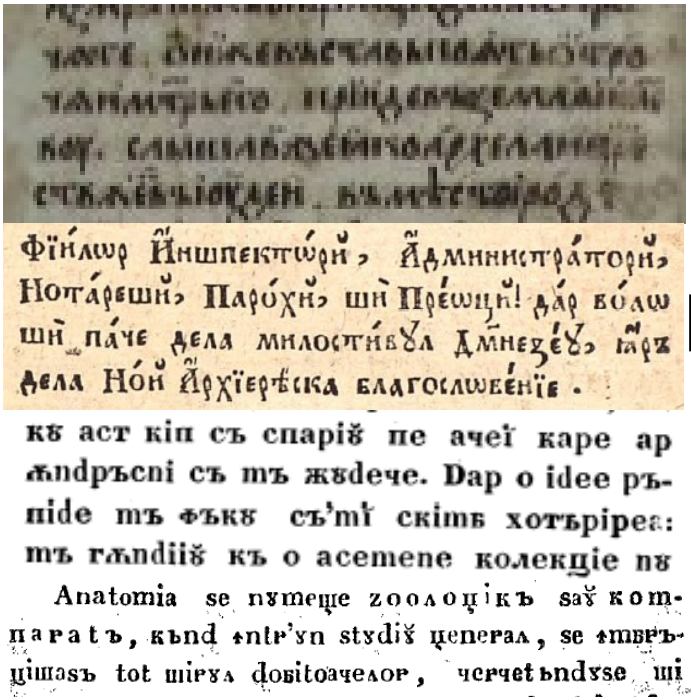
3 Processing the Old Romanian

3.1 Building an Optical Character Recognition Technology

Both in Romania and in Republic of Moldova, the old texts, the first printed books are written in an old Cyrillic alphabet that has 47 letters and is not recognized as the ASCII encoding. Some of the corresponding Unicode points were introduced only since 2009, and we found only three

[8] http://www.ninjal.ac.jp/english/research/project/a/corpus/
[9] http://www.hf.uio.no/ifikk/english/research/projects/proiel/
[10] https://github.com/UniversalDependencies

fonts covering them. The technology of their recognition is developing by a group of researchers of the Institute of Mathematics and Computer Science of the Academy of Sciences of Moldova. This technology uses FineReader OCR program and proposes the corresponding set of character templates, user languages and dictionaries, transliteration programs. The program has been trained on old Romanian Cyrillic texts, and also on texts written in the nineteenth century that replaces some Cyrillic letters by Latin ones (so called transition alphabets).



Coresi 1560

Bobb 1808

Alexandrescu 1838

Kretzulescu, 1843

Figure 1. Some Romanian Cyrillic characters in printed books.

In the third example, there is a text with transition characters, the letters: *a*, *d*, *n*, *i*, *e*, *m* are Latin, the rest of letters are Cyrillic. In the fourth example, the Latin letters *t*, *s*, *z* were added to the above ones.

In all the European countries there exist projects to scan the old books, threatened with destruction, such as Impact digitization.eu[11]. In Romanian, the old books of the big libraries were scanned by Dacoromanica[12], the Soros foundation, in the Republic of Moldova by Moldavica[13]. More than 100 old books were downloaded to train the OCR for old Romanian. The challenge for the OCR is the different form of characters (see Figure 1), the necessity to make them editable and to translate them into Latin characters.

To support OCR, a list of words and inflected forms that could be found in the processed texts should be introduced. Because there are old texts, written before the fixation of rules for the correctness of the language, this list must be very big, all phonetic or orthographic variations of words being permitted for the writers.

The word list was obtained indexing a big corpus of texts transcribed with Latin letters by experts. The modern editions of old books were processed with an OCR program for Latin letters and saved in TXT format. These texts must be not only checked to eliminate the OCR mistakes, but also "cleaned" of all items that do not belong to the old text: meta-text, i.e. information about the editor, publisher, print, critical tools as: content, indexes, list of abbreviations, preface, comments, notes, observations, bibliography. These critical comments form more than half of the modern edition, and their language is contemporary.

The "cleaned", (now, entirely old) texts were processed with the program Lucon 03.16[14]. The tool has created an index of 117,000 entries.

3.2 Building a lexicon for the UAIC-RoBinPOS-tagger

[11] http://www.digitisation.eu/tools-resources/language-resources/impact-es/

[12] https://www.google.ro/webhp?sourceid=chrome-instant&ion=1&espv=2&ie=UTF-8#q=dacoromanica.ro

[13] http://www.moldavica.bnrm.md/index.html

[14] https://sourceforge.net/projects/lucon/, by Cătălin Mititelu

The lexicon obtained is perhaps sufficient for the OCR program, but for the POS-tagger for old Romanian it is not yet sufficient.

The POS-tagger is a program that includes more functions in pipeline: a splitter, that separates sentences, a tokenizer that splits words and punctuation elements, a lemmatizer that finds lemma of each word, i.e. the generic form of the dictionary, and finally it adds the "postag", i.e. the part of speech and the morphological analysis of each word form (occurrence) in the text. The UAIC RoBinPOS-tagger [11] is hybrid, i.e. statistical, but also permitting introduction of rules for eliminating frequent mistakes.

In order to be able to make all these operations, the POS-tagger needs a big lexicon, containing not only the word form, but also the lemma and the "postag" of each word form, and finally it needs a big gold corpus for training. The lexicon must contain not only all the word forms found by indexing the texts, but all the inflected forms possible for these word forms.

The solutions found for carrying out this difficult task are described below. For this task, we cannot ignore the indexes at the end of each modern edition of old books. The indexes must be also "cleaned" eliminating the comments, and then the morphological categories added by the experts (having different logos) must be converted in the system of labels used by our POS-tagger, the system of the MULTEXT-East[15] project [2], [3] (that was a little simplified). For example, *subst. com. masc. sing. gen. articulat* (common noun masculine singular genitive case with definitness) becomes Ncmsoy in the conventions of our POS-tagger.

3.2.1 Extraction of the list of stable MWEs

After the addition of all indexes at the index obtained by the Lucon 03.16, another type of information is needed. We extracted the stable, unanalyzable MWEs from the DELS dictionary [6]. The fixed MWEs, called *Locuțiuni* (set phrases) have in the dictionary the labels (Loc. vb.),

[15] https://nl.iis.si/ME

(Loc. adv.) and so on, containing part of speech of synonym word. The lexicographic conventions (parentheses, tag usage labels, *and*, *or*, *someone*, *somebody*, punctuation, etc.) has been eliminated from the list of the stable MWEs, like in the Table 1. The table is continuing containing all the inflected forms of the verb *da* (give).

Table 1. Fragment of the lexicon for the POS tagger.

Lemma	Form	POS
a_o_da_prin_șperlă	a_o_da_prin_șperlă	Vmn
a_o_da_prin_șperlă	dat_prin_șperlă	Vmp
a_o_da_prin_șperlă	o_dădui_prin_șperlă	Vmis1s
a_o_da_prin_șperlă	o_dădușI_prin _șperlă	Vmis2s
a_o_da_prin_șperlă	o_dădu_prin_șperlă	Vmis3s
…	…	…

3.2.2 Example for the extraction of variants

Another type of information that must be added to our lexicon is the list of variants with the word entry from eDTLR, the electronic form of DTLR [9], [10]. The variants look like in the Example 1, where we extracted the first (the word entry) and the penultimate paragraph of the dictionary entry (containing the variants). If the lexical variant is found in the quotations, no citation of sources is provided, if the variant is found in other bibliographical source, this source is cited in this paragraph.

Example 1

"**POJĂRI** vb.

– Şi: (învechit) **pojerí, pojorí** vb. IV.

POJGHÍŢĂ s. f.

– Şi: **poşghíţă** (TEODORESCU, P. P. 365), (învechit) **puşghíţă** (LM), (regional) **pojíţă, pojníţă** (H X IV 437), **pojvíţă, pujíţă** (com. din STRAJA - RĂDĂUŢI) s. f."

[En: **SCORCH** verb

– And (old) **pojerí**, **pojorí** verb. IVth conjugation.

CRUST feminine noun.
– And: **poşghíţă** (TEODORESCU, P. P. 365) (obsolete) **puşghíţă** (LM), (regional) **pojíţă, pojníţă** (H 437 X IV) **pojvíţă, pujíţă** (comunicated from STRAJA - RADAUTI) feminine noun.]

In this fragment, *pojări* (*scorch*) is lemma for the words-form *pojeri*, *pojorî*, and *pojghiță* (*crust*) is lemma for the words-form *poșghiță*, *puşghiță*, *pojiță*, *pojniță*, *pojviță*, *pujiță*. These variants must be introduced in our lexicon together with the word entry, that is their lemma. To obtain the postag, we must replace *vb.* with Vmn and *s. f.* with Ncfsrn. In the dictionary there exist only generic forms, infinitive for verbs, nominative without article for nouns, etc. The parentheses containing the attestation, the style or other information must be eliminated.

These word forms will be also added to our index. In the needed format, each word form must be a separate line followed by the lemma and the postag, like in the Example 2. Introducing these variants found in DTLR (16 volumes), our lexicon for POS-tagger will have also regional word forms to be able to process the regional texts.

Example 2

pojeri pojări Vmn
pojeri pojori Vmn
pojiță pojghiță Ncmsrn
pojniță pojghiță Ncmsrn
pojviță pojghiță Ncmsrn
pujiță pojghiță Ncmsrn … and so on.

The following step is the alphabetical ordering and the elimination of repetitions of the same information, after introducing all indexes, MWEs and variants found in our big amount of sources.

The last step is the processing of the resulted index with a program that generated paradigms of old inflexions. This program was developed by Radu Simionescu and Daniela Gîfu [4]. We must verify and complete these paradigms using the forms found in the index.

The variety of forms is very big. For example, the list of demonstrative pronouns and demonstrative pronominal adjectives looks like this:

acea, aceaia, aceaste, ace, acee, aceae, aceaea, aceaeaș, aceaeași, aceaești, aceaiașii, aceai, aceaiă, aceaiașe, aceaiaști, aceaialaltă, aceaiașu, aceaiia, (*this, these, those, that*)… and so on, over 250 forms.

The paradigms must be: with or without the diphthong of accented a=ae, with or without definite article i=ii, with or without composition of more demonstratives, with various kinds of composition: *aceaiaști, aceallalt, aceaialaltă, cealalaltu, cestulaltu, acestălant* (*the same, the other*), and so on.

This lexicon will be introduced in the RoBinPOS-tagger and then, the first old book that will be processed is the New Testament of Bălgrad (Alba Iulia) printed in 1648, the variant with Latin letters. This book will be introduced in the PROEL project. There exists another Romanian New Testament, 20 years older, but it is a manuscript, it is not printed and will have a modern edition in 2017, carried out by Eugen Munteanu. We will compare them in 2017 and show the differences between the first New Testament manuscript and the first printed New Testament from 1648.

We expect that the output of the first old book automatically processed by the UAIC-RoBinPOS-tagger will have numerous mistakes and will need a carefully supervision before processing by the UAIC syntactic parser.

4 The Training Corpus for the UAIC-RoBinPOS-tagger

4.1 Deficiencies in the Current Training Corpus

In the lexicon of the POS-tagger there are not all the word-forms that the tool can find in the text that it must process. In other cases, the same word form can have more morphological analyses and sometimes different lemmas. The word form *sare* can have the lemma *sări* and the postag=Vmip3s or the lemma *sare* and the postag=Ncfsrn.

These situations, called homonymies or ambiguities, explain the necessity of statistical functioning of the POS-taggers, in all the languages. To train the statistical machine learning, the training corpus must be as bigger as the number of labels it needs to learn to apply tags is bigger. As a variant, the number of labels in the MULTEXT east morphological conventions of annotation is more than 600 (reduced for our POS-tagger as 430). It needs a training corpus formed on millions of words correctly morphological annotated.

Unlike the syntactic parsers, that can be language independents, the POS-taggers are language dependents, each language having its peculiar morphology. But the languages with an unsatisfactory level of computerization have not big corpora correctly annotated and cannot find big corpora for the training of their POS-taggers.

This situation is solved by the computer scientists in different modalities. The Acquis Communitaire, a big corpus in legal style, containing rules, laws and procedures, is aligned in all the languages of the European Community. If it has been morphological annotated in one of these languages, the others can import the annotations, but the result can be not perfectly adequate with the specific morphology of the other language.

Another question to be asked is if the morphological annotation in the trained corpus obtained by this way has the same annotation conventions like the POS-tagger to be trained; probably, it does not. Then, the computer scientists add the condition that the machine replaces the annotations inexistent in its system by the nearest label in its own set of labels… and the result is approximately, or random correct!

The MULTEXT East corpora were also used for the training of POS-taggers. The aligned versions of the Orwell's *1984* novel translated in more languages were manually annotated with morphologic information by native linguists in each participant language. However, the conventions of annotation are sometimes debatable, sometimes outdated in the current state of research. We give here only some examples:

1. Part of computer scientists prefer to annotate all the verbal participles as adjectives; therefore, after the syntactic annotation, the

passive constructions are formed by an adjective head for an auxiliary verb and the root of the passive sentence is the adjective.

2. The words are split in letters, but actually the NLP is separated in two specializations: the speech processing and the written language processing. The first group needs to split the words in sons and not in letters (it isn't the same thing) and the second group takes the word as the smallest unit annotated and continues with the syntactic annotation. The segmentation in letters is obsolete.

3. Because the POS-taggers failed in 2001 to correctly annotate hyphen or apostrophe, marking the union between two words as part of one of them losing sound by stacking reported, MULTEXT East has added more than 170 logos marking the words with and without hyphen. That is called "clitic" and, moreover, it has an homonym annotation as the definite article: "y". If there are 2 y, it is "+definitiveness +clitic"; if there is one, it indicates either one, either the other! In the example 3, we transcribe a fragment of the list of labels of the MULTEXT East used by the POS-tagger of RACAI (Research Academic Institute for the Artificial Intelligence in Bucharest).

Example 3:

Ncfp-ny	Noun common feminine plural -definiteness +clitic
Ncfpoy	Noun common feminine plural oblique +definiteness
Ncfpoyy	Noun common feminine plural oblique +definiteness +clitic
Ncfpry	Noun common feminine plural direct +definiteness
Ncfpryy	Noun common feminine plural direct +definiteness +clitic…

In Romanian, the hyphen can appear between any words in poetry to decrease the number of syllables: *fecioară-ntre femei* (*virgin between women*) and it is not acceptable that it was always called "clitic". We removed all labels from the set of labels "with clitic" and, however, the UAIC-POS-tagger annotates the hyphen almost perfectly (as separating two words and appertaining of word belonging incomplete).

4. The big number of labels is difficult to be managed. CONLL-U, a modern universal set of conventions of annotation, is adopted by more and

more corpora, because it has a reduced number of labels, structured in more levels: UPOSTAG (Universal part-of-speech tag[16]), XPOSTAG (Language-specific part-of-speech tag) and FEATS List of morphological features from the universal feature inventory[17]. The permanent change of the format, in a compatible way with the new universal adopted, is required in order to maintain a resource in the attention of the researchers, to increase and to reuse it.

4.2 Building a New Balanced Training Corpus

Another problem is the style of the training corpus. As yet, the training corpus was formed by legal style, fictional and journal style, all contemporary standardized texts. But a POS-tagger trained in this way cannot annotate the old and regional unstandardized texts.

The conclusion is that we must renounce at the opportunistic solutions, no time consuming, without charges, and become to construct a new corpus for the training of our tools. We will use in this way a consistent set of conventions. If one of these conventions is disputable, we can automatically change it in the entire corpus, only if it is consistently annotated everywhere.

The UAIC-RoDepTb will be entirely manually checked at the level of the morphological annotation. In present, it is entirely checked at the level of the syntactic annotation, and approximately 110,000 tokens were checked at the morphological level, the rest of 95,000 tokens must be checked after the end of this year. It will become a new gold corpus for the training, but 205,000 items are still little.

The introduction of new texts will be progressive and balanced, in all the styles, standardized and unstandardized; the regional Romanian texts will be processed and checked by Augusto Perez simultaneously with the New Testament checked by Cătălina Mărănduc and with texts from Republic of Moldova, checked by other linguists.

[16] http://universaldependencies.org/u/pos/index.html
[17] http://universaldependencies.org/u/feat/index.html

We intend to apply for a financing project, because we need a big number of researchers (linguists and computer scientists) for a big balanced corpus. By applying the bootstrapping method (increasing the training corpus by the addition of carefully checked outputs of the same tools), the corpus for training and the accuracy of the POS-tagger in the processing of old and regional texts will increase.

