Vladimir Andrunachievici Institute of Mathematics
and Computer Science

Logic & Artificial Intelligence

Edited by
Svetlana Cojocaru
Ioachim Drugus
Mykola Nikitchenco
Alexei Muravitsky

Chisinau • 2023

CZU 004.8:510.6(082)
L 74

Copyright © Vladimir Andrunachievici Institute of Mathematics and Computer Science, Moldova State University, 2023.
All rights reserved.

VLADIMIR ANDRUNACHIEVICI INSTITUTE OF MATHEMATICS AND COMPUTER SCIENCE, MOLDOVA STATE UNIVERSITY
5, Academiei str., 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. Svetlana Cojocaru, Prof. Ioachim Drugus,
Prof. Mykola Nikitchenco, Prof. Alexei Muravitsky

Technical Editor: Tatiana Verlan

Authors are fully responsible for the content of their papers.

DESCRIEREA CIP A CAMEREI NAȚIONALE A CĂRȚII DIN REPUBLICA MOLDOVA

Logic & Artificial Intelligence / Vladimir Andrunachievici Institute of Mathematics and Computer Science ; edited by Svetlana Cojocaru [et al.]. – Chişinău : S. n., 2023 (Valinex). – 252 p. : fig., tab.
Referinţe bibliogr. la sfârşitul art. – 70 ex.
ISBN 978-9975-68-484-2.

004.8:510.6(082)

Firma poligrafică „VALINEX" SRL, Chişinău, str. Florilor, 30/1A, 26B, tel./fax 43-03-91,
e-mail: info@valinex.md, http:\\www.valinex.md

Coli editoriale 12,02. Coli de tipar conv. 14,65. Format 60x84 1/16. Garnitură „Times".
Hirtie ofset. Tirajul 70.

Contents

Foreword

The term *Artificial Intelligence*, commonly referred to as AI for short, is usually applied to the field of computer science dealing with tasks that require efforts naturally considered human. Although these artificial intelligence endeavors are carried out with the help of computer algorithms, they also seem to be rational. This allows one to speak of a (special) *logic* (more precisely, several *logics*) that can be formulated to carry out or support such attempts. While classical and intuitionistic logics have their roots in mathematical discourse, AI logic is supposed to govern algorithmic behavior that can be qualified as rational.

A group of researchers from the Vladimir Andrunachievici Institute of Mathematics and Computer Science is the founders of the International Society for Logic and Artificial Intelligence (ISLAI), which is registered as a member of the Division of Logic, Methodology and Philosophy of Science and Technology of the International Union of History and Philosophy of Science and Technology. ISLAI is an organization preoccupied with the promotion of logic and AI as well as the convergence of the two disciplines. Already for the fourth year, the collective organizes seminars and discussions on the interferences of logic and artificial intelligence and participates in the organization of thematic conferences, among which we will mention the Mathematical Foundations of Informatics (MFOI) series.

In January 2022, ISLAI organized The Symposium on Logic and Artificial Intelligence (SLAI) in cooperation with:

- Louisiana Scholars' College at Northwestern State University, Louisiana, USA, (Host),
- Taras Shevchenko National University of Kyiv, Ukraine,
- Alexandru Ioan Cuza University of Iasi, Romania,
- Vladimir Andrunachievici Institute of Mathematics and Computer Science, Chisinau, Moldova,

- Ukrainian Logic Society, Ukraine,
- Academy of Sciences, Moldova.

The main purpose of this symposium was to add synergy to the efforts of researchers working on logic, AI, and their confluence, to organize round-table debates on the state of the art and the new directions in the domain.

This volume contains the contribution papers presented during the symposium, reflecting the broad spectrum of the topics: axiomatic systems for propositional logics, methodological foundations of propositional logics, timed modal logics, the logic of knowledge bases, the logic of natural languages, application of neural networks to solve a variety of problems, especially in medicine. The papers presented at the symposium brought new and valuable contributions to the development of both logic and artificial intelligence and their interaction. All papers were subject to double reviews: one preliminary to acceptance of the papers for the symposium and the second – to their consideration for inclusion in this collection of papers.

Our special thanks go to the keynote speakers: Irina Perfilieva (Ostrava, Czech Republic), Leora Morgenstern (Palo Alto, USA), David Makinson (London, UK), Anatol Reibold (Darmstadt, Hessen, Germany), and Dimiter Vakarelov (Sofia, Bulgaria) for their brilliant invited lectures.

We sincerely thank Professors Adrian Iftene, Inga Țițchiev, Tudor Bumbu, and Tatiana Verlan for their valuable contribution to the organization of the symposium and the editing of this volume.

Svetlana Cojocaru, Ioachim Drugus, Alexei Muravitsky, Mykola Nikitchenko.

More Formalized Axiomatic Systems for Propositional Logic in Isabelle/HOL

Agnes Moesgård Eschen, Asta Halkjær From, Jørgen Villadsen

Abstract

We consider a plethora of well-known axiomatic systems, and some less well-known variants, for classical propositional logic all with soundness and completeness theorems formalized in the proof assistant Isabelle/HOL. We have previously only formalized systems based on either implication and falsity or on disjunction and negation. We have now formalized 12 additional systems based on implication and negation. Each of them has three or more axioms. We have also formalized single-axiom systems, by Łukasiewicz and Tarski as well as by Meredith, and we describe the formalization of the former in detail. A few systems with two axioms are described too.

1 Introduction

Given an axiomatic system, using the sole rule of Modus Ponens for propositional logic, arguably the most important results are the soundness and completeness theorems [1]. Often a lot of tedious work is necessary. Can a modern proof assistant help and provide even more trust in the results? We think so, and furthermore it is then quite easy to experiment with variants of the main axiomatic systems. For more details about this approach, we refer to our recent work [2], where we have formalized systems based either on implication and falsity or disjunction and negation.

Our main new contributions are formalized soundness and completeness theorems for the axiomatic systems for classical propositional logic based on implication and negation as listed on the Wikipedia page (plus a few variants):

©2023 by A. M. Eschen, A. H. From, J. Villadsen

`https://en.wikipedia.org/wiki/List_of_Hilbert_systems`

The entire formalization in the proof assistant Isabelle/HOL [3] is available online together with a short introductory video:

`https://github.com/logic-tools/axiom`

`https://hol.compute.dtu.dk/Systems/`

Open the `Systems.thy` file to have easy access to the formalizations of the 12 axiomatic systems.

The paper is organized as follows. We continue with a discussion of the closest related work (§2). We then define the axiomatic systems (§3). Then we present a particularly interesting single-axiom system by Łukasiewicz and Tarski (§4). We describe the syntax and semantics we use to write and evaluate propositional formulas in Isabelle/HOL (§5). We then perform derivations of new propositions from the single axiom (§6). These results are then used to show the soundness and completeness of the system (§7). We briefly describe another single-axiom system by Meredith that we have also shown completeness for (§8). Finally we conclude and discuss future work (§9).

The Isabelle/HOL formalization discussed in §5, §6, and §7 can be found in the appendix for easy reference.

2 Related Work

The use of automated reasoning tools to investigate axiomatic systems has a long history [4],[5]. The soundness and completeness theorems for axiomatic systems were first formalized by Michaelis and Nipkow [6], also for classical propositional logic, but with a much larger syntax with falsity, negation, conjunction, disjunction and implication. Only a single axiomatic system was proved complete.

We have recently formalized a number of axiomatic systems based either on implication and falsity or disjunction and negation [2], [7]. The present paper is our first take on axiomatic systems based on implication and negation. Our formalizations are stand-alone, and this is a deliberate choice in order to make the approach simpler, which is useful when teaching logic and automated reasoning [8].

Proof systems for first-order logic have also been formalized [9]–[11], but the sole formalized axiomatic system for first-order logic does not include specific axioms for propositional logic [12] (instead it characterizes tautologies semantically).

3 The 12 Axiomatic Systems

We use the same naming and numbering of the axiomatic systems as the Wikipedia page mentioned above, following mainly the work by Imai and Iséki [13].

Frege

$p \to (q \to p)$
$(p \to (q \to r)) \to ((p \to q) \to (p \to r))$
$(p \to q) \to (\neg q \to \neg p)$
$\neg \neg p \to p$
$p \to \neg \neg p$

Hilbert

$p \to (q \to p)$
$(p \to (q \to r)) \to (q \to (p \to r))$
$(q \to r) \to ((p \to q) \to (p \to r))$
$p \to (\neg p \to q)$
$(p \to q) \to ((\neg p \to q) \to q)$

The following Łukasiewicz 1 system (called L1) is very concise, and we have chosen it as the starting point for our formalization.

Łukasiewicz 1

$(p \to q) \to ((q \to r) \to (p \to r))$
$(\neg p \to p) \to p$
$p \to (\neg p \to q)$

Łukasiewicz 2

$((p \to q) \to r) \to (\neg p \to r)$
$((p \to q) \to r) \to (q \to r)$
$(\neg p \to r) \to ((q \to r) \to ((p \to q) \to r))$

Łukasiewicz 3

$p \to (q \to p)$
$(p \to (q \to r)) \to ((p \to q) \to (p \to r))$
$(\neg p \to \neg q) \to (q \to p)$

Łukasiewicz Variant

$(p \to q) \to ((q \to r) \to (p \to r))$
$(\neg p \to q) \to ((q \to p) \to p)$
$p \to (\neg p \to q)$

Mendelson

$p \to (q \to p)$
$(p \to (q \to r)) \to ((p \to q) \to (p \to r))$
$(\neg p \to \neg q) \to ((\neg p \to q) \to p)$

Mendelson Variant

$p \to (q \to p)$
$(p \to (q \to r)) \to ((p \to q) \to (p \to r))$
$(\neg p \to q) \to ((\neg p \to \neg q) \to p)$

Russell

$p \to (q \to p)$
$(p \to q) \to ((q \to r) \to (p \to r))$
$(p \to (q \to r)) \to (q \to (p \to r))$
$\neg \neg p \to p$
$(p \to \neg p) \to \neg p$
$(p \to \neg q) \to (q \to \neg p)$

Sobociński 1

$\neg p \rightarrow (p \rightarrow q)$
$p \rightarrow (q \rightarrow (r \rightarrow p))$
$(\neg p \rightarrow r) \rightarrow ((q \rightarrow r) \rightarrow ((p \rightarrow q) \rightarrow r))$

Sobociński 2

$(p \rightarrow q) \rightarrow (\neg q \rightarrow (p \rightarrow r))$
$p \rightarrow (q \rightarrow (r \rightarrow p))$
$(\neg p \rightarrow q) \rightarrow ((p \rightarrow q) \rightarrow q)$

Sobociński Variant

$\neg q \rightarrow ((p \rightarrow q) \rightarrow (p \rightarrow r))$
$p \rightarrow (q \rightarrow (r \rightarrow (s \rightarrow p)))$
$(\neg p \rightarrow q) \rightarrow ((p \rightarrow q) \rightarrow q)$

The last two axiomatic systems are special in that no axiom has more than 5 occurrences of propositional symbols.

As mentioned above, the Łukasiewicz 1 system is used as the starting point, but there are two exceptions:

- The Sobociński variant system is used as the starting point for the Sobociński 2 system.
- The Łukasiewicz 3 system is used as the starting point for the Frege system.

4 Łukasiewicz and Tarski's Single Axiom

This axiomatic system of classical propositional logic was created by Łukasiewicz and Tarski in 1930, in their work *Untersuchungen über den Assagenkalkül*. During the early 20th century, where many formalizations of classical propositional logic were being developed, some researchers strove for minimality: to formalize a system with only a single axiom, that was both sound and complete. This is such a system with the single axiom:

$$((p \to (q \to p)) \to (((\neg\, r \to (s \to \neg\, t)) \to ((r \to (s \to u)) \to ((t \to s) \to (t \to u)))) \to v)) \to (w \to v)$$

The axiom is long, and it is not straightforward to see that it is itself valid. It is also complicated to use when attempting to perform derivations to form new theorems. It is not easy to see that this axiom alone is sufficient to derive all theorems of classical propositional logic, but in the following sections we describe how Isabelle/HOL can be used to show this.

We point out that due to the particular sequence of theorems in §6, we also have completeness of the following two axioms, which are subformulas of the single axiom:

$$p \to (q \to p)$$

$$(\neg\, r \to (s \to \neg\, t)) \to ((r \to (s \to u)) \to ((t \to s) \to (t \to u)))$$

We consider other systems with single and double axioms in §8.

5 Syntax, Semantics and the L1-System

The file `SLAI.thy` starts off by importing our Isabelle/HOL formalization of the L1-system: `System_L1.thy`.

```
theory SLAI imports System_L1 begin
```

This lets us use the datatypes, functions abbreviations, etc. that we have specified there. We use $\sim$ as the symbol for logical negation, instead of $\neg$, in our Isabelle/HOL formalization. Propositional formulas are here formed from propositional symbols, logical negation $\sim$ and logical implication $\to$:

```
datatype form = Pro nat (‹·›) |
                Neg form (‹∼›) |
                Imp form form (infix ‹→› 0)
```

We define a propositional symbol as a natural number with a preceding `Pro` or · symbol. A negated formula is specified with a preceding `Neg` or $\sim$ symbol, and a logical implication can be written either with a preceding `Imp` or in infix notation using the → symbol. The truth value of a formula is specified by recursively dividing the formula into its parts and evaluating them under a given interpretation:

```
primrec semantics (infix ‹⊨› 0) where
  ‹(I ⊨ · n) = I n› |
  ‹(I ⊨ ∼ p) = (if I ⊨ p then False else True)› |
  ‹(I ⊨ (p → q)) = (if I ⊨ p then (I ⊨ q) else True)›
```

We can use the ⊨ symbol in infix notation to evaluate a formula under a given interpretation. Having defined the semantics, we specify a formula as valid when it is true in all interpretations (sometimes we choose not to write `valid p` and write `∀I. (I ⊨ p)` in full):

```
definition ‹valid p ≡ ∀I. (I ⊨ p)›
```

Importing `System_L1.thy` also lets us use the theorems we have shown about the L1-system, including its soundness and completeness. We symbolize the axiomatics of the L1-system in Isabelle/HOL with ⊢. For a formula p we specify that p is provable in the L1-system as $\vdash p$. Soundness is then shown by induction using the automatic theorem prover `auto`:

```
theorem L1_soundness: ‹⊢ p ⟹ I ⊨ p›
```

The proof of completeness is much more involved and cannot be solved directly by any automatic theorem prover. After about 600 lines of establishing lemmas we get to show the theorem:

```
theorem L1_completeness: ‹∀I. (I ⊨ p) ⟹ ⊢ p›
```

The proof strategy for showing completeness of this single-axiom system is to use the established completeness of the L1-system. If we can derive all of the L1-system's axioms from the single axiom and Modus Ponens, then the completeness of the L1-system directly implies the completeness of the single-axiom system. This is what we do in `SLAI.thy`.

6 From Axiom to Theorems

Shôtarô Tanaka has in 1965 shown derivations of the L3-system's axioms from this single-axiom system in *On Axiom Systems of Propositional Calculi. XIII.* His proof consists of 19 derivations and is the inspiration for the derivations of the L1-system's axioms in `SLAI.thy`. The number n in the name of a lemma in `SLAI.thy` corresponds to the n'th derivation in Shôtarô Tanaka's proof. Lemmas `LT_3`, `LT_5`, and `LT_19` are the three axioms of the L3-system. Lemmas `LT_20` and `LT_21` are extensions to the proof: along with `LT_11` they constitute the three axioms of the L1-system. We specify the Łukasiewicz-Tarski (abbreviated `LT`) single-axiom system in Isabelle as an inductive definition with the symbol ⊩. Modus Ponens is specified as the single inference rule and labelled `LT_MP`, and the axiom is specified and labelled `LT_1`:

```
inductive LT (‹⊩›) where
  LT_MP: ‹⊩ q› if ‹⊩ p› and ‹⊩ (p → q)› |
  LT_1:  ‹⊩ (((p → (q → p))
              → (((∼ r → (s → ∼ t))
                → ((r → (s → u))
                  → ((t → s) → (t → u)))) → v))
             → (w → v))›
```

The first derivation we perform is the third derivation in Shôtarô Tanaka's proof:

```
lemma LT_3: ‹⊩ (p → (q → p))›
  using LT_1 LT_MP by metis
```

Lemma `LT_3` can be identified as the premise of the premise of `LT_1`. It is derivable using only `LT_1` and `LT_MP`. The proof is done by the automatic theorem prover `metis`.

```
lemma LT_4: ‹⊩ ((∼ r → (s → ∼ t))
          → ((r → (s → u)) → ((t → s) → (t → u))))›
  using LT_1 LT_3 LT_MP by metis
```

Lemma `LT_4` is the conclusion of the premise of `LT_1` and it can be derived from `LT_1` and `LT_3` by `metis`, using Modus Ponens (`LT_MP`) as the proof method.

```
lemma LT_5: ‹⊩ ((p → (q → r))
                 → ((p → q) → (p → r)))›
  using LT_3 LT_4 LT_MP by metis
```

Lemma LT_5 is the conclusion of LT_4 under the substitution {r ↦ p, s ↦ q, u ↦ r, t ↦ p}. Under this same substitution, the premise of LT_4 becomes (∼ p → (q → ∼ p)) which under the substitution {p ↦ ∼ p} is LT_3, and so from a single application of Modus Ponens with LT_3 and LT_4 as the premises, metis can derive LT_5.

```
lemma LT_11: ‹⊩ ((p → q) → ((q → r) → (p → r)))›
  using LT_3 LT_5 LT_MP by metis
```

Lemma LT_11 is derivable with Modus Ponens from LT_3 and LT_5 by metis. In Shôtarô Tanaka's proof, derivations 6-10 are setup for derivations 11 and 12. It turns out that these setup lemmas are not strictly needed, which we found out by utilizing Isabelle's Sledgehammer tool. Sledgehammer deploys a suite of automated reasoning software, along with the user's already established lemmas, to perform a proof search. From this search it turned out that LT_3 and LT_5 are sufficient to show LT_11.

```
lemma LT_13: ‹⊩ (((p → q) → (p → r))
                  → (q → (p → r)))›
  using LT_3 LT_5 LT_11 LT_MP by metis
```

It also turned out that Shôtarô Tanaka's derivation 12 is not strictly needed either. Lemma LT_13, which is originally proved by Shôtarô Tanaka from his derivations 5, 8, and 12, can be shown by metis from LT_3, LT_5, and LT_11 instead, resulting in an overall shorter proof.

```
lemma LT_19: ‹⊩ ((∼ p → ∼ q) → (q → p))›
  using LT_3 LT_4 LT_5 LT_11 LT_13 LT_MP by metis
```

Providing metis with all the derivations shown up until now turns out to be enough for it to show lemma LT_19. Shôtarô Tanaka's original derivations 14-18 can thus be omitted and the proofs of L3's three axioms are now completed. It turns out that the above lemmas are sufficient to show the two last axioms of the L1-system. From LT_3, LT_5, and LT_19, metis can derive the first:

```
lemma LT_20: ‹⊩ ((∼ p → p) → p)›
  using LT_3 LT_5 LT_19 LT_MP by metis
```

From LT_3, LT_5, LT_11, LT_13, and LT_19, metis can derive the last:

```
lemma LT_21: ‹⊩ (p → (∼ p → q))›
  using LT_3 LT_5 LT_11 LT_13 LT_19 LT_MP by metis
```

Together with LT_11 these are the three axioms of the L1-system.

7 Soundness and Completeness

Having derived the axioms of the L1-system, we can use the completeness of L1 to show the completeness of LT. We, however, show a stronger result: that LT and L1 are equivalent in deductive capability. We express this in Isabelle/HOL with a meta-logic bi-implication:

```
theorem LT_iff_L1: ‹⊩ p ⟷ ⊢ p›
```

To show this, we write proof, which queries Isabelle/HOL to suggest a proof method. Isabelle/HOL, in this case, responds by declaring two sub-goals: 1. ⊩ p ⟹ ⊢ p and 2. ⊢ p ⟹ ⊩ p. In other words, if p is provable in ⊩, then p is also provable in ⊢, and vice versa. If we can show this, then Isabelle/HOL reasons that our original theorem is proven.

```
proof
  have L1_LT_1:
    ‹⊢ (((p → (q → p))
         → (((∼ r → (s → ∼ t)) → ((r → (s → u))
           → ((t → s) → (t → u)))) → v))
        → (w → v))› for p q r s t u v w
    using L1_completeness by simp
```

We show the first sub-goal by utilizing the established completeness of the L1-system. By completeness, all valid propositions are provable in ⊢, and since the single axiom of ⊩ is valid, Isabelle/HOL's simplifier can immediately show that it is provable in ⊢ when provided with the L1_completeness fact. Since all propositions that are provable in ⊩

have been proven using this single axiom and Modus Ponens only, we can, from the derivation of the single-axiom in ⊢, mimic the ⊩ proof and show that if ⊩ p, then ⊢ p.

```
  show ‹⊢ p› if ‹⊩ p›
    using that by (induct) (metis MP, metis L1_LT_1)
```

To show the second sub-goal, we utilize our derivations of the three axioms of the L1-system. Using the same argument, if a proposition is provable in ⊢, then we can, from the derivation in ⊩ of L1's three axioms, repeat the same proof in ⊩.

```
  show ‹⊩ p› if ‹⊢ p›
    using that by (induct) (metis LT_MP, metis LT_11,
                                         metis LT_20,
                                         metis LT_21)
qed
```

Having shown ⊩ and ⊢ to be deductively equivalent, which we label `LT_iff_L1`, Isabelle/HOL can infer the completeness of ⊩ from the completeness of ⊢.