5 The Introduction of the New Testament of Bălgrad (1648) in PROIEL

At this moment, the "Romanian treebank" introduced in UD (the Universal Dependencies project) is formed only by Contemporary standardized Romanian, being not representative for our treebank. We obtained the accept to introduce our oldest printed New Testament in the PROIEL project, affiliated at UD, so we will make known of the 30 countries participating in UD, our work in building a diachronic corpus illustrating geographical and historical variation of Romanian. The Bulgarian Treebank works also at the annotation of the old and dialectal Bulgarian.

We have now the printed version of the New Testament of Bălgrad (NT), downloaded from the site of the Library of the University of Cluj-Napoca. The print was processed by the free Scan Taylor program. We scanned also the modern edition of the New Testament of Bălgrad, with Latin letters, and the scanned text was processed by an OCR.

The editable text has been checked for eliminating the OCR mistakes and then "cleaned" eliminating all the editor contributions and preserving only the old text. In Figure 2, fragment of the first printed Romanian NT (1648) can be seen. The text is as follows (Latin letters):

"DE LA MATTEIU SFÎNTA EVANGHELIE. CAP 1. Neamul şi naşterea lui Iisus Hristos carele iaste Mesia făgăduit izbăvitor părinţilor. Cartea de neamul lui Iisus Hristos, fiiul lui David, fiiul lui Avraam. 2. Avraam născu pre Isaac, iară Isaac născu pre Iacov, iară Iacov născu pre Iuda şi pre fraţii lui. 3. Iuda născu pre Fares şi pre Zara din Tamar, iară Fares născu pre Esrom şi Esrom născu pre Aram. 4. Aram născu pre Aminadav, iară Aminadav născu pre Nasson, Nasson născu pre Salmon."

[BY MATTHEW SAINT GOSPEL. Chapter 1.1 The book of the generation of Jesus Christ, the son of David, the son of Abraham. 2 Abraham begat Isaac; and Isaac begat Jacob; and Jacob begat Judas and his brethren; 3 And Judas begat Phares and Zara of Thamar; and Phares begat Esrom; and Esrom begat Aram; 4 And Aram begat Aminadab; and Aminadab begat Naasson; and Naasson begat Salmon;]

Figure 2. Printed and processed with the Scan Taylor versions of the first page of the NT Bălgrad (1648).

The cleaned text has been introduced in the set of texts indexed by the LUCON 03.16 program, so any word in the NT can be found in the lexicon for the OCR and for the POS-tagger.

Because the PROIEL is a project of old languages, having the texts written of Greek, old Armenian and old Slavonic letters, we intend to introduce in the project the NT with editable old Romanian Cyrillic letters. It was a challenge for Ludmila Malahov, Alexandru Colesnicov and their colleagues to process this printed book from the XVII century.

The NT will be processed by the UAIC-RoBinPOS-tagger and by the syntactic parser, and then it will be checked, both at the morphological and at the syntactic level.

The following step will be the transformation of the NT from the UAIC convention of annotation into the UD ones. Although we have made the transposition table for transposing our conventions into UD

ones, the author of the automatic program for the transposition is Radu Ion, from the RACAI group.

Finally, the researchers of the Institute of Mathematics and Computer Science of the Academy of Sciences of Moldova participate in the alignment of two versions and export of the annotations to the versions with Cyrillic editable Letters, that will be introduced in the PROIEL.

6 Conclusion

After the training of the OCR and POS-tagger on old texts, we will be able to increase the RoDia corpus.

This corpus has the beginning in the balanced UAIC-RoDepTreebank that contains 230 sentences in popular and regional style of the language and 650 sentences in the old Romanian. The annotation of these first texts in the new styles was manually carried out.

For the increase of the tools accuracy, we must construct a big lexicon of the old Romanian and also lexicons for the three South Danube dialects of Romanian. We must also construct a big gold corpus for the training of the tools, especially of the POS-tagger.

Until they manage to obtain financing, the continuation of this project is based on friendship and mutual respect between the linguists and computer scientists from Iaşi and Chişinău.

References

[1] S. Dipper, L. Faulstich, U. Leser, A. Ludeling, *Challenges in Modelling a Richly Annotated Diachronic Corpus of German*. Proceedings of LREC (2010).

[2] T. Erjavec, *Harmonized Morphosyntactic Tagging for Seven Languages and Orwell's 1984*. Proceedings of the 6th Natural Language Processing Pacific Rim Symposium, Tokyo, (2001), pp. 487-492.

[3] T. Erjavec, *MULTEXT-East Version 3: Multilingual Morphosyntactic Specifications, Lexicons and Corpora*. Proceedings of the Fourth Intl. Conf. on Language Resources and Evaluation, LREC'(2004).

[4] D. Gîfu, R. Simionescu, *Tracing Language Variation for Romanian*. Proceedings of the 17th International Conference on Intelligent Text

Processing and Computational Linguistics, CICLing 2016, 3-9 Apr. 2016, Konya, Turkey.

[5] E. Hinrichs, T. Zastrow, *Linguistic Annotations for a Diachronic Corpus of German.* In Linguistic Issues in Language Technology, No. VII, (2012).

[6] C. Mărănduc *The Dictionary of Romanian Expressions, Syntagms and Set Phrases* (DELS). Corint Publishing, Bucharest (2010), 558 pp.

[7] T. McEnery, R. Xiao, Y. Tono, *Corpus-based Language Studies: An Advanced Resource Book.* Routledge publisher, London, New York (2006), 389 pp.

[8] C. Onelli, D. Proietti, C. Seidenari, *The DiaCORIS project: a diachronic corpus of written Italian.* Proceedings LREC (2006), pp. 1212-1215.

[9] Romanian Academy, editor, *Dictionary of Romanian Language.* I. Socec Universul Publishing, Bucharest, (1913-1949).

[10] Romanian Academy, editor, *Dictionary of Romanian Language*. II. Romanian Academy Publishing, Bucharest, (1965-2010).

[11] R. Simionescu, *Hybrid POS Tagger.* Language Resources and Tools in Industrial Applications, Eurolan 2011 summer school.

[12] Taavitsainen, A. H. Jucker, J. Tuominen (Eds.), *Diachronic Corpus Pragmatics,* John Benjamins publisher, Amsterdam, (2014) 335 pp.

Cătălina Mărănduc[1,3], Cenel-Augusto Perez[1], Ludmila Malahov[2], Alexandru Colesnicov[2]
[1]Faculty of Computer Science, Al. I. Cuza University, Iași, Romania
E-mail:{catalina.maranduc, augusto.perez}@info.uaic.ro
[2]Institute of Mathematics and Computer Science of the Academy of Sciences of Moldova
E-mail: lmalahov@gmail.com, acolesnicov@gmx.com
[3]Academic Institute of Linguistics "Iorgu Iordan – Al. Rosetti" Bucharest, Romania.

Proceedings of the Conference on Mathematical Foundations of Informatics
MFOI2016, July 25-29, 2016, Chisinau, Republic of Moldova

Deniable-encryption protocol using commutative transformation

Nicolai A. Moldovyan, Alexandr A. Moldovyan,
Alexei V. Shcherbacov

Abstract

It is proposed a new design of the deniable-encryption protocols, which is based on using commutative transformation of messages sent between two parties of communication session. For performing the data encryption there is used a shared key having small size (16 to 56 bits), high security and bi-deniability of the protocol being provided though. The protocol has sufficiently high performance and resists active coercive attacks due to using the shared key. It is designed so that data send via insecure channel during the protocol cannot be used to compute any information about the shared key and the deniable encryption is computationally indistinguishable from probabilistic encryption.

Keywords: cryptography, encryption, commutative encryption, deniable encryption, shared key, probabilistic encryption.

MSC 2000: 94A60, 11S05.

1 Introduction

The notion of deniable encryption (DE) relates to the cryptographic schemes that resist the attacks performed by the so-called coercive adversary that intercepts the ciphertext and has power to force sender or/and receiver to open both the sent message and the encryption key [1]. The DE schemes are potentially applicable for preventing vote

buying in the internet-voting systems [3] and to provide secure multiparty computations [6]. One can distinguish sender-deniable [1], [9], receiver-deniable [15], [4], and bi-deniable [13], [10] schemes in which coercer attacks the sender of secret message, the receiver, and the both parties of the communication session, respectively. The deniability is provided, if the sender or/and the receiver has possibility to open a fake message instead of the secret one and the coercer is not able to disclose their lie. One of the important problems relating to the deniable encryption schemes is justifying their deniability [2], i.e. computational infeasibility for the coercive adversary to prove that the ciphertext can be decrypted into a message different from the fake message. There are known the public-key DE schemes [10], [11] and the shared-key DE ones [12] which are well suitable for practical application. In the known shared-key DE schemes the deniability is provided with using the shared keys having sufficiently large size.

The present paper introduces an approach to the design of the bi-deniable shared-key schemes using short keys (16 to 56 bits), the schemes resisting passive and active coercive attacks. The main feature of the proposed design consists in using the three-path protocol for sending a secret message without exchanging keys [7] as internal part of the DE scheme (about Shamir's no key protocol see p. 500 in[7]). The shared key is used only for authenticating the ciphertexts sent between the parties of the constructed three-pass protocol that is secure against both the passive and active coercive attacks. To justify bi-deniability of the proposed protocol it is provided its computational indistinguishability from the probabilistic three-pass protocol with which a fake message is sent to the message receiver.

The paper is organized as follows. Section 2 describes the model of the coercive adversary and the design criteria. Section 3 describes the proposed method based on the computational indistinguishability between the deniable encryption and the probabilistic one. In the proposed protocol there is used a shared key having sufficiently small size, security of the data encryption being provided with the commutative-encryption algorithm proposed in [5]. The shared key is used for au-

thentication of the values sent between parties of the communication protocol. Section 4 discusses security, bi-deniability, and the practical applicability of the proposed protocol. Section 5 concludes the paper.

2 Design criteria and model of the coercive attack

For constructing a practical bi-deniable encryption protocol the following design criteria have been proposed:

1) the protocol should use a key shared by sender and receiver of secrete message, the size of the key being sufficiently small (16 to 56 bits);

2) data sent via insecure channel should provide no possibility to find the shared key with the help of the exhaustive-search attack;

3) security of the data encryption should be sufficiently high; at least it should be provided the level of 80-bit security;

4) the protocol should provide bi-deniability, i.e. it should resist simultaneous coercive attacks on the sender and the receiver;

5) the base encryption procedure should be implemented using the commutative transformation that is free from using any shared key; the commutative transformation is to be dependent on some local parameter called local key;

6) under coercive attack the parties of the protocol disclose some fake shared key and their local fake keys as secret values; while using the disclosed keys, a fake message is produced from the ciphertexts sent via insecure channel during the deniable-encryption protocol;

7) ciphertexts produced at all steps of the protocol should be computationally indistinguishable from the ciphertexts produced by some probabilistic-encryption protocol (that is called associated probabilistic three-pass protocol) in the case when the last protocol is used for enciphering the fake message with using the disclosed keys;

8) while using the secret shared key, the receiver is able to disclose secret message;

9) performance of the protocol should be sufficiently high, for example, less than ten modulo exponentiation operation are to be performed during the protocol.

The used design criteria are aimed to providing resistance to potential coercive attacks implemented by passive and active adversaries. In present paper it is assumed the model of the coercive attack that is characterized by the following items:

- the adversary intercepts all ciphertexts sent via communication channel;

- coercive adversary attacks the parties of the protocol after the ciphertexts have been sent via public channel;

- both the sender and the receiver are forced to disclose the plaintext corresponding to the ciphertexts sent via communication channel; each of them is also forced to disclose the keys that have been used during the encryption process;

- the parties of the protocol should disclose decryption algorithm, the output of which depends on each bit of the ciphertexts; if the decryption is performed using the disclosed keys, then the output message should be equal to the disclosed plaintext;

- the adversary can try to impose a false communication session and to send two different messages to the receiver; if the receiver (under coercive attack) will not disclose one of the messages, then such attack is considered as the successful one, since the adversary is able to prove that the receiver is lying;

- in false communication session the adversary can try to play the role of the receiver and to obtain the secret message directly during the execution of the protocol.

3 Bi-deniable encryption protocol using a short shared key

3.1 Associated probabilistic three-pass protocol

Suppose the sender (Alice) and receiver (Bob) of secret message $M < p$, where p is a large prime having size $|p| \geq 1024$ bits, share a secret key $K = (k_1, k_2)$, where $8 \leq |k_1| = |k_2| \leq 28$ bits. The following probabilistic protocol provides a secure method (if the number $p-1$ contains a large prime divisor having size $|q| \geq 160$ bits) for sending the message M via an insecure channel:

1. Alice generates a random number e_A such that $|e_A| \geq 160$ bits and the greatest common divisor $gcd(e_A, p-1) = 1$ and computes the value $d_A = e_A^{-1} \bmod (p-1)$. Then she encrypts the message M as follows:

1.1. Compute the intermediate ciphertext $C_A = M^{e_A} \bmod p$.

1.2. Generate a random number $R_A < p$ and compute the value

$$S_A = (C_A - k_1 R_A) k_2^{-1} \bmod p.$$

1.3. Form the ciphertext $C_1 = (R_A, S_A)$.

Then Alice sends the ciphertext C_1 to Bob.

2. Bob generates a random number e_B such that $|e_B| \geq 160$ bits and $gcd(e_B, p-1) = 1$ and computes the value $d_B = e_B^{-1} \bmod (p-1)$. Then he encrypts the intermediate ciphertext C_A as follows:

2.1. Compute the value

$$C_A = (k_2 S_A + k_1 R_A) \bmod p$$

and the intermediate ciphertext $C_{AB} = C_A^{e_B} \bmod p$.

2.2. Generate a random number $R_B < p$ and compute the value

$$S_{AB} = (C_{AB} - k_1 R_B) k_2^{-1} \bmod p.$$

2.3. Form the ciphertext $C_2 = (R_B, S_{AB})$.

Then Bob sends the ciphertext C_2 to Alice.

3. Alice decrypts the intermediate ciphertext C_{AB} and creates the ciphertext C_3 as follows:

3.1. Compute the value $C_{AB} = (k_2 S_{AB} + k_1 R_B) \bmod p$ and the intermediate ciphertext $C_B = C_{AB}^{d_A} \bmod p$.

3.2. Generate a random number $R'_A < p$ and compute the value

$$S_B = (C_B - k_1 R'_A) k_2^{-1} \bmod p.$$

3.3. Form the ciphertext $C_3 = (R'_A, S_B)$.

Then Alice sends the ciphertext C_3 to Bob.

4. Bob computes the intermediate ciphertext $C_B = (k_1 R'_A + k_2 S_B) \bmod p$ and the message $M' = C_B^{d_B} \bmod p$.

Proof of the protocol correctness:

$$M' \equiv C_B^{d_B} \equiv (C_{AB}^{d_A})^{d_B} \equiv (C_A^{e_B})^{d_A d_B} \equiv (M^{e_A})^{e_B d_A d_B} \equiv$$
$$\equiv M^{e_A e_B d_A d_B} \equiv M \bmod p \Rightarrow M' = M.$$

During the protocol the pair of numbers (e_A, d_A) is used only by Alice, therefore the pair (e_A, d_A) can be called Alices local key. The pair of numbers (e_B, d_B) serves as Bob's local key.

3.2 Deniable-encryption three-pass protocol

To provide encryption deniability in the protocol described in Section 3.1 one can use the ciphertexts produced by means of encryption of secret message $T < p$ (like encryption of the message M) as random values R_A, R_B, and R'_A. The message M will serve as a fake message. Suppose also the parties of the deniable encryption protocol share the fake key $K = (k_1, k_2)$ and the secret key $Q = (q_1, q_2)$, where $8 \le |q_1| = |q_2| \le 28$ bits, such that $q_1 k_2 \ne q_2 k_1$. Indicated modification leads to the following bi-deniable encryption protocol in which Alice sends the secret message T to Bob:

1. Alice generates a fake message $M < p$, two random numbers e_A and ε_A such that $|e_A| = |\varepsilon_A| \ge 160$ bits, $gcd(e_A, p-1) = 1$, and

$gcd(\varepsilon_A, p-1) = 1$ and computes the values $d_A = e_A^{-1} \bmod (p-1)$ and $\delta_A = \varepsilon_A^{-1} \bmod (p-1)$. Then she encrypts the messages M and T as follows:

1.1. Compute the intermediate ciphertexts $U_A = T_A^{\varepsilon} \bmod p$ and $C_A = M^{e_A} \bmod p$.

1.2. Solve the following system of two linear equations relative to unknowns R_A and S_A:

$$\begin{cases} k_1 R_A + k_2 S_A = C_A \bmod p \\ q_1 R_A + q_2 S_A = U_A \bmod p. \end{cases}$$

1.3. Form the ciphertext $C_1 = (R_A, S_A)$. Alice sends the ciphertext C_1 to Bob.

2. Bob generates two random numbers e_B and ε_B such that $|e_B| = |\varepsilon_B| \geq 160$ bits, $gcd(e_B, p-1) = 1$, and $gcd(\varepsilon_B, p-1) = 1$ and computes the values $d_B = e_B^{-1} \bmod (p-1)$ and $\delta_B = \varepsilon_B^{-1} \bmod (p-1)$.