⊩'s soundness can be shown by structural induction using the `auto` automatic theorem prover:

```
theorem LT_soundness: ‹⊩ p ⟹ I ⊨ p›
  by (induct rule: LT.induct) auto

theorem LT_completeness: ‹∀I. (I ⊨ p) ⟹ ⊩ p›
  using LT_iff_L1 by (simp add: L1_completeness)
```

Finally, by combining soundness and completeness, we can show the main theorem of `SLAI.thy`:

```
section ‹Soundness and Completeness›

theorem main: ‹valid p = ⊩ p›
  unfolding valid_def using LT_soundness LT_completeness
  by blast

lemmas LT = LT.intros main
```

The last command defines LT as the axiomatic system together with the main theorem.

8 Meredith's Single Axiom

Around 20 years after Łukasiewicz and Tarski created their single-axiom system, Carew Arthur Meredith developed another single-axiom system in his 1953 work *Single Axioms for the Systems (C, N), (C, 0) and (A, N) of the Two-Valued Propositional Calculus* (a paper which has turned out to be impossible for us to obtain):

$$(((((\mathrm{p} \to \mathrm{q}) \to (\neg\, \mathrm{r} \to \neg\, \mathrm{s})) \to \mathrm{r}) \to \mathrm{t}) \to ((\mathrm{t} \to \mathrm{p}) \to (\mathrm{s} \to \mathrm{p})))$$

This single axiom is significantly shorter and also manages to deploy fewer unique propositional symbols. In the same work, Meredith showed the completeness of this system by deriving the three axioms of the L1-system. We have done the same and again Isabelle/HOL has proved to be an invaluable asset in establishing meta-theoretical results of propositional axiomatic systems. The proof uses both `metis` and `meson` in Isabelle/HOL (file `SLAI_1.thy`). Just seven specific theorems are needed (compared to 5 theorems, namely `LT_3`, `LT_4`, `LT_5`, `LT_13`, and `LT_19` for the file `SLAI.thy`, since `LT_11`, `LT_20`, and `LT_21` are the axioms for the L1 system).

We have also shown the soundness and completeness of the following axioms (the first axiom is from the Meredith proof and the second axiom is from the Łukasiewicz and Tarski proof):

$$((\mathrm{p} \to \mathrm{q}) \to \mathrm{r}) \to (\mathrm{q} \to \mathrm{r})$$

$$(\neg\, \mathrm{p} \to (\mathrm{q} \to \neg\, \mathrm{r})) \to ((\mathrm{p} \to (\mathrm{q} \to \mathrm{s})) \to ((\mathrm{r} \to \mathrm{q}) \to (\mathrm{r} \to \mathrm{s})))$$

This is essentially automatic using `metis` and `meson` in Isabelle/HOL (file `SLAI_2.thy`).

9 Conclusions and Future Work

We have described our approach to formalizing axiomatic systems for classical propositional logic, extending our previous work [2], and with focus on single-axiom systems. Our novel Isabelle/HOL formalization consists of 16 files with 1794 lines and is available on GitHub with the link in §1. Isabelle2021-1 checks all files in just a few seconds.

For future work, we plan to consider other classical logics and also non-classical logics, in particular intuitionistic logic. We find that our approach is useful when teaching logic and automated reasoning [8], and we would like to develop proper teaching material.

We note that the axiomatic system in §8 with two axioms has been shown sound and complete essentially automatically, just by entering the two axioms in the proposed template, and it would be interesting if this can be done for other, new or old, axiomatic systems.

Acknowledgements We thank Frederik Krogsdal Jacobsen for help.

References

[1] A. Church, *Introduction to Mathematical Logic.* Princeton University Press, 1956.

[2] A. H. From, A. M. Eschen, and J. Villadsen, "Formalizing axiomatic systems for propositional logic in Isabelle/HOL," in *Intelligent Computer Mathematics - 14th International Conference, CICM 2021, Timisoara, Romania, July 26-31, 2021, Proceedings*, ser. Lecture Notes in Artificial Intelligence, F. Kamareddine and C. Sacerdoti Coen, Eds., vol. 12833. Springer, 2021, pp. 32–46.

[3] T. Nipkow, L. C. Paulson, and M. Wenzel, *Isabelle/HOL - A Proof Assistant for Higher-Order Logic*, ser. Lecture Notes in Computer Science. Springer, 2002, vol. 2283.

[4] F. Pfenning, "Single axioms in the implicational propositional calculus," in *9th International Conference on Automated Deduction, Argonne, Illinois, USA, May 23-26, 1988, Proceedings*, ser. Lecture Notes in Computer Science, E. L. Lusk and R. A. Overbeek, Eds., vol. 310. Springer, 1988, pp. 710–713.

[5] B. Fitelson and L. Wos, "Finding missing proofs with automated reasoning," *Studia Logica*, vol. 68, no. 3, pp. 329–356, 2001.

[6] J. Michaelis and T. Nipkow, “Formalized proof systems for propositional logic,” in *23rd International Conference on Types for Proofs and Programs, TYPES 2017, May 29-June 1, 2017, Budapest, Hungary*, ser. LIPIcs, A. Abel, F. N. Forsberg, and A. Kaposi, Eds., vol. 104. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2017, pp. 5:1–5:16.

[7] A. H. From, “Formalizing Henkin-style completeness of an axiomatic system for propositional logic,” in *Proceedings of the Web Summer School in Logic, Language and Information (WeSSLLII) and the European Summer School in Logic, Language and Information (ESSLLI) Virtual Student Session*, 2020, accepted for Springer post-proceedings.

[8] J. Villadsen and F. K. Jacobsen, “Using Isabelle in two courses on logic and automated reasoning,” in *Formal Methods Teaching*, J. F. Ferreira, A. Mendes, and C. Menghi, Eds. Springer, 2021, pp. 117–132.

[9] T. Ridge and J. Margetson, “A mechanically verified, sound and complete theorem prover for first order logic,” in *Theorem Proving in Higher Order Logics, 18th International Conference, TPHOLs, Oxford, UK, August 22-25, 2005, Proceedings*, 2005, pp. 294–309.

[10] J. C. Blanchette, “Formalizing the metatheory of logical calculi and automatic provers in Isabelle/HOL (invited talk),” in *Proceedings of the 8th ACM SIGPLAN International Conference on Certified Programs and Proofs, CPP 2019, Cascais, Portugal, January 14-15, 2019*, A. Mahboubi and M. O. Myreen, Eds. ACM, 2019, pp. 1–13.

[11] A. H. From, A. Schlichtkrull, and J. Villadsen, “A sequent calculus for first-order logic formalized in Isabelle/HOL,” in *Proceedings of the 36th Italian Conference on Computational Logic - CILC 2021, Parma, Italy, September 7-9, 2021*, ser. CEUR Workshop Proceedings, S. Monica and F. Bergenti, Eds., vol. 3002, 2021, pp. 107–121. [Online]. Available: http://ceur-ws.org/Vol-3002/paper7.pdf

[12] A. H. From, "Soundness and completeness of an axiomatic system for first-order logic," *Archive of Formal Proofs*, September 2021, https://www.isa-afp.org/entries/FOL_Axiomatic.html.

[13] Y. Imai and K. Iséki, "On axiom systems of propositional calculi, I," *Proceedings of the Japan Academy*, vol. 41, no. 6, pp. 436–439, 1965.

Appendix: The SLAI.thy File

```
theory SLAI imports System_L1 begin

text ‹ System from Jan Lukasiewicz and Alfred Tarski (1930): Untersuchungen über den Aussagenkalkül ›

text ‹ Inspired by Shotaro Tanaka (1965): On Axiom Systems of Propositional Calculi. XIII ›

inductive LT (‹⊩ ›) where
  LT_MP: ‹⊩ q› if ‹⊩ p› and ‹⊩ (p → q)› |
  LT_1:  ‹⊩ (((p → (q → p))
             → (((∼ r → (s → ∼ t)) → ((r → (s → u)) → ((t → s) → (t → u)))) → v))
            → (w → v))›

lemma LT_3: ‹⊩ (p → (q → p))›
  using LT_1 LT_MP by metis

lemma LT_4: ‹⊩ ((∼ r → (s → ∼ t)) → ((r → (s → u)) → ((t → s) → (t → u))))›
  using LT_1 LT_3 LT_MP by metis

lemma LT_5: ‹⊩ ((p → (q → r)) → ((p → q) → (p → r)))›
  using LT_3 LT_4 LT_MP by metis

lemma LT_11: ‹⊩ ((p → q) → ((q → r) → (p → r)))›
  using LT_3 LT_5 LT_MP by metis

lemma LT_13: ‹⊩ (((p → q) → (p → r)) → (q → (p → r)))›
  using LT_3 LT_5 LT_11 LT_MP by metis

lemma LT_19: ‹⊩ ((∼ p → ∼ q) → (q → p))›
  using LT_3 LT_4 LT_5 LT_11 LT_13 LT_MP by metis

lemma LT_20: ‹⊩ ((∼ p → p) → p)›
  using LT_3 LT_5 LT_19 LT_MP by metis

lemma LT_21: ‹⊩ (p → (∼ p → q))›
  using LT_3 LT_5 LT_11 LT_13 LT_19 LT_MP by metis

theorem LT_iff_L1: ‹⊩ p ⟷ ⊢ p›
proof
  have L1_LT_1:
    ‹⊢ (((p → (q → p))
         → (((∼ r → (s → ∼ t)) → ((r → (s → u)) → ((t → s) → (t → u)))) → v))
        → (w → v))› for p q r s t u v w
    using L1_completeness by simp
  show ‹⊢ p› if ‹⊩ p›
    using that by (induct) (metis MP, metis L1_LT_1)
  show ‹⊩ p› if ‹⊢ p›
    using that by (induct) (metis LT_MP, metis LT_11, metis LT_20, metis LT_21)
qed

theorem LT_soundness: ‹⊩ p ⟹ I ⊨ p›
  by (induct rule: LT.induct) auto
```

```
theorem LT_completeness: ‹∀I. (I ⊨ p) ⟹ ⊩ p›
  using LT_iff_L1 by (simp add: L1_completeness)

section ‹Soundness and Completeness›

theorem main: ‹valid p = ⊩ p›
  unfolding valid_def using LT_soundness LT_completeness by blast

lemmas LT = LT.intros main

end
```

Agnes Moesgård Eschen, Asta Halkjær From
Jørgen Villadsen

Received June 06, 2022
Accepted August 22, 2022

Technical University of Denmark
E–mail: jovi@dtu.dk

Use of the Model "State-Probability of Choice" for Modelling Decision Making within Uncertain Reasoning

Oleksiy Oletsky

Abstract

An approach to uncertain reasoning, in particular for dealing with contradictory information, which is based on the model "state-probability of choice" has been suggested. Within this approach, uncertain reasoning is performed by calculating products of rectangular stochastic matrices associated with the facts and rules of inference, which are related to the nodes and the edges of the inference graph. Some examples which illustrate applying a single rule of inference and a conflict between two rules are presented. Situations of dynamic equilibrium of alternatives are showcased as well.

Keywords: model "state-probability of choice", uncertain reasoning, rectangular stochastic matrices, inconsistent reasoning, dynamic equilibrium.

1 Introduction

Decision making is often based on reasoning, but it may be not reliable enough. If we regard the knowledge and beliefs of a human or an artificially intelligent agent as a logical theory, this theory may be inconsistent and incomplete. For example, if A, B, and L are the statements, it is possible that A implies L, B implies $\overline{L}$, and an agent believes both in A and in B.

Such situations may be studied within paraconsistent logical systems (for instance, [1, 2]). But there is another approach related to probabilistic inference. If an agent can infer both A and not A, and it

©2023 by Oleksiy Oletsky

comes to decision making, we may consider a probability that they will choose A (or not A). A well-known approach to considering such issues is related to Bayesian networks (for instance, [3]). But the fact is that such a probabilistic inference doesn't need to be the Bayesian one only. Moreover, though we can say about probabilities of choices as they are, the process of reasoning may be not Bayesian. Moreover, it may be not of probabilistic nature at all.

It appears important to develop models describing the process of decision making and behavioral aspects of that on the base of reasoning. Another important issue arises if we take into consideration possible influences that agents of influence can take up in order to affect the decisions of other agents.

We are developing an approach based on the model "state-probability of choice" reported in [4] and developed in [5]. We are going to make a short summary of this model and then outline how this model can be generalized and applied for modeling the process of decision making based on reasoning.

2 Model "state-probability of choice" and dynamic equilibrium of alternatives

The model "state-probability of choice" postulates that there are different states, and each of them represents a possible distribution of probabilities among alternatives under consideration.

In more detail, let there be n alternatives for choice and m be a number of possible distributions of probabilities interpreted as states. In [4], there is a matrix $Z = \left(z_{ij}, i = \overline{1,m}, j = \overline{1,n}\right)$, where z_{ij} is a probability that an agent which is being in the i-th state will choose the j-th alternative. So, each state corresponds to a specific distribution of probabilities and is represented by the corresponding row of the matrix. Possible distributions of probabilities should be specified explicitly, but these distributions can be taken rather arbitrarily. Below we are going to discuss how such an arbitrariness can be mitigated.

The sum of each row of Z equals to 1. So, by the analogy with usual stochastic matrices, it was suggested to call such matrices rectangular stochastic matrices.

A rectangular stochastic matrix is said to be balanced if the sums of all its columns are equal to each other. It was shown that these sums are equal to $\frac{m}{n}$.

A vector $p = (p_1, \ldots, p_m)$, where p_i is the probability that an agent is being in the i-th, state should be regarded. Then the overall probability that an agent will choose the i-th alternative equals [4] to:

$$v_j = \sum\nolimits_{i=1}^{m} p_i z_{ij}, j = 1, .., n, \tag{1}$$

or in a matrix notation

$$v = pZ. \tag{2}$$

A vector p can be either postulated explicitly or obtained from a Markov chain with the given matrix of transition probabilities for it. This was described in more detail in [4].

An important problem is the problem of dynamic equilibrium of alternatives, which means that no alternative has advantages over the others. It can be formulated as follows [4, 5]: given Z, find a vector p (or transition probabilities with p as a stationary distribution) that is a solution of the system of equations [4]

$$\sum\nolimits_{i=1}^{m} p_i z_{ij} = \frac{1}{n} \forall j \tag{3}$$

with the constraints

$$\sum\nolimits_{i=1}^{m} p_i = 1, 0 \leq p_i \leq 1. \tag{4}$$

The problem of dynamic equilibrium has been explored in [4, 5]. It is of the most importance if $n=2$ and decisions are made collectively by a majority of votes [4]. As it was showcased in [4], if the number of agents is large enough, a situation of the dynamic equilibrium is practically the only situation when alternatives are rotated, change each other, and are chosen by turn. In addition to this, if an agent of influence wants to promote an alternative that is losing at the moment, they may need first of all to reach the nearest point of the dynamic equilibrium and then to move it away from this point in the desired direction [5].

3 Graphs of inference and probabilities of choice

When constructing procedures for reasoning and logical inference, graphs of inference like AND-OR-graphs are typically applied. The nodes of such a graph correspond to statements, and the edges of it are related to logical inference.

In this paper, we are suggesting a way of implementing the model "state-probability of choice" in reasoning models based on inference graphs. Each element Q (node or edge) should be associated with one or more systems of states $S_1^Q, S_2^Q, \ldots$ and corresponding rectangular stochastic matrices $M_1^Q, M_2^Q, \ldots$. Now we are going to illustrate this approach in some simple cases.

4 A chain of uncertain inference

Let's consider a typical chain of implications

$$A_0 \Rightarrow A_1 \Rightarrow \ldots A_n = L. \qquad (5)$$

The statement $A_n = L$ is obviously related to the goal of reasoning. It is a final decision that has to be accepted or rejected. As it was stated below, we are interested in a certain probability of such a decision. The initial statement A_0, which is a starting point of the given chain of reasoning, is typically considered as a fact known or believed with some degree of certainty. And a fragment $A^i \Rightarrow A^{i+1}$, which is related to the corresponding edge of the inference graph, represents the rule of inference.

Within the framework of the model "state-probability of choice", for the terminal node L, we are regarding two alternatives: either to accept a decision or to reject it. So, a number of alternatives n=2. We are interested in probabilities $v_j, j = \overline{1,2}$ for each of these alternatives and in building a matrix "state-probability of choice" associated with L. Technically speaking, these probabilities are merely some values in the range from *0* to *1*. Similarly, as a matrix "state-probability of choice", we can take any matrix Z constructed with the help of the algorithm suggested in [4] regardless of the nature of the required decision. Each row of this matrix is a pair of some values that can be taken rather arbitrarily. In order to reduce this arbitrariness, we suggest some sort of grouping of these states by means of introducing another matrix “state-probability of choice” denoted by G.

The matrix $G = \left(g_{ij}, i = \overline{1,m}, j = \overline{1,m_G}\right)$ is constructed as follows. Each row of it corresponds to a state, which means that an agent is being in some group of states represented by Z. More formally, let $\xi^G$and ξ^Z be the random variables meaning which states of S^Gor S^Zrespectively an agent is being in. Then

$$g_{ij} = P(\xi^Z = z_j | \xi^G = g_i). \tag{6}$$

The very important fact is that the matrix product GZ is a new matrix “state-probability of choice”.

For rules like $A^k \Rightarrow A^{k+1}$, we are building systems of states and corresponding matrices in a similar way. Namely,

$$M^{A^k \Rightarrow A^{k+1}} = (m_{ij}^{A^k \Rightarrow A^{k+1}}), \tag{7}$$

where

$$m_{ij}^{A^k \Rightarrow A^{k+1}} = P(\xi^{k+1} = s_j^{k+1} | \xi^k = s_i^k). \tag{8}$$

For input statement A^o, a vector p^0 which determines stationary probabilities of being in each state, can be postulated. As it was mentioned above, this vector can be either postulated explicitly or obtained from a Markov chain with given transition probabilities between states.

Then it can be easily seen that

$$v = p^0 \cdot M^{A^0 \Rightarrow A^1} \cdot \ldots \cdot M^{A^{n-2} \Rightarrow A^{n-1}} \cdot M^{A^{n-1} \Rightarrow A^n} \cdot G \cdot Z. \tag{9}$$

The study carried out in [4] makes it possible to postulate input probabilities and matrices so that dynamic equilibrium shall hold. As it ensues from the results of [4], dynamic equilibrium shall hold provided that the vector p is symmetric and the matrix product

$$M^{A^0 \Rightarrow A^1} \cdot \ldots \cdot M^{A^{n-2} \Rightarrow A^{n-1}} \cdot M^{A^{n-1} \Rightarrow A^n} \cdot G \cdot Z \tag{10}$$

is centrosymmetric.

Let’s illustrate the approach presented above in the following example.

Example 1. *Let's consider the simple single rule*

$$A^0 \Rightarrow L. \tag{11}$$

Let the matrix Z be as follows:

$$Z = \begin{pmatrix} 1 & 0 \\ 0.8 & 0.2 \\ 0.6 & 0.4 \\ 0.5 & 0.5 \\ 0.4 & 0.6 \\ 0.2 & 0.8 \\ 0 & 1 \end{pmatrix}.$$

For the grouping matrix G, we may introduce three groups of states with the following more or less clear interpretation:

- convinced proponents of accepting L;
- those who hesitate;
- convinced proponents of rejecting L.

Let's take

$$G = \begin{pmatrix} 0.2 & 0.6 & 0.2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.2 & 0.6 & 0.2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.2 & 0.6 & 0.2 \end{pmatrix}.$$

Finally, let's take

$$M^{A^0 \Rightarrow L} = \begin{pmatrix} 0.9 & 0.1 & 0 \\ 0.15 & 0.7 & 0.15 \\ 0 & 0.1 & 0.9 \end{pmatrix}$$

and

$$p^0 = (0.2, 0.6, 0.2).$$

Then

$$v = p^o \cdot M^{A^0 \Rightarrow L} \cdot G \cdot Z = (0.5, 0.5).$$

So, dynamic equilibrium holds.

5 Contradictory pieces of evidence

Let's consider two pieces of evidence

$$\begin{aligned} A &\Rightarrow L, \\ B &\Rightarrow L. \end{aligned} \tag{12}$$

We can repeat the procedure described above for both rules separately, but results may be very different. In particular, an agent might infer both L and $\overline{L}$.

Let's consider an example.

Example 2.

Let's take Z and G the same as in the Example 1, but now there are two different matrices related to the inference rules (12)

$$M^{A \Rightarrow L} = \begin{pmatrix} 0.9 & 0.1 & 0 \\ 0.15 & 0.7 & 0.15 \\ 0 & 0.1 & 0.9 \end{pmatrix},$$

$$M^{B \Rightarrow L} = \begin{pmatrix} 0. & 0.1 & 0.9 \\ 0.15 & 0.7 & 0.15 \\ 0.9 & 0.1 & 0. \end{pmatrix}$$

and

$$p^A = p^B = (0.35, 0.4, 0.25).$$

Then performing inference by both rules leads to contradictory results.

Really,

$$v^A = p^A \cdot M^{A \Rightarrow L} \cdot G \cdot Z = (0.527, 0.473),$$

so L wins.

But

$$v^B = p^B \cdot M^{B \Rightarrow L} \cdot G \cdot Z = (0.473, 0.527),$$

so L loses.

For combining the evidence, we might take a convex combination of them. For instance, we can consider a combined matrix

$$\begin{aligned} C^{\alpha} &= \alpha A + (1-\alpha)B, \\ &\leq \alpha \leq 1. \end{aligned} \tag{13}$$

Taking $\alpha = 0.5$ gives the vector

$$v^{\alpha=0.5} = (0.5, 0.5).$$

Dynamic equilibrium holds again, but now that is another sort of equilibrium. It depends on the parameter α, and agents of influence might try to break this situation by affecting this parameter.

For instance, taking $\alpha = 0.6$ gives the vector

$$v^{\alpha=0.6} = (0.5054, 0.4946)$$

and dynamic equilibrium has been broken. The option of accepting A wins, and it shall win permanently if decisions are made collectively by a majority of votes.