Then he encrypts the intermediate ciphertexts U_A and C_A as follows:

2.1. Compute the values $U_A = (q_1 R_A + q_2 S_A) \bmod p$ and $C_A = (k_1 R_A + k_2 S_A) \bmod p$.

2.2. Compute the intermediate ciphertexts $U_{AB} = U_A^{\varepsilon_B} \bmod p$ and $C_{AB} = C_A^{e_B} \bmod p$.

2.3. Solve the following system of two linear equations relative to unknowns R_{AB} and S_{AB}:

$$\begin{cases} k_1 R_{AB} + k_2 S_{AB} = C_{AB} \bmod p \\ q_1 R_{AB} + q_2 S_{AB} = U_{AB} \bmod p. \end{cases}$$

2.4. Form the ciphertext $C_2 = (R_{AB}, S_{AB})$.

Then Bob sends the ciphertext C_2 to Alice.

3. Alice decrypts the intermediate ciphertexts U_{AB} and C_{AB} and creates the ciphertext C_3 as follows:

3.1. Compute the values $U_{AB} = (q_1 R_{AB} + q_2 S_{AB}) \bmod p$ and $C_{AB} = (k_1 R_{AB} + k_2 S_{AB}) \bmod p$ and the intermediate ciphertexts $U_B = U_{AB}^{\delta_A} \bmod p$ and $C_B = C_{AB}^{d_A} \bmod p$.

3.2. Solve the following system of two linear equations relative to unknowns R_B and S_B:

$$\begin{cases} k_1 R_B + k_2 S_B = C_B \bmod p \\ q_1 R_B + q_2 S_B = U_B \bmod p. \end{cases}$$

3.3. Form the ciphertext $C_3 = (R_B, S_B)$.

Then Alice sends the ciphertext C_3 to Bob.

4. Bob computes the intermediate ciphertext $U_B = (q_1 R_B + q_2 S_B) \bmod p$ and the message $T' = U_B^{\delta_B} \bmod p$.

Proof of the protocol correctness:

$$T' \equiv (U_{AB}^{\delta_A})^{\delta_B} \equiv (U_A^{\varepsilon_B})^{\delta_A \delta_B} \equiv (T^{\varepsilon_A})^{e_B \delta_A \delta_B} \equiv$$
$$\equiv T^{\varepsilon_A \varepsilon_B \delta_A \delta_B} \equiv T \bmod p \Rightarrow T' = T.$$

4 Discussion

4.1 Security against passive and active attacks

The protocol described in Section 3.2 resists attacks of passive adversary due to using commutative encryption (performed as modulo exponentiation operation) which produces intermediate ciphertexts C_A, C_{AB}, C_B, U_A, U_{AB}, and U_B. Suppose the adversary knows the keys $K = (k_1, k_2)$ and $Q = (q_1, q_2)$. Then, after intercepting the ciphertexts C_1, C_2, and C_3 he can compute the indicated intermediate ciphertexts, including the values U_{AB}, C_{AB}, U_B, and C_B.

Solving exponential equations

$$U_{AB} = U_A^{\varepsilon_B} \bmod p \text{ and } C_{AB} = C_A^{e_B} \bmod p$$

relative to unknowns e_B and ε_B gives theoretically the values of two Bob's local keys used in the protocol. Correspondingly, solving exponential equations

$$U_B = U_{AB}^{\delta_A} \bmod p \text{ and } C_B = C_{AB}^{d_A} \bmod p$$

relative to unknowns d_A and δ_A gives theoretically the values of two Alice's local keys. Any of local keys (e_A, d_A) and (e_B, d_B) gives possibility to compute easily the message M. Any of local keys $(\varepsilon_A, \delta_A)$ and $(\varepsilon_B, \delta_B)$ gives possibility to compute easily the message T.

However, solving each of the last four equations means solving the computationally difficult problem of finding discrete logarithm modulo p.

Thus, one can define the required level of the protocol security against passive attacks with selecting sufficiently large size of the prime p used in the protocol. In the case $|p| \geq 1024$ bits the difficulty of the last problem is estimated as $\geq 2^{80}$ multiplications mod p [7].

The principal disadvantage of the three-pass protocol based on the exponentiation procedure used as encryption operation is weakness to active attacks in which the adversary plays role of the sender or receiver of secret message. To provide security to such attacks in the protocols presented in Sections 3.1 and 3.2 there are used shared keys.

Using the shared keys K and Q provides resistance against attacks of active adversary that can potentially impose a false communication session, the size of these keys is very small though. Indeed, values sent via communication channel during the protocol performed by Alice and Bob (legal users) cannot be used for finding the shared secret key with help of the exhaustive-search attack, even in the case of the known messages M and T, since the last values and the intermediate ciphertexts are transformed depending on the local keys having large size.

If the shared keys are not known to the adversary, the false communication session gives different values of the sent and received messages, i.e. the inequalities $M' \neq M$ and $T' \neq T$ will take place with probability $1 - 2^{-|K|}$ and $1 - 2^{-|Q|}$, correspondingly. To implement exhaustive-search procedure for finding the values K and Q the active adversary needs to impose very large number of false communication sessions (on the average $2^{|K|-1}$ and $2^{|Q|-1}$ sessions, correspondingly). Therefore the values $|K| \geq 16$ bits and $|Q| \geq 16$ bits seem to be sufficient to counteract active attacks.

4.2 Bi-deniability

Suppose coercive adversary intercepts the ciphertexts C_1, C_2, and C_3 and attacks simultaneously Alice and Bob after they have finished the communication session that has been performed in accordance with the proposed deniable-encryption protocol. Then they open their local keys (e_A, d_A) and (e_B, d_B) and message M. They also claim that they used the probabilistic-encryption protocol described in Section 3.1. (One can call this protocol associated probabilistic-encryption protocol.)

Using the values

$$C_1 = (R_A, S_A),\ C_2 = (R_{AB}, S_{AB}),\ \text{and}\ C_3 = (R_B, S_B)$$

the coercer can compute the intermediate ciphertexts C_A, C_{AB}, and C_B.

However to distinguish the pseudorandom values R_A, R_{AB}, and R_B, respectively, from random values R_A, R_B, and R'_A relating to the associated probabilistic three-pass protocol the attacker is to solve the discrete logarithm problem modulo p many times, i.e. for different values of the key $Q = (q_1, q_2)$, until he gets one of local keys $(\varepsilon_A, \delta_A)$ and $(\varepsilon_B, \delta_B)$ and the message T. For this purpose he can use, for example, the relation $R_A = (T^{\varepsilon_A} - q_2 S_A) q_1^{-1} \bmod p$, where ε_A is unknown.

Finding discrete logarithm modulo a large prime p, such that the number $p-1$ contains a large prime divisor, is a computationally difficult problem. Therefore the bi-deniability of the protocol is provided in the case of passive coercer. Suppose the coercer performs active attack of the first type, in which he plays role of the sender when performing the three-pass DE protocol. In this case the coercer sends the ciphertexts C_1 and C_3 to Bob and receives the ciphertext $C_2 = (R_{AB}, S_{AB})$. During performing the three-pass DE protocol the coercer knows no key shared by Bob and Alice, therefore he is unable to compute properly the values C_1 and C_3 and at final step Bob computes some random messages M' and T'. When being coerced, Bob opens the values M', $K = (k_1, k_2)$, and (e_B, d_B) and declares he had used the probabilistic three-pass protocol (from subsection 3.1) during the performed communication session. Without knowing the key $Q = (q_1, q_2)$ the coercer is not able to show that actually Bob acted in accordance with the

three-pass DE protocol, i.e. that Bob have computed the value R_{AB} as solution of the system of equations indicated in item 2.3 of the protocol from subsection 3.2. In active attack of the second type the coercer plays role of the receiver and sends (to Alice) the ciphertext C_2 and receives the ciphertexts C_1 and C_3, where the value C_1 contains both the fake M and secret T messages. However, during performing the communication session the coercer knows no shared key and he is unable to compute properly the value C_2. Therefore at final step of the protocol the coercer computes random messages M' and T'. When being coerced, Alice opens the values M', $K = (k_1, k_2)$, and (e_A, d_A) and declares she had performed the probabilistic three-pass protocol.

To disclose her lie the coercer has to distinguish the pseudorandom value R_B from the random value R'_A. Without knowing Alice's local key $(\varepsilon_A, \delta_A)$ the last problem is computationally infeasible.

Thus, in the both variants of the active coercive attack the bideniability of the proposed protocol is provided. One should mention that in the first-type attack Bob concludes that the received message T' was sent by an adversary since T' is not a sensible message, whereas in the case of the second-type attack Alice does not know whether Bob received her secret message. If it is needed to notify that Bob received the message T, then one can include into the protocol the additional step at which Bob sends the hash value computed from M to Alice. If Bob is not able to present correct hash value, then Alice concludes the performed communication session is false.

4.3 Practical applicability

The computational complexity of the proposed protocol is defined mainly by eight modulo exponentiation operations. As compared with the well known ElGamal public encryption protocol [14] and with Shamir's three-pass protocol [7] that uses the exponentiation operation as commutative encryption [5] the proposed protocol has performance about 4 and 2 times lower, correspondingly. Comparing with the earlier published deniable encryption protocols the proposed one is significantly faster. Besides, the last provides subexponential bi-

deniability whereas many of the known deniable-encryption schemes are not bi-deniable or have polynomial deniability [2].

Therefore, for providing resistance to coercive attacks the proposed protocol is very attractive, it does not suite well for application in electronic election systems to prevent vote buying though. For the mentioned application some modification of the proposed protocol is required, the basic design idea can be used to create protocols suitable for preventing vote buying though. However details of the design of such protocols are out of the topic of present paper. The associated probabilistic-encryption protocol from subsection 3.1 has individual and independent practical significance due to its providing high security level of the communication via insecure channels in the case of availability of shared keys having small size.

5 Conclusion

It has been presented a protocol for bi-deniable encryption suitable for application in communication systems that are resistant to coercive attacks as well as in secure communications in the case of availability of shared keys having restricted size.

The protocol uses very short shared keys with which it is provided resistance to potential active attacks. High security of the data encryption is based on difficulty of discrete logarithm problem due to using the encryption procedure that includes performing the exponentiation operations modulo a large prime which depend on local keys, like in the well known method for commutative encryption [5].

The present paper proposes implementation of the method for deniable encryption based on commutative encryption in the form of the exponentiation operation in the finite field $GF(p)$, therefore the described has subexponential bi-deniability. To obtain exponential bi-deniability one can implement the method using the commutative encryption in the form of the operation of multiplying points of an elliptic curve [8], however consideration of the details of such design variant represents interest for independent research.

References

[1] Canetti R., Dwork C., Naor M., Ostrovsky R. Deniable Encryption. Proceedings Advances in Cryptology - CRYPTO 1997. Lectute Notes in Computer Science. Springer-Verlag. Berlin, Heidelberg, New York, 1997. Vol. 1294. pp. 90–104.

[2] Bendlin R., Nielsen J.B., Nordholt P.S., Orlandi C. Receiver-deniable public-key encryption is impossible. Cryptology ePrint Archive, Report 2011/046, 2011. http://eprint.iacr.org/.

[3] Bo Meng. A Secure Internet Voting Protocol Based on Non-interactive Deniable Authentication Protocol and Proof Protocol that Two Ciphertexts are Encryption of the Same Plaintext. Journal of Networks. 2009. Vol. 4. No. 5. pp. 370–377.

[4] Bo Meng, Jiang Qing Wang. A Receiver Deniable Encryption Scheme. Proceedings the 2009 International Symposium on Information Processing (ISOP'09), Huangshan, China, August 21-23, 2009 of Networks. 2009. pp. 254–257.

[5] Hellman M.E., Pohlig S.C. Exponentiation Cryptographic Apparatus and Method, U.S. Patent # 4,424,414. 3 Jan. 1984.

[6] Ishai Yu., Kushilevits E., Ostrovsky R. Efficient Non-interactive Secure Computation. Advances in Cryptology – EUROCRYPT 2011. Lectute Notes in Computer Science. Springer - Verlag. Berlin, Heidelberg, New York, 2011. Vol. 6632. pp. 406–425.

[7] Menezes A.J., Oorschot P.C., Vanstone S.A. Applied cryptography. CRC Press, New York, London, 1996.- 780 p.

[8] Menezes A.J., Vanstone S.A. "Elliptic Curve Cryptosystems and Their Implementation", Journal of Cryptology, 1993, vol. 6, no 4, pp. 209–224.

[9] Moldovyan N.A., Moldovyan A.A., Shcherbacov V.A. Provably sender-deniable encryption scheme // Computer Science Journal of Moldova/ 2015, V.23. N. 1(67). pp. 62–71.

[10] Moldovyan A.A., Moldovyan N.A. Practical Method for Bi-Deniable Public-Key Encryption // Quasigroups and related systems. 2014. Vol. 22. pp. 277–282.

[11] Moldovyan A. A., Moldovyan N. A., Shcherbakov V. A. Bi-Deniable Public-Key Encryption Protocol Secure Against Active Coercive Adversary // Buletinul Academiei de Stiinte a Republicii Moldova. Matematica. 2014. N. 3 (76). pp. 23–29.

[12] Moldovyan N.A., Moldovyan A.A., Moldovyan D.N., Shcherbacov V.A. Stream Deiable-Encryption Algorithm Satisfying Criterion of the Computational Indistinguishability from Probabilistic Ciphering // Computer Science Journal of Moldova. 2016, V.24. N. 1. pp. 68–82.

[13] O'Neil A., Peikert C., Waters B. Bi-Deniable Public-Key Encryption. Advances in Cryptology - CRYPTO 2011. Lectute Notes in Computer Science. Springer - Verlag. Berlin, Heidelberg, New York, 2011. Vol. 6841. pp. 525–542.

[14] ElGamal T. A public key cryptosystem and a signature scheme based on discrete logarithms // IEEE Transactions on Information Theory. 1985. Vol. IT-31, no. 4. pp. 469–472.

[15] Wang C., Wang J. A Shared-key and Receiver-deniable Encryption Scheme over Lattice. Journal of Computational Information Systems. 2012. Vol. 8, No. 2, pp. 747–753.

Nicolai A. Moldovyan[1], Alexandr A. Moldovyan[2], Alexei V. Shcherbacov[3]

[1]Professor, St. Petersburg Institute for Informatics and Automation of Russian Academy of Sciences 14 Liniya, 39, St.Petersburg, 199178 Russia
Email: nmold@mail.ru

[2]Professor, St. Petersburg Institute for Informatics and Automation of Russian Academy of Sciences 14 Liniya, 39, St.Petersburg, 199178 Russia
Email: maa1305@yandex.ru

[3]M.Sc., Theoretical Lyceum C. Sibirschi, Lech Kaczyski str. 4, MD-2028, Chişinău Moldova
Email: admin@sibirsky.org

Proceedings of the Conference on Mathematical Foundations of Informatics
MFOI2016, July 25-29, 2016, Chisinau, Republic of Moldova

On computer aided knowledge discovery in logic and related areas

Andrei Rusu, Elena Rusu

Abstract

The present paper describes the architecture of a software for computer aided knowledge discovery dealing with the problems of functional expressibility of formulas in a nonstandard propositional logic. The achieved system rests on a distributed agent-based platform. Some software agents have been designed in order to solve some particular problems, the user interact with them via a protocol provided by an interface software agent. The methods used by some agents to solve some particular problems are based on technics inspired from nature: genetic programming. The agent framework JADE used for these objectives is a FIPA-complained open-source agent platform.

Keywords: non-classic propositional logic, expressibility of formulas, software agents, genetic programming, symbolic regression.

1 Introduction

The well-known problem of expressibility of a formula via a system of formulas in a given logic is considered [1, 2]. E. Post have investigated this problem and the related ones in the case of classical propositional logic [3].

The purpose of this paper is to present an agent-based system that can be used to asist the researcher in investigation of the above mentioned problem of expressibility and related to it ones in an arbitrary propositional calculus. As a software platform for this task we consider the open-source distributed software agent platform JADE [4].

2 Problems related to expressibility of formulas

Consider a propositional logical calculus L. It is defined usually by an alphabet of used symbols (propositional variables, i.e. $p, q, r, \ldots$, and a set of logical connectives, i.e. in the case of classical propositional logic $\&, \vee, \supset, \neg$), by formulas based on the given alphabet, by axioms (which are some formulas of the calculus L) and by some rules of inference (one well-known rule in classical propositional logic is the Modus Ponens rule, which allows passing from two formulas A and $A \supset B$ to the formula B) [5].