6 Conclusions and discussion

The model "state-probability of choice" suggested in [4, 5] has been enhanced and generalized, so that this model could be applied for uncertain reasoning, in particular for reasoning on the basis of contradictory facts and rules of inference. Summarizing, the approach suggested in the paper involves some systems of states and corresponding stochastic rectangular matrices for given facts, rules of inference, and terminal decisions. Within this approach, probabilities of a certain decision can be calculated by calculating matrix products described in the paper. Under some conditions, a situation of dynamic equilibrium, which is a situation when none of the alternatives has advantages over the others, can be ensured. The approach has been illustrated in the following simple examples:

- a single rule of inference;
- two contradictory rules of inference.

Speaking more generally, a network like AND-OR-graph can be constructed, and nodes and edges of such a graph are associated with some states and matrices. It is important to mention that even though it is still a tool for uncertain inference with some probabilistic features, links between nodes and the overall process of reasoning may be not Bayesian, which is commonly used now. Moreover, the process of uncertain reasoning may not be strictly probabilistic.

On the other hand, the suggested approach appears to be useful not just for uncertain reasoning only, but it can be regarded as a tool for modelling agents' behavior. Another important aspect is that it can be regarded as a tool for modelling outer influences which can be taken up by other influence. In order to promote some alternatives, these agents can try affecting some parameters considered within the model, such as matrices associated with facts and rules, weighting coefficients of contradicting facts and rules, transition matrices between states in a Markov chain, etc.

It is important to develop techniques for resolving conflicts as well.

Within this approach, distributions of probabilities may not be postulated explicitly, but they can be obtained from distributions of importance among alternatives [5] or with the help of pairwise comparisons [4, 5]. It appears promising to enhance the approach itself by applying techniques involving different types of fuzzy numbers [6] and those of combining fuzzy and probabilistic approaches [7].

A game approach can be applied as well. Really, if an influencer tries to promote a specific decision and thereby to increase a probability of its acceptance, the other agents of influence can try to counteract these efforts. A question of choosing a node, where a specific influencer should make the efforts to, may be of great interest as well. If we take into consideration a level of a supervisory board, we might consider models of such a supervising by setting rules of the games within the algorithmic game theory and the theory of mechanism design [8, 9].

References

[1] *Handbook of Paraconsistency*. J.-Y. Béziau, W. Carnielli, and D. Gabbay, Eds. London: King's College, 2007.

[2] W. Carnielli, M. Coniglio, and J. Marcos, “Logics of Formal Inconsistency,” in *Handbook of Philosophical Logic*, Volume 14, 2nd ed., D. Gabbay and F. Guenthner, Eds. The Netherlands: Kluwer Academic Publishers, 2007, pp. 1–93.

[3] A. Darwiche, *Modeling and Reasoning with Bayesian Networks*, Cambridge University Press, 2009.

[4] O.V. Oletsky and E.V. Ivohin, “Formalizing the Procedure for the Formation of a Dynamic Equilibrium of Alternatives in a Multi-Agent Environment in Decision-Making by Majority of Votes,” *Cybern Syst Anal*, vol.57, no. 1, pp.47–56, 2021. DOI: https://doi.org/10.1007/s10559-021-00328-y.

[5] O. Oletsky, “Exploring Dynamic Equilibrium Of Alternatives On The Base Of Rectangular Stochastic Matrices,” in *Modern Machine Learning Technologies and Data Science Workshop. Proc. 3rd International Workshop (MoMLeT&DS 2021),* Volume I: Main Conference, Lviv-Shatsk, Ukraine, June 5-6, 2021. CEUR Workshop Proceedings 2917. - CEUR-WS.org 2021, 2021, pp. 151–160. http://ceur-ws.org/Vol-2917/.

[6] E.V. Ivokhin and D.V. Apanasenko, “Clustering of Composite Fuzzy Numbers Aggregate Based on Sets of Scalar and Vector Levels,” *Journal of Automation and Information Sciences*, vol. 50, no. 10, pp. 47–59, 2018. DOI: 10.1615/JAutomatInfScien.v50. i10.40.

[7] O. Provotar, “Fuzzy Probabilities of Fuzzy Events,” *Cybern Syst Anal*, vol. 56, pp. 171–180, 2020. DOI: 10.1007/s10559-020-00232-x.

[8] T. Borgers, D. Krahmer, and R. Strausz, *An introduction to the theory of mechanism design*, Oxford University Press, 2015.

[9] T. Roughgarden, *Twenty lectures on algorithmic game theory*, Cambridge University Press, 2016.

Oleksiy Oletsky

Received September 05, 2022
Accepted October 19, 2022

National Academy of Kyiv-Mohyla University, Faculty of Informatics,
Kyiv, Ukraine
E–mail: `oletsky@ukr.net`

Privacy-Aware Self-Supervised Deep Learning Approaches for Medical Data

Cristian Simionescu

Abstract

Training models for medical image analysis tasks requires large amount of labeled data. It is often difficult to procure this data due to it is required that experts manually label it. Due to this, creating self-supervision learning methods and domain adaptation techniques dedicated to this domain is essential. In this Ph.D. research plan, we propose the development of unsupervised/self-supervised learning methods for medical data. We plan to develop self-supervised region of interest models using deep multitask learning and deep domain adaptation. While pursuing these directions, we want to analyze our and others' results from a deep learning data privacy perspective.

Keywords: deep learning, medical image analysis, self-supervised learning.

1 Introduction

Deep learning has been successfully applied to medical image analysis to help medical professionals over the past years. Machine learning is being used for offering diagnosis suggestions, identifying regions of interest in images or augmenting data to remove noise and improve results. Training models for such tasks requires large amount of labeled data. It is often difficult to procure this data due to it is required that experts manually label it. This raises problems such as requiring training of medics to annotate datasets. In the context of a prolonged shortage of medical professionals around the globe coupled with an expending life expectancy and population, retraining large segments of this workforce to digitalize and annotate the immense amounts of data

©2023 by Cristian Simionescu

required by modern deep learning algorithms is undesirable. While posing a prohibitive cost associated with such an endeavor for both academic research and product development, pulling medics from other life-saving work would be unfavorable. Another issue arises when using data provided by different sources, and our algorithms require us to harmonize data that is labeled with unlabeled data and with data that may be using different, unreliable standards for annotation. All of these considerations lead to the fact that annotated data isn't always available or found in desirable volumes for many deep learning training.

2 Motivation

2.1 A case for self-supervision

Having weighted in the specific characteristics of working with medical data, we propose to research self-supervised deep learning methods developed specifically for medical tasks. Following the success of language models on natural language processing tasks, self-supervision has been validated as an incredible approach for scenarios with unlabeled data. These algorithms can be leveraged in various ways, such as anomaly detection in images or region of interest segmentation and few-shot learning for classification. Having the possibility of creating labels automatically from input data or training on large unlabeled datasets and then utilizing a small labeled set, we are able to obtain good results that fit very well with the unique needs and issues generally associated with medical imaging datasets.

It is also important to note that unsupervised/self-supervised learning has proven that it produces state-of-the-art results in similar conjectures of few or none labeled datasets; as an example, there are language models, which are currently the clear dominant solution for all natural language processing tasks. The success of these methods comes specifically from the fact that they are able to leverage the vast quantities of language data readily available to learn the underlying structure of language spaces and then only use a small amount of labeled training data to fine-tune the model for the desired task. Very recently similar concepts have begun to be applied to image data, which have shown promising results [1], [2], [3].

2.2 A case for deep domain adaptation

Using the same motivations as above, another research direction that can produce valuable results are transfer learning/domain adaptation methods [4]. This comes from the fact that multiple medical imaging tasks exist which use volumetric data of similar characteristics. While each task will revolve on different types of imaging and focus on different parts of the body, we believe we can leverage the fact that all of them are working on information from the same underlying domain, nuclear imaging of the human body. By using data from multiple tasks to pretrain models, we aim to learn general features from these spaces. Doing this enables us to use multiple small datasets to extract underlying knowledge of the medical image space and then apply this general features to specific tasks by domain adaptation. This way, datasets that present a small number of samples either because it refers to a rare occurrence or data acquisition is expensive, will still manage to obtain good performance by leveraging knowledge from other data domains.

2.3 A case for data privacy

With the increase of medical data digitalization and advances in machine learning, the medical field has a potential to make drastic improvements in its practice. While this is happening, more and more importance is given to the privacy of data, especially inherently highly sensitive medical data. This fact raises privacy concerns when handling patient information in any commercial product or research project. When talking about deep learning algorithms, an additional problem arises, the current lack of reliable model interpretability and privacy guarantees. Only recently, the issue of data privacy has begun gaining attention from the deep learning community as membership inference attacks have been shown to pose a serious threat [5], [6]. With the true extent of possible attacks being unknown and the fact that an ill unmentioned attacker would not disclose his methods to the wider scientific community, the importance of researching attack and defense algorithms is heightened in these early stages of the field.

Due to this, we intend to analyze the models we develop during the Ph.D. program from a data privacy perspective, to find vulnerabilities and propose potential defenses. Since we find this aspect to be of great importance when discussing medical data tasks, any algorithms we propose and compare to previous works will also be compared in this dimension, and we will push for greater awareness of the importance of measuring security.

3 Related and previous work

3.1 ImageCLEF 2020 Tuberculosis

During my master's program, I have worked with Cosmin Pascaru and participated in the ImageCLEF 2020 Tuberculosis challenge. This was a valuable learning experience of working with medical imaging datasets and is partly what fueled the conviction of pursuing such a direction for my Ph.D. The issues encountered in the competition were the very small size of the dataset compared to its task difficulty and the very immense size of each sample.

While we resorted to use a traditional supervised algorithm for the task, in retrospect a self-supervised, domain adaptation approach would have been a much better solution to dealing with the small number of samples and high dimensionality of the data.

3.1.1 Challenge description

"In the 2020 edition, the Tuberculosis task concentrated on the automated CT (Computed Tomography) report generation task, since it has important outcome that can have a major impact in the real-world clinical routines. In order to make the task both more attractive for participants and practically valuable, the report generation is lung-based rather than CT-based, which means the labels for left and right lungs are provided independently. The set of target labels in the CT Report was updated with accordance to the opinion of medical experts. The participants must predict the probability of 3 labels for each lung:

presence of TB lesions in general, presence of pleurisy, and caverns in particular." [7]

3.1.2 Proposed method description

Input data format

The dataset contains chest CT scans of 403 (283 for train and 120 for test) tuberculosis patients. Since the labels are provided on lung-wise scale rather than CT-wise scale, the total number of cases is virtually increased twice. Each 3D CT scan has a shape of $nr\_slices \times 512 \times 512$, where the number of slices is varying around 100. For each CT, two versions of automatically extracted masks of the lungs [8] [9] are provided.

Preprocessing

The preprocessing pipeline consisted of the following steps (some of which were inspired by this paper [10] which detects pulmonary nodules from CT scans):

1. All values were clipped to $[-1200, 600]$;

2. Each value was proportionally scaled to the $[0, 1]$ range accordingly (e.g. -1200 becomes 0.0 and 600 becomes 1.0);

3. Each mask was used to extract the relevant section of the CT. Every value not inside the mask is set to 0.66 (the luminance of common tissues);

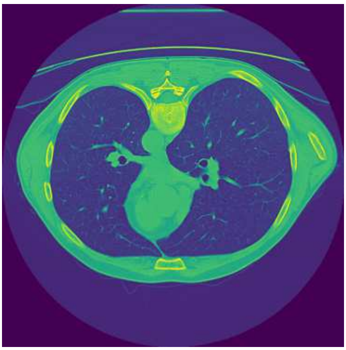
4. There may be some bones present in the CT, which have a very high luminance. Because they may be mistaken for tissue damage or nodules, values above 0.85 are also clipped to the same 0.66 value;

5. Each single lung (which results from applying mask 1) is padded to an aspect ratio of $1 \times 2 \times 1$, and then scaled to a shape S_1. Similarly, each extracted set of 2 lungs (which results from applying mask 2) is padded to an aspect ratio of $1 \times 2 \times 2$, and then scaled to the shape S_2;

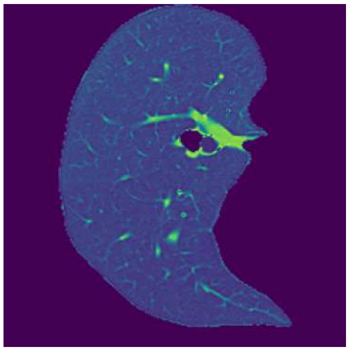
6. After those steps, for each input CT, we end up with three 3D tensors: two of shape S_1, one for each lung, as extracted by mask 1, and one of shape S_2, containing both lungs, as extracted by mask 2;

7. We execute the preprocessing twice, with two choices of S_1 and S_2: one with $S_1 = 64 \times 128 \times 64$, $S_2 = 64 \times 128 \times 128$, and one with $S_1 = 128 \times 256 \times 128$, $S_2 = 128 \times 256 \times 256$. This fundamentally gives us two datasets, one with a faster training time, and one with the more fine details of the CT intact.

An example slice of the CT raw data can be seen in Figure 1a. The following figures show approximately the same area after processing, specifically the left lung (Figure 1b), the right lung (Figure 2a) and both lungs (Figure 2b).



(a) Raw CT slice

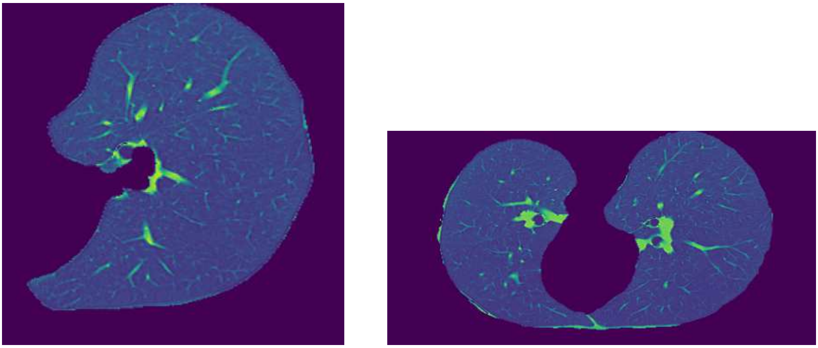


(b) Left lung processed (mask 1)

Figure 1

3.1.3 Training procedure

There are 6 labels that we must predict for each CT: **LeftLungAffected**, **RightLungAffected**, **CavernsLeft**, **CavernsRight**, **PleurisyLeft**, **PleurisyRight**. For each one of those, we must output a probability in $[0, 1]$ of that condition being present.



(a) Right lung processed (mask 1) **(b)** Both lungs processed (mask 2)

Figure 2

For our models, we used a 3D variant of Pre-Activation ResNet with Identity Mapping [11] with 18 layers (PreResNet-18). In fact, 9 models were trained per each preprocessed dataset described above (and 18 in total):

1. Six models were individually trained to only predict each of the six labels, taking the corresponding lung as input, as provided by mask 1.

2. Three models were trained to predict one condition for both lungs at the same time, using the second mask. Due to a more stable training process, the output was the probability of each possible combination of existing / non-existing condition for each given lung (four possible outputs: (0, 0), (0, 1), (1, 0), (1, 1), where 0 represents the non-existing condition, and 1 represents the existing condition, for each lung).

The entire method described above placed us in the 7^{th} place out of 65 teams, but we have noticed several design flaws in our data processing pipeline, which severely hindered our training.

We have revisited this task for the 2021 edition of the challenge [12] and proposed a K-means based projection transformation [13] to reduce the data dimensionality while maintaining the most representative information along the projection direction. We intend to participate in this task in the future with a more advanced approach.

4 Related work

The current state of work being done with medical images is based on convolutional networks, the most widely used by the scientific community being the U-Net [14] due to its ability to extract a lot of information from each sample, working on different levels of generalization of features.

Self-supervision learning for medical imaging tasks has begun to attract attention in the form of image restoration [3] and some unsupervised domain adaptation methods [15], [4]. Building upon such previous work gives us confidence in the potential our research directions pose.

While work in pushing accuracy performance results is continuously being published, data privacy has yet to be adopted by the community as an important metric to track when developing their solutions. In the last couple of years, this has begun to change with interesting results coming out in terms of both attacks and defenses. Currently, the main focus of the field is centered around membership inference attacks and data masking techniques [16], [6]. Another recent development is the appearance of the NeurIPS 2020 hide-and-seek privacy challenge, marking an important step towards motivation research in the field.

5 Future Work

Going forward from previous work, we intend to approach established datasets found on Kaggle, Grand Challenge previous and future ImageCLEF competitions. Some examples include the ”OSIC Pulmonary Fibrosis Progression” competition and the ImageCLEF 2020 - Tuberculosis challenge.

There are six research objectives we currently identified:

O_1: A survey on self-supervised deep learning in medical image analysis. This includes the necessary literature review, re-implementation and analysis of state of the art approaches. We find this first step crucial in order to properly guide the other research objectives. The deliverable for this objective being a survey paper with publication potential;

O_2: A self-supervised approach to region of interest segmentation by healthy sample encoding. Training auto-encoders to learn the general representation of a "healthy" chest CT which will then be used to point out differences to new data in the hopes of identifying abnormal areas. The deliverable for this objective being a research paper with publication potential;

O_3: Domain adaptation/Multitask learning for medical imaging tasks. Using different datasets from brain MRIs to chest CT scans and X-rays, train a base multitask self-supervised model capable of adapting to a specific task with minimal labeled data. The deliverable for this objective being a research paper with publication potential;

O_4: Privacy analysis of deep learning medical imaging solutions. Complete a comprehensive study of the potential vulnerabilities and potential defenses of state of the art machine learning algorithms used for medical data with the goal of comparing various approaches and identifying what techniques are preferable from a data privacy perspective. This analysis will be applied to the $O_{2,3}$ projects as well. The deliverable for this objective being a research paper with publication potential.

These are the general directions we plan to move towards. However, additional projects might appear through the Imago-Mol Cluster (North East Regional Innovative Cluster for Structural and Molecular Imaging) or joint projects between our faculty and "Grigore T. Popa" University of Medicine and Pharmacy. These collaborations include assistance from medical specialists to help guide our work in the objectives mentioned above as well as working on research projects that the cluster participates in, which are specifically medical imaging related.

Auxiliary smaller endeavors, such as the continuation of our previous work, such as Perfect Ordering Approximation and Sum-Augmentation, will continue as side projects and, depending on results, will be utilized in the main objective work.

6 Conclusion

The field of deep learning applied to medical data analysis poses a very exciting research direction. Due to the domain's characteristics, surrounding data scarcity, dimensionality, and privacy, many problems remain to be solved, for which the solutions of self-supervised methods, transfer learning, and domain-specific data augmentation show great promise. We hope that the recent increase in interest for medical tasks and the proliferation of GPU compute capabilities will lead to advances that propagate all the way to the end clients, improving the life of patients and medical professionals.

Acknowledgments. Data processing and analysis in this paper were supported by the Competitiveness Operational Programme Romania, under project SMIS 124759 - RaaS-IS (Research as a Service Iasi).

References

[1] M. Chen, A. Radford, R. Child, J. Wu, H. Jun, P. Dhariwal, D. Luan, and I. Sutskever, "Generative pretraining from pixels," in *Proceedings of the 37th International Conference on Machine Learning*, 2020.

[2] D. Qi, L. Su, J. Song, E. Cui, T. Bharti, and A. Sacheti, "Imagebert: Cross-modal pre-training with large-scale weak-supervised image-text data," *arXiv preprint arXiv:2001.07966*, 2020.

[3] L. Chen, P. Bentley, K. Mori, K. Misawa, M. Fujiwara, and D. Rueckert, "Self-supervised learning for medical image analysis using image context restoration," *Medical image analysis*, vol. 58, p. 101539, 2019.

[4] M. Wang and W. Deng, "Deep visual domain adaptation: A survey," *Neurocomputing*, vol. 312, pp. 135–153, 2018.

[5] R. Shokri, M. Stronati, C. Song, and V. Shmatikov, "Membership inference attacks against machine learning models," in *2017 IEEE Symposium on Security and Privacy (SP)*. IEEE, 2017, pp. 3–18.

[6] S. Truex, L. Liu, M. E. Gursoy, L. Yu, and W. Wei, "Towards demystifying membership inference attacks," *arXiv preprint arXiv:1807.09173*, 2018.

[7] B. Ionescu, H. Müller, R. Péteri, A. B. Abacha, V. Datla, S. A. Hasan, D. Demner-Fushman, S. Kozlovski, V. Liauchuk, Y. D. Cid, V. Kovalev, O. Pelka, C. M. Friedrich, A. G. S. de Herrera, V.-T. Ninh, T.-K. Le, L. Zhou, L. Piras, M. Riegler, P. l Halvorsen, M.-T. Tran, M. Lux, C. Gurrin, D.-T. Dang-Nguyen, J. Chamberlain, A. Clark, A. Campello, D. Fichou, R. Berari, P. Brie, M. Dogariu, L. D. Ştefan, and M. G. Constantin, "Overview of the ImageCLEF 2020: Multimedia retrieval in medical, lifelogging, nature, and internet applications," in *Experimental IR Meets Multilinguality, Multimodality, and Interaction*, ser. Proceedings of the 11th International Conference of the CLEF Association (CLEF 2020), vol. 12260. Thessaloniki, Greece: LNCS Lecture Notes in Computer Science, Springer, September 22-25 2020.

[8] Y. Dicente Cid, O. A. Jiménez del Toro, A. Depeursinge, and H. Müller, "Efficient and fully automatic segmentation of the lungs in ct volumes," in *Proceedings of the VISCERAL Anatomy Grand Challenge at the 2015 IEEE ISBI*, ser. CEUR Workshop Proceedings, O. Goksel, O. A. Jiménez del Toro, A. Foncubierta-Rodríguez, and H. Müller, Eds. CEUR-WS, May 2015, pp. 31–35.

[9] V. Liauchuk and V. Kovalev, "Imageclef 2017: Supervoxels and co-occurrence for tuberculosis ct image classification," in *CLEF2017 Working Notes*, ser. CEUR Workshop Proceedings. Dublin, Ireland: CEUR-WS, September 11-14 2017.