They say formula F is a consequence of the formulas $G_1, \ldots, G_n$ (formulas $G_1, \ldots, G_n$ are said to be hypotheses for F) in the logic L, denoted by

$$G_1, \ldots, G_n \vdash F,$$

if there is a sequence of formulas $F_1, \ldots, F_k$ with the following properties:

- F_k is actually formula F,
- for any index i, $i = 1, \ldots, k$ at least one of the following takes place:
  - F_i is an axiom of L,
  - $F_i \in \{G_1, \ldots, G_n\}$,
  - F_i is obtained from $F_{i_1}, \ldots, F_{i_j}$ by some rule of inference of L, where $\{i_1, \ldots, i_j\} \subseteq \{1, \ldots, i-1\}$

In the case $\{G_1, \ldots, G_n\} = \emptyset$ they say formula F is a theorem of L, denoted by

$$\vdash F.$$

Formulas F and G are said to be equivalent in L, denoted by $F \sim G$ if the following relations are valid in L:

$$F \vdash G,$$
$$G \vdash F.$$

Usually the equivalence of the formulas F and G in L is an equivalence relation over formulas of L [6].

They say [1, 2, 7] the formula F is expressible via a system of formulas Σ in the calculus L if there exists a finite sequence of formulas $F_1, \ldots, F_n$ (called expression of F in L) with the following properties:

- F_n is actually formula F,
- for any $i = 1, \ldots, n$ at least one of the following holds:
  - F_i is a variable;
  - F_i belongs to Σ;
  - F_i is obtained from some previous formulas in the sequence using:
    - the week substitution rule, which allows passing from two formulas A and B to the result of substitution of B in A instead of every occurence of a given variable p of A (denoted by $A[p/B]$, or $A[B]$);
    - the rule of replacement by an equivalent formula in L, which permits passing from formula A to formula B if formulas A and B are equivalent in L.

A system of formulas Σ is said to be complete with respect to expressibility of formulas in the calculus L if any formula of L is expressible via formulas of Σ. The system Σ is known to be pre-complete (relative to expressibility of formulas) in L if Σ is not complete in L, but for any formula F which is not expressible in L via Σ the system $\Sigma \cup \{F\}$ is already complete in L [1].

Now there are some general problems related to expressibility of formulas [2]:

- The problem of expressibility of a formula F via a system of formulas Σ in L is one mentioned for boolean funtions by Emil Post [3].

- The problem of completeness relative to expressibility in L of system of formulas Σ.
- different variations of the above problems regarding formula bases.

In order to approach the above problems there is usually a need to perform multiple various calculations.

3 Tools and main ideas

We use Genetic Programming (GP) to discover new knowledge about expressibility problems in logic L and combine this technique with an agent system based on Jade framework. Genetic Programming is a computational method inspired by biological evolution, which discovers computer programs tailored to a particular task [8]. GP maintains a population of individual programs. Computational analogs of biological mutation and crossover produce program variants. Each variant's suitability is evaluated using a user-defined fitness function depending on the concrete problem related to expressibility of formulas in the given logic L.

The main steps of a genetic programming algorithm are:

1. **Preparatory steps** to specify:
   (a) the set ot terminals, usually these are variables;
   (b) the set of primitive functions, i.e. initial formulas;
   (c) the fitness measure (for explicitly or implicitly measuring the fitness of individuals in the population);
   (d) certain other parameters for controlling the execution of the algorithm;
   (e) the termination criterion
2. **Execution steps** of the algorithm are:

(a) create in random fashion an initial population, i.e the 0-generation, of formulas composed of the available formulas;

(b) iterate the following steps on the population until termination criterion is fulfilled:

(a) Evaluate each formula in the population according to the fitness of the desired problem;

(b) Select necessary amount of individuals with a probability based on fitness to participate in the genetic operations at the next step;

(c) Create new individuals for the population by applying the following genetic operations with specified probabilities:

i. *Reproduction:* Copy the selected individual to the new population;

ii. *Crossover:* Create new individuals by random recombination of the randomly chosen parts of the selected individuals;

iii. *Mutation:* Create one new offspring formula for the new population by randomly mutating a randomly chosen part of one selected formula;

iv. *Architecture-altering operations:* Choose an architecture-altering operation from the available ones and create one new offspring formula for the new population by applying the chosen architecture-altering operation to one selected formula.

3. When termination criterion is fulfilled, the single best formula in the population produced during the execution is considered the result of the algorithm. If the execution is successful, the result formula may be a solution to the problem.

4 The agent system based on Jade

The idea of the built software is to provide intelligent software agents which can assist researcher in investigating problems related to problems of expressibility of formulas mentioned in previous section. A short introduction into the domain of agent-oriented software engineering can be consulted in [9].

Thus we consider agents for various particular tasks related to problems of expressibility of formulas in a propositional logic a researcher comes in touch with, such as:

- What models has the given propositional calculus?
- Does a given formula F of L conserves a given relation R on the given model?
- Which unary formulas are in a given class of formulas defined by a predicate on an algebra?
- Is the system of formulas Σ complete with respect to expressibility in the given logic L?
- Is the formula F expressible in the given logic L via the given system of formulas Σ?
- etc.

In order to answer the above mentioned questions we design various agents, some of them implement intelligent search based on genetic programming algorithm to do symbolic regression.

The designed agents respects the standards of The Foundation for Intelligent Physical Agents (FIPA), which is a body for developing and setting computer software standards for heterogeneous and interacting agents and agent-based systems [10]. As a software platform for the designed agents we consider the open-source Java Agent DEvelopment Framework, or JADE, which is a software framework for the development of intelligent agent, implemented in Java [4]. JADE provides:

- An environment where JADE agents are executed.
- Class Libraries to create agents using heritage and redefinition of behaviors.
- A graphical toolkit to monitoring and managing the platform of Intelligent Agent agents.

and is a distributed agents platform, which has a container for each host where agents are running. Additionally the platform has various debugging tools, mobility of code and content agents, the possibility of parallel execution of the behavior of agents, as well as support for the definition of languages and ontologies.

Depending on the concrete problem related to the expressibility of formulas in the given logic L the agents cooperate to solve it or to assist the researcher to solve it.

5 Conclusion

The developed multi-agent system for support for knowledge discovery in logic is robust, extensible, relatively simple to implement and new features can be easily added to it. Each agent can be implemented to support different intelligent behaviour depending on various situations. Since the system is based on Java it can run on almost all platforms.

The present research can also be used in other fields of interest, for example, for measuring the impact of science research for different actors in Moldova, using the established National Bibliometric Tool (`http://ibn.idsi.md/`).

Acknowledgments. Information Society Development Institute have supported part of the research for this paper in SCIFORM project Ref. Nr. 15.817.06.13A.

References

[1] A. V. Kuznetsov, *Über funktionelle Ausdrückbarkeit in superintuitionistischen Logiken.* Mat. Issled., vol. 6, no. 4 (1971), pp. 75–122.

[2] M. F. Ratsa, *Expressibility in propositional calculi.* Chişinău: Ştiinţa, 1991, 204 pp, ISBN: 5-376-00961-0.

[3] E. L. Post, *Two-valued iterative systems of mathematical logic.* Annals of Mathematics Studies, no. 5, Princeton University Press, Princeton, N.J., 1941, 122 p.

[4] JAVA Agent DEvelopment Framework, `http://jade.tilab.com/`.

[5] Mendelson, E., *Introduction to Mathematical Logic.* Chapman and Hall/CRC, 5th ed, 2010.

[6] W. J. Blok, D. Pigozzi, *Algebraizable Logics.* Memoirs of the American Mathematical Society, vol.77, no. 396 (1989), 89 p.

[7] A. V. Kuznetsov, *Analogs of the Sheffer stroke in constructive logic.* Sov. Math., Dokl., vol. 6 (1965), pp. 70–74.

[8] Koza, J. R. *Genetic programming: on the programming of computers by means of natural selection.* MIT Press, 1992.

[9] Wooldridge, M., Ciancarini, P. *Agent-oriented Software Engineering: The State of the Art.* in: First International Workshop, AOSE 2000 on Agent-oriented Software Engineering, Springer-Verlag New York, Inc., 2001, pp. 1-28.

[10] IEEE Foundation for Intelligent Physical Agents, `http://www.fipa.org/`.

Andrei RUSU[1,2], Elena RUSU[2]

[1]Information Society Development Institute
Email: andrei.rusu@idsi.md

[2]Ovidius University of Constanta
Email: agrusu@univ-ovidius.ro

[3]Technical University of Moldova
E–mail: elenarusu2006@yahoo.com

Proceedings of the Conference on Mathematical Foundations of Informatics
MFOI2016, July 25-29, 2016, Chisinau, Republic of Moldova

About spectrum of T_2-quasigroups

Alexandra V. Scerbacova, Victor A. Shcherbacov

Abstract

There exist medial T_2-quasigroups of any order of the form

$$2^{k_1}3^{k_2}5^{k_3}11^{k_4}17^{k_5}23^{k_6}p_1^{\alpha_1}p_2^{\alpha_2}\dots p_m^{\alpha_m},$$

where $k_1 \geq 2$, $k_2, \dots, k_6 \geq 1$, p_i are prime numbers of the form $6t+1$, $\alpha_i \in \mathbb{N}$, $i \in \{1, \dots, m\}$. Some other results about T_2-quasigroups are given.

2000 Mathematics Subject Classi cation: 20N05, 05B15

Key words and phrases: quasigroup, medial quasigroup, T_2-quasigroup, spectrum

1 Introduction

Definitions and elementary properties of quasigroups can be found in [1, 2, 18]. Most of presented here results are given in [20]. Quasigroups have some applications in cryptology [22]. The most usable in cryptology quasigroup property is the property of orthogonality of quasigroups [9].

V. D. Belousov [3, 4] (see also [10]) by the study of orthogonality of quasigroup parastrophes proved that there exists exactly seven parastrophically non-equivalent identities which guarantee that a quasigroup

is orthogonal to at least one its parastrophe:

$$x(x \cdot xy) = y \qquad (C_3 \text{ law}) \tag{1}$$
$$x(y \cdot yx) = y \qquad \text{of type } T_2 \text{ [3]} \tag{2}$$
$$x \cdot xy = yx \qquad \text{(Stein's 1st law)} \tag{3}$$
$$xy \cdot x = y \cdot xy \qquad \text{(Stein's 2nd law)} \tag{4}$$
$$xy \cdot yx = y \qquad \text{(Stein's 3nd law)} \tag{5}$$
$$xy \cdot y = x \cdot xy \qquad \text{(Schroder's 1st law)} \tag{6}$$
$$yx \cdot xy = y \qquad \text{(Schroder's 2nd law).} \tag{7}$$

The names of identities (3)–(7) originate from Sade's paper [19]. We follow [6] in the calling of identity (1).

All these identities can be obtained in a unified way using criteria of orthogonality of quasigroup parastrophes on the language of quasigroup translations [15, Theorem 8]. For example, identity (2), that guarantees orthogonality of a quasigroup $(Q, \cdot)$ and its $(2\,3)$-parastrophe, can be obtained from the following translation identity:

$$L_y^2 x = P_y x. \tag{8}$$

Using table about translations of quasigroup parastrophes [21, Table 1] we can rewrite identity (8) in the following "parastrophically equivalent" ([4]) forms:

$$\begin{aligned} &R_y^2 x = P_y^{-1} x, \\ &P_y^{-2} x = L_y^{-1} x, \\ &L_y^{-2} x = R_y x, \\ &R_y^{-2} x = L_y x, \\ &P_y^2 x = R_y^{-1} x. \end{aligned} \tag{9}$$

Passing to "standard" identities we obtain from the identities (9) the following identities that are parastrophically equivalent to the identity

(2):

$$\begin{aligned} &(xy \cdot y)x = y, \\ &(y\backslash x)(y/x) = y, \\ &y(y \cdot xy) = x, \\ &(yx \cdot y)y = x, \\ &x(y/(x/y)) = y. \end{aligned} \tag{10}$$

A quasigroup $(Q, \cdot)$ with the identity $x \cdot x = x$ is called idempotent. The set $\mathfrak{Q}$ of natural numbers for which there exist quasigroups with a property T, for example, the property of idempotency, is called the spectrum of the property T in the class of quasigroups. Often it is used the following phrase: spectrum of quasigroups with a property T. Therefore we can say that spectra of quasigroups with identities (3)–(7) were studied in [12, 6, 5, 17, 24, 8].

It is clear that the identity (2) and identities (10) have the same spectrum.

Idempotent models of the identity $(yx \cdot y)y = x$ can be associated with a class of resolvable Mendelsohn designs [5]. In [5] "it is shown that the spectrum of $(yx \cdot y)y = x$ contains all integers $n \geq 1$ with the exception of $n = 2, 6$ and the possible exception of $n \in \{10, 14, 18, 26, 30, 38, 42, 158\}$. It is also shown that idempotent models of $(yx \cdot y)y = x$ exist for all orders $n > 174$".

Here we study in the main spectrum of medial T_2-quasigroups. Such quasigroups can be easily constructed and therefore they can be used in cryptology. For example such quasigroups can be used by construction of crypto-codes [23]. See also [22].

2 Medial T_2-quasigroups

The problem of the study of T_2-quasigroups is posed in [3, 4]. In [25] the following proposition (Proposition 7) is proved. We formulate this proposition in the slightly changed form.

Theorem 2.1. *If a T_2-quasigroup $(Q, \cdot)$ is isotopic to an abelian group*

$(Q, \oplus)$, then for every element $b \in Q$ there exists an isomorphic copy $(Q, +) \cong (Q, \oplus)$ such that $x \cdot y = IL_b^3(x) + L_b(y) + b$, for all $x, y \in Q$, where $x + Ix = 0$ for all $x \in Q$.

De nition 2.2. *A quasigroup $(Q, \cdot)$ of the form $x \cdot y = \varphi x + \psi y + b$, where $(Q, +)$ is an abelian group, φ, ψ are automorphisms of the group $(Q, +)$, b is a fixed element of the set Q, is called T-quasigroup. If, additionally, $\varphi\psi = \psi\varphi$, then $(Q, \cdot)$ is called medial quasigroup [16, 1, 18, 2].*

Theorem 2.3. *A T-quasigroup $(Q, \cdot)$ of the form*

$$x \cdot y = \varphi x + \psi y + b \tag{11}$$

satisfies T_2-identity if and only if $\varphi = I\psi^3$, $\psi^5 + \psi^4 + 1 = (\psi^2 + \psi + 1)(\psi^3 - \psi + 1) = 0$, where 1 is identity automorphism of the group $(Q, +)$ and 0 is zero endomorphism of this group, $\psi^2 b + \psi b + b = 0$.

Proof. We rewrite T_2-identity using the right part of the form (11) as follows:

$$\varphi x + \psi(\varphi y + \psi(\varphi y + \psi x + b) + b) + b = y \tag{12}$$

or, taking into consideration that $(Q, +)$ is an abelian group, φ, ψ are its automorphisms, after simplification of equality (12) we have

$$\varphi x + \psi\varphi y + \psi^2\varphi y + \psi^3 x + \psi^2 b + \psi b + b = y. \tag{13}$$

If we put $x = y = 0$ in the equality (13), then we obtain

$$\psi^2 b + \psi b + b = 0, \tag{14}$$

where 0 is the identity (neutral) element of the group $(Q, +)$.

Therefore we can rewrite equality (13) in the following form:

$$\varphi x + \psi\varphi y + \psi^2\varphi y + \psi^3 x = y. \tag{15}$$

If we put $y = 0$ in the equality (15), then we obtain that $\varphi x + \psi^3 x = 0$. Therefore $\varphi = I\psi^3$, where, as well as above, $x + Ix = 0$ for all $x \in Q$.

Notice, in any abelian group $(Q,+)$ the map I is an automorphism of this group. Really, $I(x+y)=Iy+Ix=Ix+Iy$.

Moreover, $I\alpha=\alpha I$ for any automorphism of the group $(Q,+)$. Indeed, $\alpha x+I\alpha x=0$. From the other side $\alpha x+\alpha Ix=\alpha(x+Ix)=\alpha 0=0$. Comparing the left sides we have $\alpha x+I\alpha x=\alpha x+\alpha Ix$, $I\alpha x=\alpha Ix$, i.e., $\alpha I=I\alpha$.

It is well known that $I^2=\varepsilon$, i.e., $-(-x)=x$. Indeed, from equality $x+Ix=0$ using commutativity we have $Ix+x=0$. From the other side $I(x+Ix)=0$, $Ix+I^2x=0$. Then $Ix+x=Ix+I^2x$, $x=I^2x$ for all $x\in Q$.

If we put $x=0$ in the equality (15), then we obtain that

$$\psi\varphi y+\psi^2\varphi y=y. \tag{16}$$

If we substitute in the equality (16) expression $I\psi^3$ instead of φ, then we have $I\psi^5y+I\psi^4y=y$, $\psi^5y+\psi^4y=Iy$, $\psi^5y+\psi^4y+y=0$. The last condition can be written in the form $\psi^5+\psi^4+1=0$, where 1 is identity automorphism of the group $(Q,+)$ and 0 is zero endomorphism of this group.