[10] F. Liao, M. Liang, Z. Li, X. Hu, and S. Song, "Evaluate the malignancy of pulmonary nodules using the 3-d deep leaky noisy-or network," *IEEE transactions on neural networks and learning systems*, vol. 30, no. 11, pp. 3484–3495, 2019.

[11] K. He, X. Zhang, S. Ren, and J. Sun, "Identity mappings in deep

residual networks," in *European conference on computer vision.* Springer, 2016, pp. 630–645.

[12] B. Ionescu, H. Müller, R. Péteri, A. Ben Abacha, M. Sarrouti, D. Demner-Fushman, S. A. Hasan, S. Kozlovski, V. Liauchuk, Y. Dicente, V. Kovalev, O. Pelka, A. G. S. de Herrera, J. Jacutprakart, C. M. Friedrich, R. Berari, A. Tauteanu, D. Fichou, P. Brie, M. Dogariu, L. D. Ştefan, M. G. Constantin, J. Chamberlain, A. Campello, A. Clark, T. A. Oliver, H. Moustahfid, A. Popescu, and J. Deshayes-Chossart, "Overview of the ImageCLEF 2021: Multimedia retrieval in medical, nature, internet and social media applications," in *Experimental IR Meets Multilinguality, Multimodality, and Interaction*, ser. Proceedings of the 12th International Conference of the CLEF Association (CLEF 2021). Bucharest, Romania: LNCS Lecture Notes in Computer Science, Springer, September 21-24 2021.

[13] A. Hanganu, C. Simionescu, L.-G. Coca, and A. Iftene, "Uaic2021: Lung analysis for tuberculosis classification," *Proceedings of the Working Notes of CLEF*, pp. 21–24, 2021.

[14] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *International Conference on Medical image computing and computer-assisted intervention.* Springer, 2015, pp. 234–241.

[15] Y. Zhang, Y. Wei, Q. Wu, P. Zhao, S. Niu, J. Huang, and M. Tan, "Collaborative unsupervised domain adaptation for medical image diagnosis," *IEEE Transactions on Image Processing*, vol. 29, pp. 7834–7844, 2020.

[16] M. Abadi, A. Chu, I. Goodfellow, H. B. McMahan, I. Mironov, K. Talwar, and L. Zhang, "Deep learning with differential privacy," in *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, 2016, pp. 308–318.

Cristian Simionescu[1,2]

Received June 13, 2022
Accepted August 17, 2022

[1]"Alexandru Ioan Cuza" University, Faculty of Computer Science, Iasi
[2]Nexus Media SRL
E–mail: `cristian@nexusmedia.ro`

Flower Recognition Using Neural Networks

Bogdan-Antonio Cretu, Adrian Iftene

Abstract

Recognizing flowers and plants can be of interest to the curious, who walk through nature, whether it is a park or a Botanical Garden. In this paper, we will present how we created a neural network and how we train it to recognize flowers. The target is to obtain web service based on a working AI component that can analyze a photo of a plant and say which species that plant belongs to with acceptable accuracy. For this, we created a dataset through web crawling, which we validated and then filtered both automatically and manually.

Keywords: neural networks, image recognition, image dataset.

1 Introduction

Over the last years, hardware and software technologies reached levels hardly imaginable just a decade before. Unrivaled processing power in smaller and smaller devices and the development of AI allows incredible apps to be developed and released in months, with capabilities impossible to reach through traditional methods of programming and on older machines. A phone now has 4 or more CPU cores and it can compute complex operation instantly. This allows emerging technologies to grow faster and faster; such are virtual and augmented reality [1], [2], image recognition [3], [4], language processing [5], [6], and many more.

Image recognition is very easy for humans because the brain is very well designed for doing this operation, however, for computers, it is much more difficult. Image recognition is most commonly implemented using neural networks (NNs), images are turned into vectors of features extracted from the pixel level composition of the image [3], [4]. These pixel features are then used as input for a neural network with a

©2023 by Bogdan-Antonio Cretu, Adrian Iftene

varied amount of "neurons" split across multiple layers. By introducing images that we already have labeled, we can change the weights of neurons and train the network to give a specific output based on the general proprieties of the images used for training. Accuracy is not 100% because all images are different and confusion can appear, but with enough training samples, the network can reach correct prediction rates of upwards of 90%.

2 Existing solutions

In flower recognition, the researchers used neural networks [7], [8] and created applications for botanical gardens [9], [10]. There are a number of apps incorporating various technologies for flower image recognition, and with different features, these have small differences and different capabilities. In the following, we will enumerate the most popular and feature-rich ones.

2.1 PictureThis: Identify Plant, Flower, Weed, and More

PictureThis[1] key features are (see Fig.1): (1) instantly identify thousands of plants, flowers, and trees with advanced artificial intelligence; (2) learn about plants and discover beautiful plant pictures taken by our users around the world; (3) get suggestions and advice from a network of friendly horticulture specialists; (4) plant care tips and water reminders, help you better grow your lovely plants; (5) easy-to-use interface with friendly guides, help you get the best photo; (6) quickly and easily share your photos with a growing community of plant lovers; (7) keep track of all the plants, trees, and flowers you identify in your own personal collection.

2.2 Plant Lens – Plant & Flower Identification

Plant Lens[2] has the following key features (see Fig.2): (1) identify various plants by pictures, more than 60,000 species; (2) keep track

[1] https://play.google.com/store/apps/details?id=cn.danatech.xingseus&hl=en_US&gl=US

[2] https://play.google.com/store/apps/details?id=app.plant.identification

Figure 1. PictureThis

of all the plants, trees, and flowers in your own personal collection; (3) your photos showing on the map creates a personal plant map; (4) explore plant world with an identifier and mobile encyclopedia.

Figure 2. Plant Lens

2.3 LeafSnap - Plant Identification

LeafSnap[3] key features are (see Fig.3): (1) instantly Identify thousands of plants, flowers, fruits and trees with advanced artificial intelligence;

[3] https://play.google.com/store/apps/details?id=plant.identification.snap

(2) learn more about plants, including beautiful pictures around the world; (3) quickly identify plants, flowers, trees, and more. Have instant access to a huge Plant Database that is constantly learning and adding information on new plant species.

Figure 3. LeafSnap

2.4 PlantNet Plant Identification

PlantNet[4] key features are (see Fig.4): (1) the ability to filter recognized species by genus or family; (2) the differentiated data revision that gives more weight to users who have demonstrated the most skills (in particular, the number of species observed, validated by the community); (3) the re-identification of shared observations, whether yours or those of other users of the application; (4) the multi-flora identification that allows you to search for the photographed plant in all the flora of the application and not only in the one you have selected. Very useful when you are not sure what flora to look for; (5) the selection of your favorite floras to access them more quickly; (6) the navigation at different taxonomic levels in image galleries; (7) the mapping of your observations; (8) links to many fact sheets.

[4] https://play.google.com/store/apps/details?id=org.plantnet

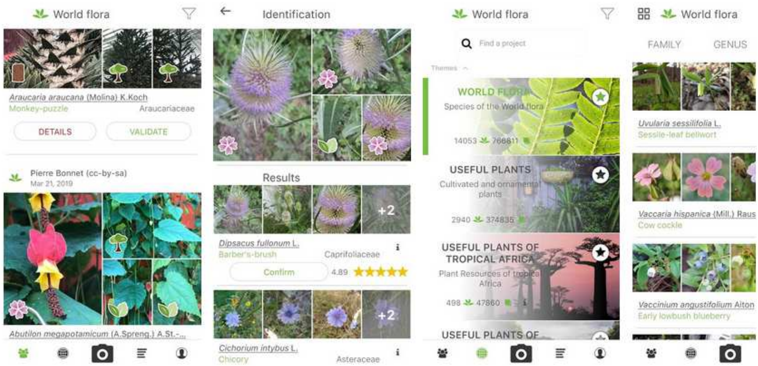


Figure 4. PlantNet

3 Dataset

The chosen theme for the dataset is Flower Species, the first problem encountered was finding a list of common and popular species to train the network to recognize. It was decided to use a list of 300 species [11] available online. The best approach for this, without using an existing dataset or a team to help find relevant images, would be crawling on various search engines in order to find relevant images for each label we want to be able to recognize.

Researching was done regarding ways to crawl various image search engines such as Bing, Yandex, Yahoo, and Google. By seeing through empirical testing that Google will give the most relevant results for the given searches, we decided to create a crawler for it, using python.

3.1 Finding the data

First iteration: One basic crawler that can be done with good results is to simply scroll through the search results page and downloading the links for each results using its ID. However, this method obtains only the thumbnails of the images, which have a small resolution, so the choice to invest in a more resource-consuming method was made. *Second iteration*: An improved crawler that can obtain good quality images, at the price of time, can be done by slowly scrolling through the search results. Clicking each image once, this will generate the URL to the image, which we can take and put into a list. After obtaining a sufficient number of URLs for a search query, we go through each URL and download each image. This method can be very time-consuming, so in order to achieve the dataset in a reasonable amount of time, we limited the search to 500 images per search query. Also, a single encompassing query for each label was used in order to maximize the quality of the download images while keeping the search time as short as possible.

Figure 5. Bad image and good image example found by the crawler

Once the crawler has been set up, and the most common of the possible exceptions have been treated, the crawler can download a few dozens of full labels of images per day, the script ran multiple times using a list of all the labels that have yet to be searched that was updated after each finished search. The dataset obtained this way comprised approximately 148,000 images spread across 300 labels, each with between 480 and 500 images. The images however contained a number of items that were too small or exact duplicates of each other. To eliminate them, a python script was used, utilizing hashing to search for duplicates quickly by simply comparing the hash values of each pair of images, and a simple deletion for images having either width or height lower than a threshold of 180 pixels, chosen arbitrarily. The number of images eliminated this way was surprisingly low; less than 10 images per label were trimmed this way. The dataset had however a good amount of images that were not representative of corresponding labels. The number of wrong images varied between a few dozens and

150-200.

Sorting the bad images was a challenge because automating this process was unachievable without solving the problem of recognizing the flowers in the first place. In the end, we manually filtered each label, deleting any image not containing relevant information. In this process, we were left with between 320-460 images per label, and a total of over 100,000 images across all 300 labels (see Fig.6).

Figure 6. Images from various labels

4 Models

With the final dataset prepared, experimenting begun with various configuration changes to the considered models.