It is easy to check that $\psi^5+\psi^4+1=(\psi^2+\psi+1)(\psi^3-\psi+1)$.

Converse. If we take into consideration that $\psi^2b+\psi b+b=0$, then from equality (13) we obtain equality (15). If we substitute in equality (15) the following equality $\varphi=I\psi^3$, then we obtain $\psi I\psi^3y+\psi^2I\psi^3y=y$, $\psi^4Iy+\psi^5Iy=y$ which is equivalent to the equality $\psi^5y+\psi^4y+y=0$. Therefore T-quasigroup $(Q,\cdot)$ is T_2-quasigroup. □

Corollary 2.4. *Any T-T_2-quasigroup is medial.*

Proof. The proof follows from equality $\varphi=I\psi^3$ (see Theorem 2.3). □

Corollary 2.5. *A T-quasigroup $(Q,\cdot)$ of the form*

$$x\cdot y=\varphi x+\psi y \tag{17}$$

satisfies T_2-identity if and only if $\varphi=I\psi^3$, $\psi^5+\psi^4+1=0$.

Proof. It is easy to see. □

Corollary 2.6. *A T-quasigroup* $(Q, \cdot)$ *of the form* $x \cdot y = \varphi x + \psi y + b$ *satisfies* T_2*-identity if* $\varphi = I\psi^3$, $\psi^2 + \psi + 1 = 0$.

Proof. The proof follows from Theorem 2.3 and the following fact: if $\psi^2 + \psi + 1 = 0$, then $\psi^5 + \psi^4 + 1 = 0$. □

Example 2.7. The following T_2-quasigroup is non-medial and therefore it is not a T-quasigroup (see Corollary 2.4). It is clear that this quasigroup is non-idempotent.

$*$	0	1	2	3	4	5	6	7	8
0	0	1	3	4	2	5	6	7	8
1	2	0	1	6	7	3	5	8	4
2	1	4	5	8	0	6	2	3	7
3	7	3	0	5	8	1	4	2	6
4	6	2	8	0	5	7	3	4	1
5	8	7	2	3	4	0	1	6	5
6	4	8	7	1	6	2	0	5	3
7	3	5	6	7	1	4	8	0	2
8	5	6	4	2	3	8	7	1	0

3 T_2-quasigroups from the rings of residues

We use rings of residues modulo n, say $(R, +, \cdot, 1)$, and Theorem 2.3 to construct T_2-quasigroups. Here $(R, +)$ is cyclic group of order n, i.e., it is the group $(Z_n, +)$ with the generator element 1. It is clear that in many cases the element 1 is not a unique generator element, $(R, \cdot)$ is a commutative semigroup [13].

Multiplication of an element $b \in R$ on all elements of the group $(R, +)$ induces an endomorphism of the group $(R, +)$, i.e., $b \cdot (x+y) = b \cdot x + b \cdot y$. If $g.c.d.(b, n) = 1$, then the element b induces an automorphism of the group $(R, +)$ and it is called an invertible element of the ring $(R, +, \cdot, 1)$.

Next theorem is a specification of Theorem 2.3 on medial T_2-quasigroups defined using rings of residues modulo n. We denote by

the symbol $\mathbb{Z}$ the set of integers, we denote by $|n|$ module of the number n.

Theorem 3.1. *Let $(Z_r, +, \cdot, 1)$ be a ring of residues modulo r such that $f(k) = (k^5 + k^4 + 1) \equiv 0 \pmod{r}$ for some $k \in \mathbb{Z}$. If $g.c.d.(|k|, r) = 1$, $k^2 \cdot b + k \cdot b + b \equiv 0 \pmod{r}$ for some $b \in Z_r$, then there exists T_2-quasigroup $(Z_r, \circ)$ of the form $x \circ y = -k^3 \cdot x + k \cdot y + b$.*

Proof. We can use Theorem 2.3. The fact that $g.c.d.(|k|, r) = 1$ guarantees that the multiplication on the number k induces an automorphism of the group $(Z_r, +)$. In this case the map $-k^3$ is also a permutation as a product of permutations. □

Example 3.2. Let $k = -3$. Then $f(-3) = (-3)^5 + (-3)^4 + 1 = -161 = -(7) \cdot (23)$. Therefore $-161 \equiv 0 \pmod{7}$ and $-161 \equiv 0 \pmod{23}$ and we have theoretical possibility to construct T_2 quasigroups of order $7, 23, 161$.

Case 1. Let $r = 7$. Then $k = -3 = 4 \pmod{7}$. In this case $-(k^3) = -(-3)^3 = 27 = 6 \pmod{7}$. It is clear that the elements 6 and 4 are invertible elements of the ring $(Z_7, +, \cdot, 1)$. Therefore quasigroup $(Z_7, *)$ with the form $x * y = 6 \cdot x + 4 \cdot y$ is T_2-quasigroup of order 7.

Check. We have $6x + 4(6y + 4(6y + 4x)) = y$, $70x + 24y + 96y = y$, $y = y$, since $70 \equiv 0 \pmod{7}, 120 \equiv 1 \pmod{7}$.

In order to construct T_2-quasigroups over the ring $(Z_7, +, \cdot, 1)$ with non-zero element b we must solve congruence $(-3)^2 \cdot b + (-3) \cdot b + b \equiv 0 \pmod{7}$. We have $7 \cdot b \equiv 0 \pmod{7}$. The last equation is true for any possible value of the element b. Therefore the following quasigroups are T_2-quasigroups of order 7: $x \circ y = 6 \cdot x + 4 \cdot y + i$, for any $i \in \{1, 2, \ldots, 5, 6\}$.

Case 2. Let $r = 23$. Then $k = -3 = 20 \pmod{23}$. In this case $-(k^3) = -(-3)^3 = 27 = 4 \pmod{23}$. It is clear that the elements 20 and 4 are invertible elements of the ring $(Z_{23}, +, \cdot, 1)$. Therefore quasigroup $(Z_{23}, *)$ with the form $x * y = 4 \cdot x + 20 \cdot y$ is T_2-quasigroup of order 23.

Check. We have $4x + 20(4y + 20(4y + 20x)) = y$, $4x + 80y + 1600y + 8000x = y$, $y = y$, since $8004 \equiv 0 \pmod{23}, 1680 \equiv 1 \pmod{23}$. This quasigroup is idempotent. Indeed, $4 + 20 = 24 \equiv 1 \bmod 23$.

In order to construct T_2-quasigroups over the ring $(Z_{23}, +, \cdot, 1)$ with non-zero element b we must solve congruence $(-3)^2 \cdot b + (-3) \cdot b + b \equiv 0 \pmod{23}$. We have $7 \cdot b \equiv 0 \pmod{23}$. This congruence modulo has unique solution $b \equiv 0 \bmod 23$, since $g.c.d.(7, 23) = 1$.

Case 3. Let $r = 161$. Then $k = -3 = 158 \pmod{161}$. Recall the number 161 is not prime. In this case $-(k)^3 = -(-3)^3 = 27 \pmod{161}$, $g.c.d.(27, 161) = 1$, the elements 158 and 27 are invertible elements of the ring $(Z_{161}, +, \cdot, 1)$. Therefore quasigroup $(Z_{161}, \circ)$ with the form $x \circ y = 27 \cdot x + 158 \cdot y$ is medial T_2-quasigroup of order 161.

Check. $27x + 4266y + 674028y + 3944312x = y$, $y = y$, since $3944339 \equiv 0 \pmod{161}$, $678294 \equiv 1 \pmod{161}$.

In order to construct T_2-quasigroups over the ring $(Z_7, +, \cdot, 1)$ with non-zero element b we must solve congruence $7 \cdot b \equiv 0 \pmod{161}$. It is clear that $g.c.d.(7, 161) = 7$. Therefore this congruence has 6 non-zero solutions, namely, $b \in \{23, 46, 69, 92, 115, 138\} = D$.

The following quasigroups are T_2-quasigroups of order 161: $x \circ y = 27 \cdot x + 158 \cdot y + i$, for any $i \in D$.

Example 3.3. We list some values of the polynomial f:

$f(-20) = -3039999, f(-19) = -2345777, f(-18) = -1784591,$
$f(-17) = -1336335, f(-16) = -983039, f(-15) = -708749,$
$f(-14) = -499407, f(-13) = -342731, f(-12) = -228095,$
$f(-11) = -146409, f(-10) = -89999, f(-9) = -52487,$
$f(-8) = -28671, f(-7) = -14405, f(-6) = -6479, f(-5) = -2499,$
$f(-4) = -767, f(-3) = -161, f(-2) = -15, f(-1) = 1, f(1) = 3,$
$f(2) = 49, f(3) = 325, f(4) = 1281, f(5) = 3751,$
$f(6) = 9073, f(7) = 19209, f(8) = 36865, f(9) = 65611,$
$f(10) = 110001, f(11) = 175693, f(12) = 269569, f(13) = 399855,$
$f(14) = 576241, f(15) = 810001, f(12) = 269569, f(17) = 1503379,$
$f(18) = 1994545, f(19) = 2606421, f(20) = 3360001.$

The set of prime divisors of the numbers of the set $\{f(-20), f(-19), \ldots, f(-1), f(1), \ldots, f(20)\}$ contains the following primes:

$$\{3, 5, 7, 13, 19, 23, 37, 43, 59, 61, 73, 101, 157, 211, 241, 307, 347,$$
$$421, 503, 719, 833, 977, 991, 1163, 1319, 2729, 3359, 5813, 6841\}.$$

It is possible to use the presented numbers for construction of T_2-quasigroups over the rings of residues.

Theorem 3.4. *There exist medial T_2-quasigroups of any prime order p such that $p = 6m + 1$, where $m \in \mathbb{N}$.*

Proof. We use Corollary 2.6. Let $(Z_p, +, \cdot, 1)$ be a ring (a Galois field) of residues modulo p, where p is prime of the form $6t+1$, $t \in \mathbb{N}$. Quadratic equation $\psi^2 + \psi + 1 = 0$ has two roots $h_1 = (-1 - \sqrt{-3})/2$ and $h_2 = (-1 + \sqrt{-3})/2$. Since p is prime, then $g.c.d(h_1, p) = g.c.d(h_2, p) = 1$.

It is known [11] that the number -3 is quadratic residue modulo any prime p such that $6m + 1$. Finally, if the number $(-1 - \sqrt{-3})$ is odd, then the number $(-1 - \sqrt{-3} + p)$ is even. □

We prove the fact that the number -3 is quadratic residue modulo any prime p such that $6m + 1$ additionally in the following

Lemma 3.5. *The number -3 is quadratic residue modulo odd prime p, if p can be presented in the form $6t + 1$, where $t \in \mathbb{N}$.*

Proof. For proving this fact we use information from [7, p. 187-188]. We represent prime p, $p > 2$, in the following form: $p = 4qt + r$, where $1 \leq r < 4q$, $g.c.d.(r, 4q) = 1$, q or $-q$ is a prime. The number q or $-q$ is quadratic residue modulo p if and only if

$$(-1)^{\frac{r-1}{2} \cdot \frac{q-1}{2}} \left(\frac{r}{q}\right) = 1,$$

where $\left(\frac{r}{q}\right)$ is Legendre symbol, or, speaking more formally, Legendre-Jacobi-Kronecker symbol.

If $r = 1$, then $(-1)^{\frac{1-1}{2} \cdot \frac{-3-1}{2}} \left(\frac{1}{-3}\right) = \left(\frac{1}{-3}\right) = 1$.

If $r = 5$, then $(-1)^{\frac{5-1}{2}\cdot\frac{-3-1}{2}}\left(\frac{5}{-3}\right) = \left(\frac{5}{-3}\right) = -1$.

If $r = 7$, then $(-1)^{\frac{7-1}{2}\cdot\frac{-3-1}{2}}\left(\frac{7}{-3}\right) = \left(\frac{7}{-3}\right) = 1$.

If $r = 11$, then $(-1)^{\frac{11-1}{2}\cdot\frac{-3-1}{2}}\left(\frac{11}{-3}\right) = \left(\frac{11}{-3}\right) = -1$.

Therefore prime p has the form $p = 12t + 1$ or $p = 12t + 7$. Combining the last equalities we have that $p = 6t + 1$. □

In order to construct T_2-quasigroups it is possible to use direct products of T_2-quasigroups. It is clear that direct product of T_2-quasigroups is a T_2-quasigroup.

It is possible to use the following arguments as well. Class of T_2 quasigroups is defined using T_2-identity and it forms a variety in signature with three binary operations, namely, with the operations $\cdot$, $/$, and $\backslash$ [13]. It is known that any variety is closed relative to the operator of the direct product [13].

Therefore we can formulate the following

Theorem 3.6. *There exist medial T_2-quasigroups of any order of the form $p_1^{\alpha_1}p_2^{\alpha_2}\dots p_m^{\alpha_m}$, where p_i are prime numbers of the form $6t + 1$, $\alpha_i \in \mathbb{N}$, $i \in \{1, \dots, m\}$.*

Notice in this section and in the following section examples of medial quasigroups of prime order of the form $6 \cdot t + 5$ (for example, $5, 11, 23, 341$) are given.

4 Examples of medial T_2-quasigroups

Using Mace4 [14] we construct the following examples of medial T_2-quasigroups.

$*$	0	1	2
0	0	1	2
1	2	0	1
2	1	2	0

$\boxtimes$	0	1	2	3
0	0	2	3	1
1	1	3	2	0
2	2	0	1	3
3	3	1	0	2

$\circ$	0	1	2	3	4
0	0	2	4	1	3
1	2	1	3	4	0
2	4	3	2	0	1
3	1	4	0	3	2
4	3	0	1	2	4

$\diamond$	0	1	2	3	4	5	6	7
0	0	2	4	1	6	3	7	5
1	6	1	5	2	0	7	3	4
2	7	4	2	5	3	6	0	1
3	4	7	0	3	5	1	2	6
4	5	3	6	7	4	2	1	0
5	2	0	7	6	1	5	4	3
6	3	5	1	4	7	0	6	2
7	1	6	3	0	2	4	5	7

$\bullet$	0	1	2	3	4	5	6	7	8	9	10
0	0	2	4	1	5	3	8	6	9	10	7
1	6	1	9	7	0	2	3	4	10	8	5
2	8	6	2	4	10	0	5	3	1	7	9
3	7	8	0	3	1	10	9	5	6	4	2
4	9	5	8	0	4	7	1	10	3	2	6
5	10	0	3	6	9	5	7	8	2	1	4
6	2	9	7	10	3	4	6	1	5	0	8
7	1	10	5	8	2	9	4	7	0	6	3
8	4	7	10	5	6	1	2	9	8	3	0
9	5	3	6	2	7	8	10	0	4	9	1
10	3	4	1	9	8	6	0	2	7	5	10

Moreover, we constructed medial T_2-quasigroups of order 17.

Lemma 4.1. *There exist medial T_2-quasigroups of order 2^k for any $k \geq 2$.*

Proof. If we suppose that the number k is even, i.e., $k = 2t$, then for the proof it is sufficient to take the direct product of t copies of medial T_2-quasigroup of order 2^2 with the operation $\boxtimes$.

We recall that T_2-quasigroup with the operation $\diamond$ is medial quasigroup of order 2^3. If we suppose that the number k is odd, $k \geq 5$, then the number $(k-3)$ is even and see previous case. □

Example 4.2. There exists medial T_2-quasigroup of order 2^{11} since $11 = 2 \cdot 1 + 3 \cdot 3$.

Combining Lemma 4.1, Theorem 3.6, and constructed examples we formulate the following

Theorem 4.3. *There exist medial T_2-quasigroups of any order of the form*

$$2^{k_1}3^{k_2}5^{k_3}11^{k_4}17^{k_5}23^{k_6}p_1^{\alpha_1}p_2^{\alpha_2}\dots p_m^{\alpha_m},$$

where $k_1 \geq 2$, $k_2, \dots, k_6 \geq 1$, p_i are prime numbers of the form $6t+1$, $\alpha_i \in \mathbb{N}$, $i \in \{1, \dots, m\}$.

References

[1] V.D. Belousov. *Foundations of the Theory of Quasigroups and Loops*. Nauka, Moscow, 1967. (in Russian).

[2] V.D. Belousov. *Elements of Quasigroup Theory: a Special Course*. Kishinev State University Printing House, Kishinev, 1981. (in Russian).

[3] V.D. Belousov. *Parastrophic-orthogonal quasigroups, Preprint*. Shtiinta, Kishinev, 1983. (in Russian).

[4] V.D. Belousov. Parastrophic-orthogonal quasigroups. Translated from the 1983 Russian original. *Quasigroups Relat. Syst.*, 13(1):25–72, 2005.

[5] F.E. Bennett. Quasigroup identities and Mendelsohn designs. *Canad. J. Math.*, 41(2):341–368, 1989.

[6] F.E. Bennett. The spectra of a variety of quasigroups and related combinatorial designs. *Discrete Math.*, 77:29–50, 1989.

[7] A.A. Buchstab. *Number Theory*. Prosveshchenie, 1966. (in Russian).

[8] D. Ceban and P. Syrbu. On qusigroups with some minimal ideteties. *Studia Universitatis Moldaviae. Stiinte Exacte si Economice*, 82(2):47–52, 2015.

[9] J. Dénes and A. D. Keedwell. *Latin Squares and their Applications*. Académiai Kiadó, Budapest, 1974.

[10] T. Evans. Algebraic structures associated with latin squares and orthogonal arrays. *Congr. Numer.*, 13:31–52, 1975.

[11] A.D. Keedwell and V.A. Shcherbacov. Construction and properties of (r,s,t)-inverse quasigroups, I. *Discrete Math.*, 266(1-3):275–291, 2003.

[12] C.C. Lindner, N. S. Mendelsohn, and S. R. Sun. On the construction of Schroeder quasigroups. *Discrete Math.*, 32(3):271–280, 1980.

[13] A.I. Mal'tsev. *Algebraic Systems*. Nauka, Moscow, 1976. (in Russian).

[14] W. McCune. *Mace 4*. University of New Mexico, www.cs.unm.edu/mccune/prover9/, 2007.

[15] G.L. Mullen and V.A. Shcherbacov. On orthogonality of binary operations and squares. *Bul. Acad. Stiinte Repub. Mold., Mat.*, (2):3–42, 2005.

[16] P. Němec and T. Kepka. T-quasigroups, I. *Acta Univ. Carolin. Math. Phys.*, 12(1):39–49, 1971.

[17] M.J. Pelling and D.G. Rogers. Stein quasigroups. I: Combinatorial aspects. *Bull. Aust. Math. Soc.*, 18:221–236, 1978.

[18] H.O. Pflugfelder. *Quasigroups and Loops: Introduction*. Heldermann Verlag, Berlin, 1990.

[19] A. Sade. Quasigroupes obéissant á certaines lois. *Rev. Fac. Sci. Univ. Istambul*, 22:151–184, 1957.

[20] A.V. Scerbacova and V.A. Shcherbacov. About spectrum of T_2-quasigroups. Technical report, arXiv:1509.00796, 2015.

[21] V.A. Shcherbacov. On definitions of groupoids closely connected with quasigroups. *Bul. Acad. Stiinte Repub. Mold., Mat.*, (2):43–54, 2007.

[22] V.A. Shcherbacov. Quasigroups in cryptology. *Comput. Sci. J. Moldova*, 17(2):193–228, 2009.

[23] V.A. Shcherbacov. Quasigroup-based hybrid of a code and a cipher. In *ICT Innovations 2012, Secure and Intelligent Systems, (Editors Smile Markovski and Marjan Gusev), Web proceedings, ISSN 1857-7288*, pages 411–418, 2012.

[24] Parascovia Syrbu and Dina Ceban. On π-quasigroups of type T_1. *Bul. Acad. Stiinte Repub. Mold. Mat.*, (2):36–43, 2014.

[25] P.N. Syrbu. On π-quasigroups isotopic to abelian groups. *Bul. Acad. Ştiinţe Repub. Mold. Mat.*, (3):109–117, 2009.

Alexandra V. Scerbacova[1], Victor A. Shcherbacov[2]

[1]Student, Gubkin Russian State Oil and Gas University
119991, Moscow, Leninsky Prospect, 65
Russia

Email: scerbik33@yandex.ru

[2]Dr., Institute of Mathematics and Computer Science
Academy of Sciences of Moldova
MD-2028, str. Academiei, 5, Chisinau
Moldova

Email: scerb@math.md

Structural Analysis of Industrial H-Type Hydraulic Press by Using Finite Element Method

Haşmet Çağrı Sezgen, Mustafa Tınkır

Abstract

In this study, structural analysis of an industrial 300 tons H-type hydraulic press is investigated for geometric optimization using finite element method. For this purpose, linear static analysis of press body parts is realized and maximum Von Misses stress locations, safety coefficients, maximum deformation results and required optimization locations are determined via ANSYS Workbench software. The obtained results are given in the form of the graphics.

Keywords: Structural analysis, hydraulic press, finite element method, stress, deformation, safety coefficients, optimization.

1 Introduction

Metal forming machines and presses are one of the classic applications of hydraulic science and they are used in many branches of the industry for high quality and series production. New materials, products and new manufacturing process are new areas of application for press technology. Major power is provided by using hydraulic in presses for effective and high-volume production. Today, hydraulic presses which are the most important part of the industrial hydraulic are used in iron and steel industry such as plastering, twisting, extrusion and forging process [1].

However, some structural problems occur in the hydraulic press manufacture and use. These critical problems are investigated and listed below:

1) Time to time cracking, plastic deformation and fracture problems are seen in press body and components. This situation prevents the

performance of press, it leads to the deformation and breakage of the molds and decreases the efficiency and capacity of hydraulic press.

2) Some values are given about power, capacities and fatigue behavior of the press but these values cannot be validated exactly.

3) Press types are increased because of every manufacturer produces different type and size of press for customer desire. But increased production rate disrupt standard production of this kind of machines. Also manufacturers use lots of raw materials to produce without using engineering calculation.

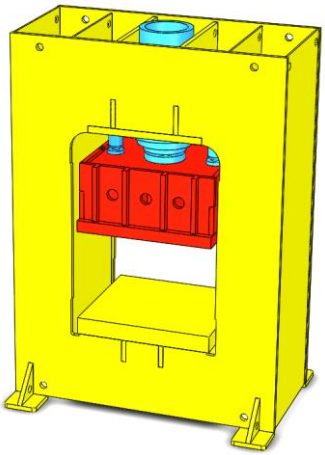


Figure 1. Computer aided design model of H-type hydraulic press.

In the light of all, these problems are considered and a large market, literature review and relationship with manufacturers are realized to help solving problems. As a result of investigations it can be said that engineering knowledge and realistic methods or formulas have not been used in press industry exactly. In literature review, different studies have been made about the hydraulic presses such as design and structural analysis. But the most important and similar studies are considered and given in this study. Arslan [2] studied the structural analysis of the body of an eccentric press using ANSYS software. Yağbasan [3] realized finite element analysis of C-type hydraulic press body. Raz et al. [4] analyzed stress-strain of hydraulic press components using finite element method. Zahalka [5] studied the modal analysis of a hydraulic press. Zhang et al

[6] have implemented structural optimization of the hydraulic press. Again, Zhang et al. [7] investigated mechanical analysis of the cylinder block of a hydraulic press.

In this paper, an industrial 300 tons H-type hydraulic press of manufacturer company given in Figure 1 is chosen and structural analysis of press is realized for geometric optimization. For this aim, linear static analysis of press body parts is realized and maximum Von Misses stress locations, safety factors, maximum deformation results and required optimization locations are determined via ANSYS Workbench finite element software. The obtained results are useful and realistic for chosen press manufacturer company to decrease using raw material for press production.

2 Structural Analysis

The relationship between external forces and displacements can be described as linear equations to solve static problems using finite element method. Spring can be used for elastic problem. Spring force is the product of displacement and spring constant. In the solutions of the finite element method, according to deformation and external forces the stiffness matrix can be written as:

$$f = k.x \tag{1}$$

where, k is the global stiffness matrix and x is the displacement vector.

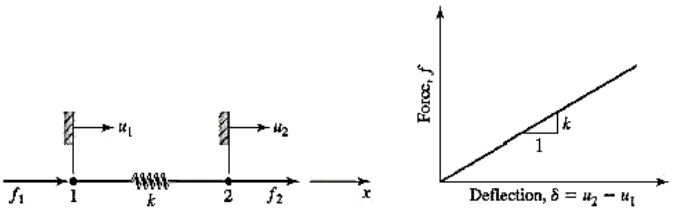


Figure 2. Relation between spring force, deflection and stiffness.

According to Figure 2, u_1 and u_2 are displacements that occur in the spring forces f_1 and f_2 applied to spring and k represents the spring constant.

Accordingly, net displacement formed in the spring is written as:

$$\delta = u_2 - u_1 . \tag{2}$$

External force applied to spring is described as:

$$f = k.\delta = k.(u_2 - u_1). \tag{3}$$

Applied forces can be written separately as:

$$f_1 = -k.(u_2 - u_1). \tag{4}$$
$$f_2 = -k.(u_2 - u_1).$$

These equations can be written as matrix form:

$$\begin{bmatrix} k & -k \\ -k & k \end{bmatrix} \begin{Bmatrix} u_1 \\ u_2 \end{Bmatrix} = \begin{Bmatrix} f_1 \\ f_2 \end{Bmatrix}. \tag{5}$$

When the initial equation is considered:

$$\begin{Bmatrix} f_1 \\ f_2 \end{Bmatrix} = [k_e] \begin{Bmatrix} u_1 \\ u_2 \end{Bmatrix}.$$
$$\{F\} = [K]\{X\}. \tag{6}$$

the stiffness matrix can be written as follow :

$$[k_e] = \begin{bmatrix} k & -k \\ -k & k \end{bmatrix}. \tag{7}$$

Assumptions for linear static finite element analysis are as follows:

- [K] is the global stiffness matrix, is constant.
- linear elastic material behavior is assumed.
- small deflection theory is used.
- {F} is the global load vector, is statically applied.
- No time-varying forces are considered.
- No damping effect is considered.

- Young's modulus and Poisson's ratio are always necessary for the linear static analysis.
- if there are the forces related inertia, mean density is needed.
- if a thermal load is applied, thermal expansion coefficient is required.
- stress-strain boundaries are required for safety coefficients

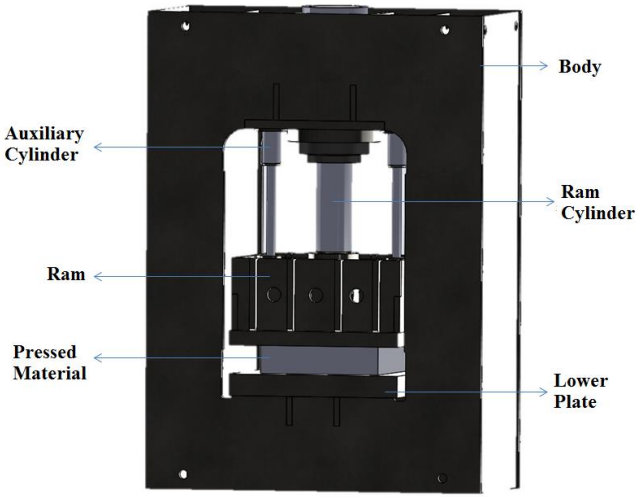


Figure 3. Hydraulic press main components.

The structural analysis of hydraulic press is realized, weak and more strength areas of the press body parts are determined using finite element method according to these assumptions. Also ANSYS Workbench finite element software is used in analysis. Computer aided design (CAD) model of the proposed hydraulic press is created in SolidWorks CAD program before the analysis and required shape optimizations of cad model are done for analysis. The connection holes, teeths and hydraulic adapter of press are removed from cad model. Some radius disturbed mesh geometry are removed. Moreover separate surfaces are drawn on touching parts of assembly components to superpose mesh geometry. Also ¼ symmetry is utilized to facilitate the solution. The mathematical model of hydraulic press given in Figure 2 is obtained by meshing cad model.

So cad model is divided into two identical or equivalent to close to geometric shapes. In this study, the mathematical model is achieved by using quadrilateral and triangular elements in ANSYS Workbench software and structural analysis is performed according to this model. The details of the mesh structure used in modeling are as follows:

For press body;

- ✓ maximum element size 36 mm.
- ✓ maximum face size 18 mm.
- ✓ minimum element size 2 mm.
- ✓ the growth rate is 1.85
- ✓ normal inclination angle is 45 degrees.
- ✓ mesh method is Sweep and Tetrahedrons.

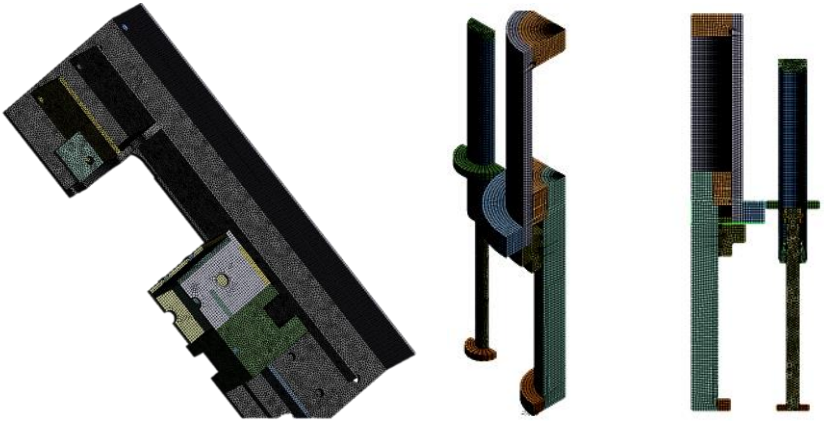


Figure 4. Mesh geometry of auxiliary and ram cylinders of the press.

For cylinders;

- ✓ maximum element size 12 mm.
- ✓ maximum face size 6 mm.
- ✓ minimum element size 2 mm.

- ✓ the growth rate is 1.85
- ✓ normal inclination angle is 45 degrees.
- ✓ mesh method is Sweep and Tethrahedrons

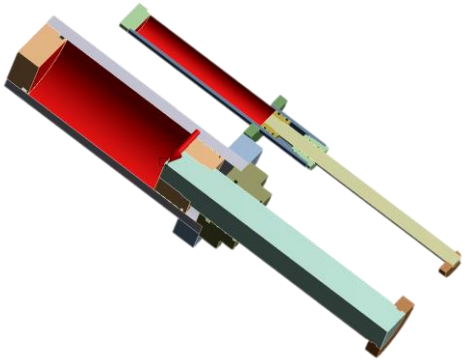


Figure 5. The applied pressure of the cylinder parts.

It is shown in Figure 4, appropriate mathematical model is obtained which contains overlapping part of close dimensional face. 250 bar hydraulic oil pressure is applied to drive the cylinder rod after mathematical modeling. In Figure 5 the pressure, formed in the piston top face, the upper face of the rod shaft, the cylinder inner face of the back cover and the inner face of the sleeve, are shown. Also the analysis model is simplified by carrying reaction forces instead of modeling material.

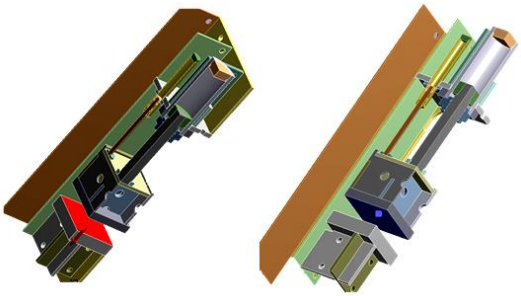


Figure 6. Assumption of pressing material.

Defined frictionless support to ram head and defined per load to bottom plate are shown in Figure 6. After loading the model the boundary conditions are determined and large deflections which are one of our boundary conditions for structural analysis is prevented. A screw connection is defined in the lower leg of press and all axes other than the direction of the normal are released (Frictionless support). The frictional connection is defined between barrel and piston, rod shaft and throat because the friction small displacements are perpendicular to surface normal. Fixed faces are shown in Figure 7.

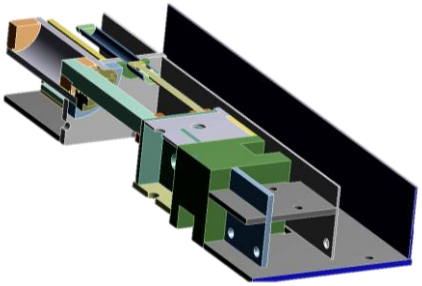


Figure 7. Fixed faces.

3 Results

To be able to comment on the modeling results, firstly stress results are obtained. The critical regions and safety factors are determined using Von Misses stress results. Also displacement results are considered to verify analysis according to given boundary conditions. In Figure 8, Von Misses stress and the displacement results of the press are given. Von Misses stress results in critical areas of the body are given in Figures 9 to 13. According to these results, the minimum safety coefficient for the outer wall layer is 12. For the front wall plate the minimum safety coefficient is obtained as 2.12. The minimum safety coefficients are for upper and rib platinum, inner wall and cylinder platinum 2.08, 2.89, 2.69 and 2.66 respectively. Moreover 1.98 and 4.44 safety coefficients are obtained for first and second lower platinum. Stress of the outer wall of press body is lower and no loads occur in this region. Thinner sheet metal can be used in this area but this time the buckling effect makes noise due to displacements while press running.

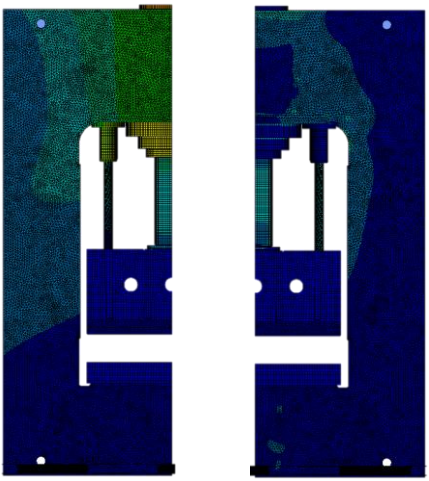


Figure 8. Von Misses stress and the displacement results of the press.

The stress as 90MPa is determined on the window radius of the front wall platinum of the body. This value can be decreased by radius optimization or rib assembly for this region. In addition, high stress rises due to the harsh design transition in connection ports.

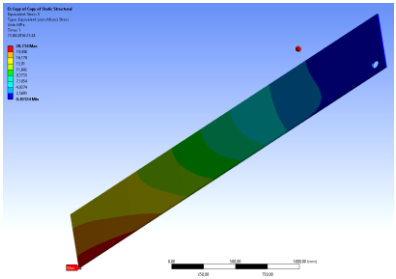


Figure 9. Critical region Von Misses stress results of the outer wall sheet metal of the body.

In connection with the upper platinum of body, the stress is obtained as 115 MPa. By smoothing, the transition stress can be reduced. Also 95 MPa. stress is found in the hole for transferring press. By changing the location of this hole stress can be reduced. Body rib platinum has the same stress with the upper platinum. Stress is determined as 80MPa in the inner

walls of the body. This stress can be decreased with parametric optimization in this area. Then part of thickness can be reduced via topological optimization. The stress as 90MPa of connection port of the cylinder can be decreased by using a radius transition.

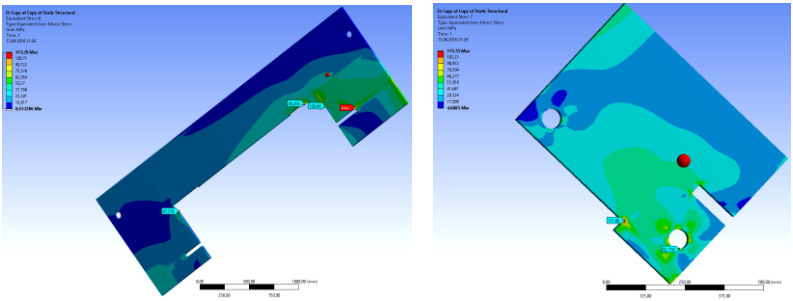


Figure 10. Von Misses stress results of the front wall and upper platinum of the body.

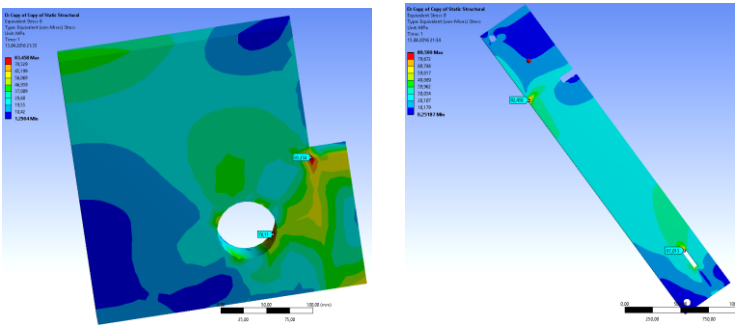


Figure 11. Von Misses stress results of the rib platinum and inner wall sheet metal of the body.

The lower platinum of the body has a stress as 120 MPa. With the design optimization in this area stress can be lowered. The stress as 50MPa occurred in the other sub platinum connection port of the body and the transfer hole can be decreased by design changes and further thickness can be reduced via topological optimization.

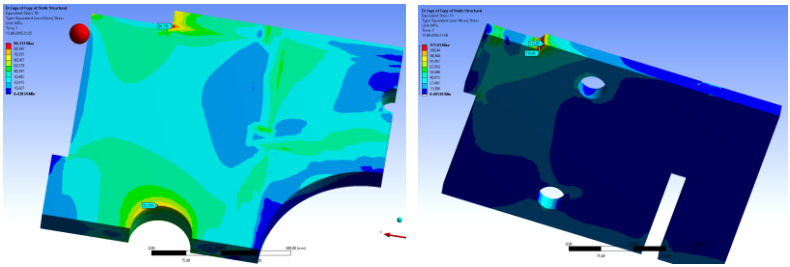


Figure 12. Von Misses stress results of cylinder port platinum and lower platinum 1 of the body.

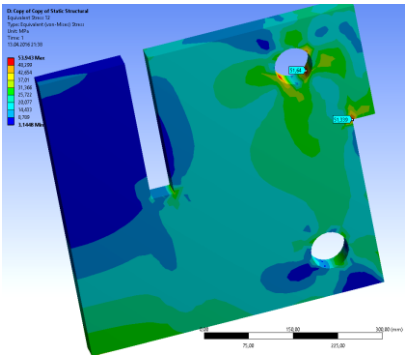


Figure 13. Von Misses stress results of lower platinum 2 of the body.

4 Conclusion

In this paper, an industrial 300 tons H-type hydraulic press is chosen and structural analysis of press body parts is realized for geometric optimization. For this aim, linear static analysis is performed and maximum Von Misses stress locations, safety coefficients, maximum displacement results and required optimization locations are determined via ANSYS Workbench finite element software. The obtained results are useful and realistic for chosen press manufacturer company to decrease using raw material for press production. The main contribution of the

paper is that press company verified analysis results with their experiences, changed design parameters and sheet metal thickness of same type hydraulic press for lower cost production. The obtained results can be improved according to fatigue analysis in the future works by topological optimization.

Acknowledgments. This work has supported by the Coordinatorships of Necmettin Erbakan University Scientific Research Projects. Also many thanks to Hidroliksan Halim Usta Press Company.

References

[1] Taş, B., "Hydraulic press design", Master thesis of Institute of Science of Gazi University, (2008), 90 pages.
[2] Arslan, O., "Analysis of an eccentric press body via ANSYS", Thesis of Mechanical Engineering Dept. of Dokuz Eylül University, (2009), 35 pages.
[3] Yağbasan, O., "Analysis of a C-type press body using finite element method", Master thesis of Mechanical Eng. Dept. of Eskişehir Osmangazi University, (2010), 77 pages.
[4] Raz, K. and Vaclav, K., 2014. Using of a Hydraulic Press in Production and Manufacturing of Large Rings. Procedia Engineering, 69, 1064–1069.
[5] Zahalka, M., 2014. Modal Analysis of Hydraulic Press Frames for Open Die Forging. Procedia Engineering, 69, 1070–1075.
[6] Zhang, W., Wang, X., Wang, Z. and Yuan, S., 2014. Structural optimization of cylinder-crown integrated hydraulic press with hemispherical hydraulic cylinder. Procedia Engineering, 81, 1663-1668.
[7] Zhang, W.W., Wang X.S., Wang, Z.R., Yuan, S.J., He, Z.B., Liu G. and Dai, K., 2015. Mechanical analysis on the cylinder-crown integrated hydraulic press with a hemispherical cylinder. Journal of Mechanical Engineering Science, 299(3), 407-416.

Haşmet Çağrı Sezgen [1], Mustafa Tınkır [2]

[1]Mechanical Engineering Department of Institute of Science of Necmettin Erbakan University, Konya-TURKEY
E-mail: cagrisezgen@gmail.com

[2]Mechanical Engineering Department of Necmettin Erbakan University, Konya-TURKEY
E-mail: mtinkir@konya.edu.tr

Performance evaluation of the evacuation system by Generalized Stochastic Petri nets

Inga Titchiev

Abstract

The aim of this article is to perform a quantitative analysis of the evacuation system by using Generalized Stochastic Petri nets and capturing all the properties and characteristics related to its dynamics.

Keywords: modeling, Petri nets, properties verification, quantitative analysis

1 Introduction

To check properties of distributed systems various methods can be used. Petri nets is one of the methods which demonstrated good results. For a more accurate modeling it is required to define a configuration of the system, intended to work in a certain context. It should correspond certain performance restrictions. Performance restrictions aim to ensure functional characteristics in the current context related to response time. In particular, this study is focused on quantitative investigations related to dynamics of the modeled system.

Social disaster can lead to other accidents and catastrophes and it may be necessary to keep human health and in some cases, human life, and for these it will be opportune to evacuate inhabitants in useful time.

The formalism of Petri net can be applied in the both theoretical and practical ways. In order to surprise as close as possible modeled real systems, the classical Petri net has been extended with the notion of time [1,3]. Petri nets are a powerful modeling technique because they can be

used to model complex systems and to verify if the modeled systems satisfy some criteria.

In order to perform a case study of extended evacuation system we used analysis modules of PIPE [2] and obtained properties and characteristics of them.

In this way the formal method like Petri nets becomes an important tool for detecting, monitoring, modelling and mitigating social disasters [5,6] caused by actions of different nature.

2 Generalized Stochastic Petri Nets

We will use the Generalized Stochastic Petri Nets (GSPN) [4] to perform quantitative analysis. They are characterized by two types of transitions:

1. *Stochastic transitions*: associated with an exponentially distributed firing delay.
2. *Immediate transitions*: associated with a null firing delay.

Formally, a GSPN can be defined as follows:

GSPN = (P; T; Π; I; O; H; M_0; W),

where

- P is a set of places;
- T is a set of transitions, $P \cap T = \varnothing$;
- I; O; H : $T \rightarrow N$ ($N = P \cup T$), are the input, output and inhibition functions;
- $M_0 : P \rightarrow N$ is the initial marking;
- $\Pi : T \rightarrow N$ is the priority function that associates the lower priorities to timed transitions and higher priorities to immediate transitions.
- W : $T \rightarrow R$ is a function that associates a real value to the transitions, w(t) is:
    - a (possibly marking dependent) rate of a negative exponential distribution specifying the firing delay, when transition t is a timed transition (represented by a hollow rectangle).
    - a (possibly marking dependent) firing weight, when transition t is immediate (represented by a filled rectangle).

When a new marking is reached, if only timed transitions are enabled, this marking is called *tangible*; if at least one immediate transition is enabled, the marking is called *vanishing*.

The selection of which transition will fire is based on the priorities and weights. First, the set of transitions with the highest priority is found and if it contains more than one enabled transition, the selection is based on the rates or weights of the transitions according to the expression:

$$P\{t\} = \frac{w(t)}{\sum_{t' \in E(M)} w(t')} , \qquad (1)$$

where *E(M)* is the set of transitions enabled at the marking *M*, i.e. the set of enabled transitions with the highest priority.

3 Evacuation system

Suppose that there is a building as it is specified in Fig. 1. M1-M8 are the rooms, M11-M81 are the doors. Petri Net of this building, is given in Fig. 2.

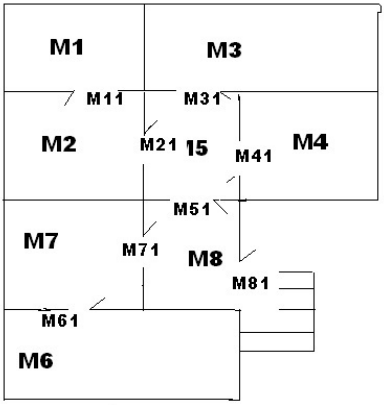


Figure 1. Building plan.

Rooms are modeled using places R1-R8, doors are modeled using places D11-D81, movings from the rooms to the doors are modeled by timed transitions t0-t7, movings from the doors into the rooms are modeled by immediate transition t8-t14. Each inhabitant is modeled by one token, respectively. The people are accumulated in the places. The

transfer function is used, which takes into account the time spent in the queue (moving time of human in the room) and the density and flow rate. The initial marking is $M_0 = (1,1,0,0,0,0,1,0,3,0,2,0,1,0,0,0)$.

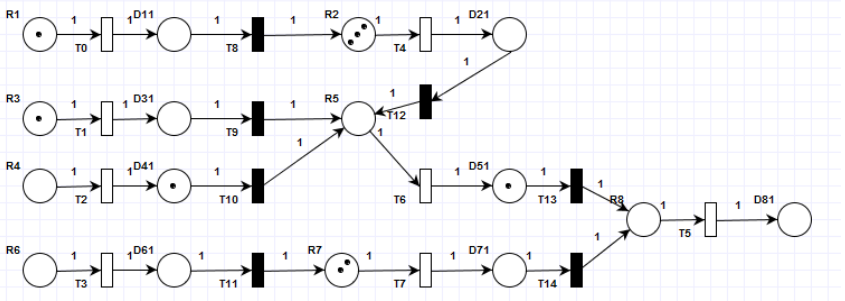


Figure 2. GSPN which models the building from Fig. 1.

4 Performance evaluation

After the simulation we obtained the results from which it is observed the exponential growth of the number of tangible states.

Table 1. Tangible states

Nr.	Number of people in rooms	Number of tangible states	Number of arcs in reachability graph
1	4	242	364
2	9	1456	2389
3	15	1888	3086

The average number of tokens (people) were also obtained. The number of people from the initial state is constant.

Petri net simulation results

Place	Average number of tokens	95% confidence interval (+/-)
R1	0.35	0.2147
R3	0.1	0.21825
R4	0	0
R6	0	0
D11	0.025	0
D31	0.025	0
D41	0	0.024
D61	0	0
R2	0.475	0.93823
R5	2.375	1.35811
R7	0.55	0.45015
D21	0.1	0
D51	0.175	0.024
D71	0.05	0
R8	3.325	1.37582
D81	1.45	1.04339

Figure 3. Average number of tokens in places

Throughput of Timed Transitions

Transition	Throughput
T0	0.28205
T1	0.07692
T2	0.28205
T3	0.28205
T4	0.28205
T5	0.35897
T6	0.35897
T7	0.07692

Figure 4. Throughput of timed transitions

The net is bounded, have a finite set of states (1124 states) which lead to a finite number of steps necessary for evacuation. Also it is safe, transitions do not influence each other, each place works independently of one another. It is conservative, the number of people is constant, new people do not appear, all 9 people from the initial state have been accumulated in

the last place D81. The net is without deadlock, it means that there are no persons who are unable to evacuate.

5 Conclusion

In this study, a method of Generalized Stochastic Petri Nets was proposed for simulation system that represents emergency evacuation of people in case of social disaster. This method allows checking such properties as boundedness, conservativeness, deadlock, safeness.

Acknowledgments. The work is performed as a part of the project: ”Modeling and Mitigation of Social Disasters Caused by Catastrophes and Terrorism” supported by NATO.

References

[1] Aalst van der W.M.P., *Interval Timed Coloured Petri Nets and their Analysis*. In M. Ajmone Marsan, editor, Application and Theory of Petri Nets, v.691 of Lecture Notes in Computer Science, Springer-Verlag, Berlin, (1993), pp. 453-472.

[2] Akharware, N. PIPE2: *Platform Independent Petri Net Editor*. http://pipe2.sourceforge.net/documents/PIPE2-Report-20050814.pdf, (2005).

[3] Berthomieu B. and Diaz M., *Modelling and verification of time dependent systems using Time Petri Nets*. IEEE Transactions on Software Engineering, 17(3), March (1991), pp. 259-273.

[4] Gutuleac E., *Evaluarea performantelor sistemelor de calcul prin Retele Petri stochastice*, Editura “Tehnica Info”, Chisinau, (2004), 276 p.

[5] Takashi M., Yoshifumi N., Yasuhiro F.a and Atsushi M., *Development of Tsunami refuge PETRI-NET simulation system utilizable in independence disaster prevention organization*, The 14th World Conference on Earthquake Engineering October 12-17, Beijing, China, (2008).

[6] Titchiev I., *Petri nets to model disaster prevention*, Proceedings of the Workshop on Foundations of Informatics, August 24-29, Chisinau, Republic of Moldova, (2015), pp. 445-449.

Inga Titchiev

Institute of Mathematics and Computer Science
E-mail: inga.titchiev@gmail.com

Complementing Tweets Sentiment Analysis with Semantic Roles

Diana Trandabăț, Adrian Iftene

Abstract

Slowly but surely, social media replaced the traditional sources of information: people's need to be constantly updated changed our behavior from buying a newspaper or watching TV, to using a Facebook or Twitter account to visualize, in a customizable manner, the day's hottest news, with the bonus of being able to also comment on them. This paper presents a method to identify a tweet's polarity (negative, positive, neutral) using SentiFrameNet, a naïve Bayes classifier and an off-the-self semantic role labeling API.

Keywords: natural language processing, semantic roles, sentiment analysis.

1 Introduction

Social media sites gained their popularity due to the "freedom" of expression they induce in people's mind: being able to post real time messages about your opinions on whatever topic you come across, discuss political and social decisions, complain, express gratitude or exchange impressions about products you use in everyday life.

Texts shared through social media applications offer us the information that we need: for example, the reviews of a product provide us useful information about its advantages and disadvantages, while the text of an advertisement invites us to eat at the new Chinese restaurant in town.

As huge amounts of texts become available through social media, a challenging task concerns the organization and processing of this information to extract knowledge. Natural language processing tools trained on large news corpora have usually problems when applied to unstandardized social media inputs, mainly due to the fact that social media content can appear in various forms (Becker et al., 2012), from photos and video updates to news, offers and literary works, and various informal formats.

Twitter is micro-blogging platform where people can send messages to one or multiple users, follow friends and read messages without much difficulty. Twitter messages, commonly known as tweets, are limited to 140 characters, and frequently include hash-tags (labels which should make it easier for users to find messages with similar content), all in one making Twitter analysis charming.

The remaining of this paper is structured as follows: Section 2 provides an overview of existing free online state of the art applications for sentiment analysis of social media, Section 3 briefly discusses the semantic frames theory, Section 4 presents our approach for analyzing social media opinions using semantic roles, Section 5 analyses our system's performance and draws conclusions.

2 State of the art

Specific processing tools (such as POS taggers or anaphora resolution systems), score a higher performance if used on the same text type as the ones they were trained on. In other words, we will have better results if using a POS tagger trained on news corpora to analyze news texts, rather than speech transcripts.

Thus, the short dimension of tweets and their creative informal spelling have raised a new set of challenges to the natural language processing field. How to handle such challenges so as to automatically mine and understand the opinions and sentiments that people are communicating has been the subject of (Jansen et al., 2009; Kouloumpis et al., 2011, Russell 2013).

.

A list of functional applications developed until now on Sentiment Analysis and API's that have a great success over the internet is presented below:

Sentiment140 (formerly known as "Twitter Sentiment") allows the discovery of the sentiment associated to a brand, product, or topic on Twitter. The API (Go et al., 2009) uses a maximum entropy classifier, trained on a set of automatically extracted tweets. The training corpus of 1.600.000 tweets is created relying on the use of emoticons (tweets with happy smileys suggest a positive contents, while tweets with sad/anger smileys refer to negative contents). The API lets users classify tweets and integrate sentiment analysis functionality into their own websites or applications, using RESTful calls and responses formatted in JSON.

Werfamous[1] is another webservice offering sentiment search ability for a user selected term.

Sentiment Analysis with Python NLTK Text Classification: It can classify the text introduced on one of three groups: positive, negative or neutral. Using hierarchical classification neutrality is determined first, and sentiment polarity is determined second, but only if the text is not neutral. The NLTK Trainer is used to train classifiers for the sentiments based on twitter sentiment or movie reviews. NLTK (Bird et al., 2009) is a leading platform for building Python programs to work with human language data. It provides easy-to-use interfaces to over 50 corpora and lexical resources such as WordNet, along with a suite of text processing libraries for classification, tokenization, stemming, tagging, parsing, and semantic reasoning, etc.

DatumBox[2]: an OpenSource API that allows users to access the web services offered by DatumBox. These services include Sentiment Analysis on any post using a 3 point scale considering that the topic of the post is given.

AlchemyAPI (Turian, 2013) launched in 2009, is a company that uses deep machine learning to perform natural language texts processing (specifically semantic text analysis, including sentiment analysis) and computer vision (face detection and recognition) for its clients both over

[1] http://werfamous.com/

[2] http://blog.datumbox.com/datumbox-machine-learning-framework-0-7-0-released/

the cloud and on-site. AlchemyAPI offers programmers the possibility to enhance their systems with context-aware modules, thus extracting entities and the sentiment associated with them and from webpages or social media messages.

LexAlytics is a web platform for media monitoring, offering nice visualization tools and powerful document processing capabilities.

The presented system uses a self-trained Naive Bayes classifier, combined with the existing Alchemy-API[3] for the cases where the classifier's output score was below an empirically established threshold. Additionally, polarity and modulation information are extracted from SentiFrameNet (Ruppenhofer and Rehbein, 2012).

3 Semantic roles

All content elements of a language are seen as *predicates*, i.e. expressions which designate events, properties of, or relations between, entities. The *predication* represents the mechanism that allows entities to instantiate properties, actions, attributes and states. Linguistic expressions can be dependent or independent. The dependent linguistic expressions are usually different *phenomena*, while the independent ones are *individuals*. For example, the word `hat` can be understood outside any circumstance, time, or person, because it does not have to be attributed to anything or anyone: it is independent, thus an individual. On the contrary, if we consider the word `red`, the denotations for this word cannot be understood outside its association with an individual: `red hat`. In linguistic terms, the dependent phenomena are predicates, while individuals are arguments. The linking between a phenomenon and individuals is known as predication.

Predicates are not treated as isolated elements, but as structures, named semantic frames. Within the predicate frames, each entity (frame element) plays a role, called semantic role. Semantic roles represent in fact the semantic relations that connect individuals to phenomena, or in the linguistic terms, arguments to predicates. After establishing the

[3] Alchemy API was selected among all analyzed variants due to the large number of events per day included in the free plan (1000) and the very well documented interface.

semantic relations within the predicate frames, syntactic and pragmatic functions are added to each predicate frame element.

The semantic relations can be exemplified within the Commercial Transaction frame, whose actors include a *buyer*, a *seller*, *goods*, and *money*. Among the large set of semantically related predicates, linked to this frame, we can mention `buy`, `sell`, `pay`, `spend`, `cost`, and `charge`, each of which indexes or evokes different aspects of the frame. The verb `buy` focuses on the *buyer* and the *goods*, backgrounding the *seller* and the *money*; `sell` focuses on the *seller* and the *goods*, backgrounding the *buyer* and the *money*; `pay` focuses on the *buyer*, the *money*, and the *seller*, backgrounding the *goods*; and so on. The idea is that knowing the meaning of any of these verbs requires knowing what takes place in a commercial transaction and, to some extent, knowing the meaning of all the predicates involved in the frame. The knowledge and experience structured by the Commercial Transaction frame provides the background and motivation for the categories represented by these verbs.

The Berkeley FrameNet project (Baker et al., 1998) is a lexicographic research project which produced a lexicon containing very detailed information about the syntax – semantics relations of the English predicational words (verbs, nouns and adjectives), based on Frame Semantics and supported by corpus evidence. The key concept in the FrameNet method of annotation is a semantic frame, defined as a type of event or state in Fillmore (1985). Each frame has its own set of roles, called frame elements (FEs), the frame-evoking words are called lexical units (LUs). A complete description of the predicates in the *commerce frame* may also include information about their grammatical properties and the various syntactic patterns in which they occur (their subcategorization frames), which frame elements may be realized as the subject of the verb, or as its object, if there is one, and what will be the syntactic surface form of the other frame elements, which ones of these frame elements are mandatory to express the meaning of the sentence and which are optional and may be missing[4].

[4] Each verb allows for a set of mandatory semantic roles, called arguments, and a set of optional, circumstantial semantic constituents, named adjuncts. The adjuncts are not specific to a verb, describing the context of the process rather than the process itself.

SentiFrameNet (Ruppenhofer and Rehbein, 2012) offers an extended model of the frame semantic representation. In SentiFrameNet, all LUs that are inherently evaluative are associated with opinion frames. All instances of such lexical units will eventually be annotated. One benefit of connecting sentiment analysis with frame semantics is immediate access to a deeper lexical semantics.

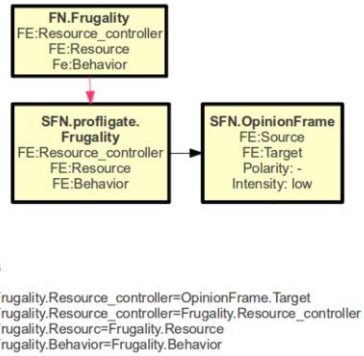


Figure 1. Example for SentiFrameNet for the adjective *profligate* [5]

SentiFrameNet tries to assign to semantic roles an opinion or target label, by adding opinion frames to FrameNet. Sentiment information is rather attached to specific LUs than to the whole frame, therefore SentiFrameNet considers two situations: splitting the frame to smaller frame segments until all LUs are sentiment-consistent, or putting each lexical unit into a newly created minimal frame which inherits the frame that the LU currently belongs to. Then, the newly derived frames are associated to a set of opinion frames which have Source and Target roles mapped to the frame semantic roles. In Figure 1, an analysis using a LU-specific frame is given for the adjective *profligate*.

4 Architecture

Being able to evaluate the opinion of the users is not a trivial matter. Evaluating their opinions requires performing Sentiment Analysis, which is the task of automatically identifying the polarity, the subjectivity and

[5] Image from (Ruppenhofer and Ines Rehbein, 2012b)

the emotional states of a particular document or sentence. It requires Machine Learning and Natural Language Processing techniques.

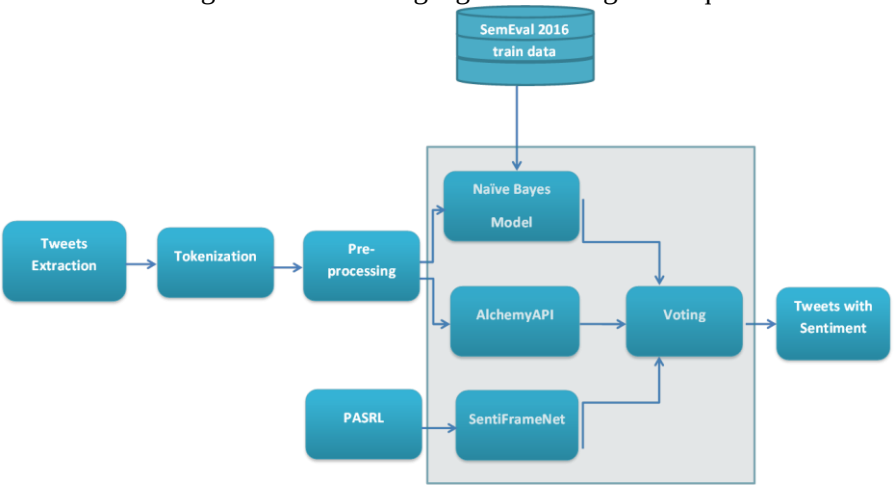


Figure 2. Architecture of the Sentiment Analyzer enhanced with semantic roles

The architecture for our system starts with extracting tweets. We used the developments and test tweets offered by SemEval 2016 task 4 (Nakov et al., 2016). The tweets are tokenized using simple punctuation delimiters. The next step consists in data cleaning and standardization. Thus, regular expressions have been built to: convert the texts to lowercase, discard words shorter than two characters, remove special diacritic signs, URLs, as well as symbols unsupported by the sentiment analyzer API (such as "?"). Users often include Twitter usernames in their tweets in order to direct their messages, using the @ symbol before the username (e.g. @radut), therefore a regex replaces all words that start with the @ symbol. Another modification proved to significantly reduce feature space, inspired by (Pang et al., 2002), removes duplicated vowels in the middle of the words (e.g. cooooool). Any letter occurring more than two times in a row is replaced with exactly two occurrences.

The sentiment extraction module used three different sentiment identification methods (a Naïve Bayes classifier, the AlchemyAPI and SentiFrameNet) and a voter. Thus, a Naïve Bayes classifier is trained on Semeval 2016 data, and a model is stored, containing all the necessary

information and probabilities used by the classifier. Similar to (Go et al, 2009 and Pang et al., 2002), the Naïve Bayes classifiers were trained using the following features: tokenized unigrams, emoticons, hashtags. Additionally, the Alchemy API runs on each tweet in order to extract information about the opinion expressed in the tweet. Simultaneously, semantic roles are identified using the in-house PASRL labeling system (Trandabat, 2011), and, for core semantic role, their polarity and modifiers are searched for in SentiFrameNet.

A voting module combines all this information, and allows the system to output a polarity on five-scale (extremely positive, positive, neutral, negative and extremely negative). Thus, this module checks how many agreements/disagreements are found in the results offered by the three different sentiment identification methods. We found that in only 14.9% of the cases, all three methods gave the same correct result. For 9.4% of the cases, the three services gave similar label, but failed to find the correct one. Out of these situations, almost 14% were mislabeled negative cases, and only 1.5% mislabeled positive tweets. However, in 30.6% of the cases, two of the services gave the same label, the correct one and in 78% of cases at least one classifier gave the right answer. These analyses lead to the decision to implement a simple voting module, based on a set of simple, empirically-derived rules.

5 Results

When analyzing the errors found in the classification of tweets, we noted that the system is positive-biased, i.e. it gave too many positive answers. Thus, out of the 29% negative instances wrongly classified, 77% were classified as positive, while 23 as neutral. Similarly, for the misclassified neutral instances, 89% were identified as positive, and 11% as negative. Table 1 presents the confusion matrix for the two categories.

Table 1. Confusion matrix for the five-scale task evaluated on test development data

	Negative	Positive
Negative	81,14	18,86
Positive	3,40	96,60

When identifying sentiments polarity on a 5-scale, the most misclassified category turns up to be the negative one. In case of doubt or when no other classification goes beyond our confidence score, the neutral classification was selected. The error matrix is presented in Table 2.

Table 2. Confusion matrix for the five-scale task, evaluated on test development data

	Very neg.	Neg.	Neutr.	Pos.	Very pos.
Very neg.	88,31	0,00	5,01	6,68	0,00
Neg.	0,00	54,37	12,90	32,73	0,00
Neutr.	0,00	1,59	36,27	62,14	0,00
Pos.	4,39	2,47	30,72	62,42	0,00
Very pos.	0,00	0,29	2,29	32,32	65,10

6 Conclusions

This paper presents a sentiment analysis solution, based on Naïve Bayes and Alchemy API, enhanced with semantic roles polarity extracted through SentiFrameNet.

In this version of the system, we did not use part-of-speeches, since initial tests showed that in this configuration, they bring more noise than relevant information, conclusion is also shared (for part-of-speeches) by (Pang et al., 2002). However, as further improvements, we intend to lemmatize the tweets before feeding them to our classifier, and use an external dictionary of sentiment valences in the voting module, to enhance our system's performance.

Acknowledgments: The system's development started with the participation of students from our Master in Computational Linguistics at Semeval 2016 - Shared Task 4: Sentiment Analysis in Twitter.

References

[1] Bird Steven, Klein E., Loper E., (2009) *Natural Language Processing with Python*, O'Reilly Media.

[2] Fillmore Charles J. (1985) Frames and the semantics of understanding. Quaderni di Semantica, 6.2:222–254.

[3] Go A., Bhayani R., Huang. L. (2009) Twitter Sentiment Classification using Distant Supervision, *Technical Report*.

[4] Jansen, B.J., Zhang, M., Sobel, K., Chowdury, A. (2009). *Twitter power: Tweets as electronic word of mouth*. Journal of the American Society for Information Science and Technology 60(11):2169-2188.

[5] Kouloumpis, E., Wilson, T., and Moore, J. (2011). *Twitter Sentiment Analysis: The Good the Bad and the OMG*! Proceedings of ICWSM. 2011

[6] Nakov P., Ritter A., Rosenthal S., Stoyanov V., Sebastiani F.(2016) *SemEval-2016 Task 4: Sentiment Analysis in Twitter*, Proc. of SemEval '16.

[7] Pang, Bo, Lillian Lee, and Shivakumar Vaithyanathan. (2002) "Thumbs up?: sentiment classification using machine learning techniques." Proceedings of EMNLP, pp. 79-86.

[8] Ruppenhofer Josef and Ines Rehbein. (2012a) *Semantic frames as an anchor representation for sentiment analysis* in Proceedings of the 3rd Workshop in Computational Approaches to Subjectivity and Sentiment Analysis. Association for Computational Linguistics.

[9] Ruppenhofer Josef and Ines Rehbein. (2012b) *Anchoring sentiment analysis in frame semantics*. Longer version of WASSA-paper, unpublished manuscript.

[10] Russell MA (2013) Mining the Social Web: Data Mining Facebook, Twitter, LinkedIn, Google+, GitHub, and More.

[11] Trandabăț Diana (2011) Mining Romanian texts for semantic knowledge, in Proceedings of Intelligent Systems and Design Application Conference, ISDA2011, Cordoba, Spain, ISSN: 2164-7143, ISBN: 978-1-4577-1676-8, DOI: 10.1109/ISDA.2011.6121799, pp. 1062-1066.

[12] Turian Joseph, (2013) Using AlchemyAPI for Enterprise-Grade Text Analysis, PhD Thesis.

Diana Trandabăț[1], Adrian Iftene[2]

[1]Affiliation/Institution: University "Al. I. Cuza" Iași, Romania
E-mail: dtrandabat@info.uaic.ro

[2]Affiliation/Institution: University "Al. I. Cuza" Iași, Romania
E-mail: adiftene@info.uaic.ro

Table of contents

Part 1. Invited papers

Part 2. Regular papers