4.1 First experiments

Adding Data augmentation layers to the models reduces the learning speed, and adding dropout further accentuates this. Therefore, the number of epochs must be increased to reach similar results compared to models without data augmentation. However, augmentation and dropout reduce overfitting and allows for better accuracy at the price of longer training time. The details about the considered models are the following (in Table 1 we can see the details about considered runs):

1. *sequential (Conv2D(32), Conv2D(32), MaxPool, Conv2D(32), MaxPool, Conv2D(64), MaxPool,Conv2D(128), MaxPool, Dense(256))*;

Model	Batch size	Number of epochs	Total time	Time per epoch	Validation accuracy	Image resized to
1	32	20	∼ 2.8 hours	∼ 500 sec	0.53	200 x 200
2	32	20	∼ 2.8 hours	∼ 500 sec	0.29	200 x 200
3	32	20	∼ 3.3 Hours	∼ 600 sec	0.28	200 x 200
4	32	50	∼ 7 Hours	∼ 500 sec	0.35	200 x 200

Table 1. Initial experiments on final dataset

2. *sequential (RandomFlip(), RandomRotation(0.2), Conv2D(32), Conv2D(32), MaxPool, Conv2D(32), MaxPool, Conv2D(64), MaxPool, Conv2D(128), MaxPool, Dense(256))*;

3. *sequential (RandomFlip(), RandomRotation(0.2), Conv2D(32), Conv2D(32), MaxPool, Conv2D(32), MaxPool, Conv2D(64), Conv2D (64), MaxPool, Conv2D(64), MaxPool, Conv2D(128), MaxPool, Dense (256))*;

4. *sequential (RandomFlip(), RandomRotation(0.2), Conv2D(32), Conv2D(32), MaxPool, Conv2D(32), MaxPool, Conv2D(64), MaxPool, Conv2D(64), MaxPool, Conv2D(128), MaxPool, Dropout(0.25), Dense(256)).*

The optimizer used for the above models is Adam [12] with a learning rate of 0.001. Training a basic model on the new dataset yields a high Validation Accuracy. However, the model is severely overfitting; therefore, improvements require reducing this comportment. One way of reducing overfit is Augmenting the data, by adding augmentation layers at the start of the model. Most images will be different, they will still be similar and the low initial quantity of data is not completely solved, but this allows for a much lower overfit, with the drawback of slower learning. Another optimization for the model is adding dropout layers. This will make the model even less susceptible to overfit, and allow more exploration while learning; however, the amount of epochs required to obtain a good validation accuracy is increased even further.

It can be observed that adding data augmentation while maintaining the rest of the parameters and model characteristics reduces the validation accuracy by almost half, but increasing the number of epochs puts in perspective the possible growth in performance (see Fig.7).

With added dropout, the 4th model maintains a controlled overfit,

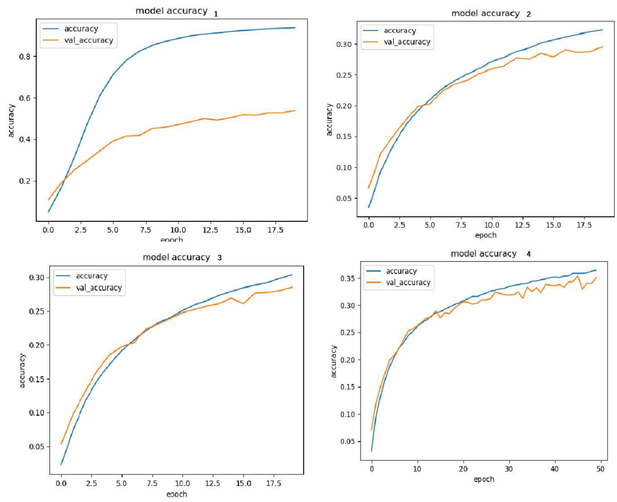


Figure 7. Accuracy comparison for initial experiments on final dataset

with some variation due to the dropout layers. While learning, however, the amount of epochs required to obtain a good validation accuracy is increased even further. It can be seen that that loss maintains a steady decrease during the first 20 epochs, and the validation loss stays stable throughout all epochs (see Fig.8). After 20 epochs the loss and validation loss are bit further from each other, but the validation loss still goes down overall as more epochs pass.

4.2 Adding Normalization and increasing the number of epochs

In order to further improve the convergence of the model, we also added Normalization layers. The increased number of epochs allows for a better observation of the dataset's properties and limitations. The last models also use the Adam optimizer with a learning rate of 0.001. The details about these models are the following (in Table 2 we can see the details about these two runs):

5. *sequential (RandomFlip(), RandomRotation(0.2), Conv2D(32),*

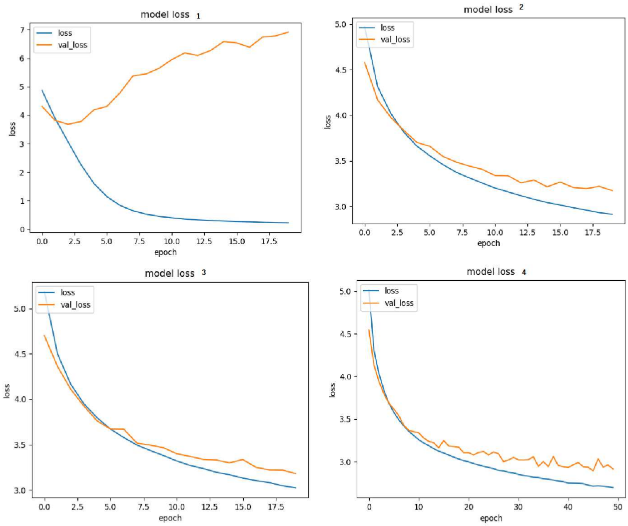


Figure 8. Loss comparison for initial experiments on final dataset

Model	Batch size	Number of epochs	Total time	Time per epoch	Validation accuracy	Image resized to
5	32	50	∼7.4 hours	∼450 sec	0.55	150x150
6	32	100	∼15.5 hours	∼600 sec	0.61	180x180

Table 2. Final experiments

Conv2D(32), MaxPool, Conv2D(64), BatchNormalization(), MaxPool, Conv2D(64), BatchNormalization(), MaxPool, Conv2D(128), BatchNormalization(), MaxPool, Dropout(0.25), Dense(256), BatchNormalization());

6. *sequential (RandomFlip(), RandomRotation(0.2), Conv2D(32), Conv2D(32), MaxPool, Conv2D(64), Conv2D(64), BatchNormalization(), MaxPool, Conv2D(64), BatchNormalization(), MaxPool, Conv 2D(128), BatchNormalization(), MaxPool, Dropout(0.25), Dense(256), BatchNormalization())*.

With the increased amount of epochs, the behavior of the model can

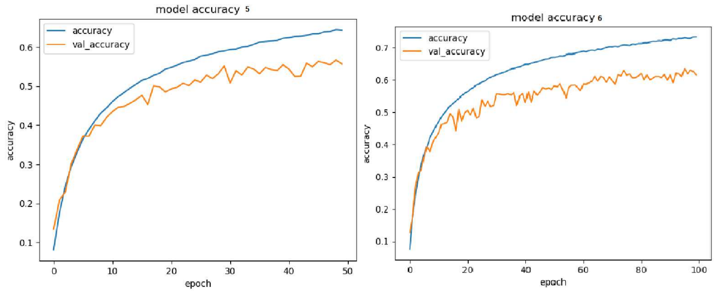


Figure 9. Accuracy comparison with normalization over more epochs

be better seen: the overfit is maintained, the training accuracy remains close to the validation accuracy throughout most of the epochs (see Fig.9), and the validation accuracy has greater variation in the second half of the training process. Having limited memory and processing power (a single device), training for 100 epochs pushes the hardware to its limits (processor package reaches 100 degrees Celsius); however, further tuning to the model and data augmentation layers can slowly increase the reachable validation accuracy for the model.

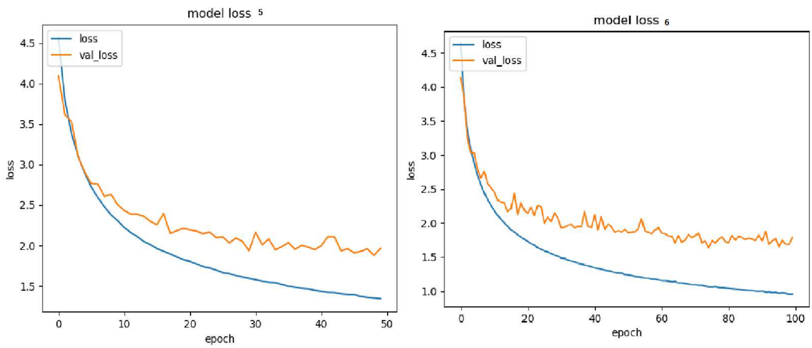


Figure 10. Loss comparison with normalization over more epochs

It can be seen that that loss maintains a steady decrease during the first 20 epochs, and the validation loss stays stable throughout all epochs (see Fig.10). However, after 50 epochs, the loss and validation loss are getting further from each other, and the validation loss starts to jump up and down without maintaining a steady decrease. The limitations of the data are noticeable since some of the flower species are similar, the low amount of initial images per label reduces the potential for correct separation of their features (see Fig.11).

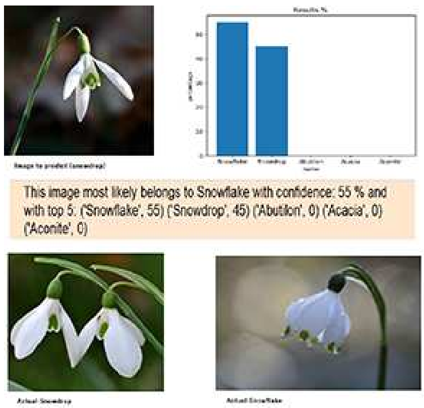


Figure 11. Confusion between Snowflake and Snowdrop flowers

5 Conclusion

It is possible to create and train a neural network for image classification by oneself. The process is slow and requires a lot of time, both in researching methods and technologies and in implementing and training the model. Creating a dataset has the biggest manual workload requirement. If the method of collecting data is through web crawling, each search will need to be filtered from images of the wrong size, wrong aspect ratio, and unfit content. Filtering the images that are not representative is a task that can take from a few days to a few months, depending on the number of labels and images per label searched.

The created flower image dataset is comprised of 299 species of flowers, with an average of 387 images per label. On average 112 images were eliminated per label due to size, content, or repetition reasons. The created dataset was uploaded to Kaggle[5]. Training has some dependence on the used hardware; however, if using a GPU for training, most modern GPUs will have a relatively good performance and over 4 GB of VRAM. In cases where a GPU is not available, the CPU can also be used; however, it may be 5-10 times slower than an average GPU of a similar generation [13].

Both the dataset and the final model obtained could be improved upon, on the dataset side, if some further improvements would be, resizing all the images in advance; also, if more time and resources are available, a second variant of the dataset can be created by increasing the number of images to be searched per label and even by adding alternate names fro labels, and alternate search engines. The model could benefit from a system with more memory, the current variant uses over 5 GB of the 6 GB available on a GTX 166Ti GPU. Testing more layer configurations could also result in finding a better-suited model for the task; however, this requires training each model and testing comparing its results, which can take considerable time and put a strain on the hardware. In the case of a laptop, the temperature reaches 100 degrees Celsius after approximately 10 hours.

Acknowledgments. This work was supported by project REVERT (taRgeted thErapy for adVanced colorEctal canceR paTients), Grant Agreement number: 848098, H2020-SC1-BHC-2018-2020/H2020-SC1-2019-Two-Stage-RTD. Data processing and analysis in this paper were supported by the Competitiveness Operational Programme Romania, under project SMIS 124759 – RaaS-IS (Research as a Service Iasi).

References

[1] M.N. Pinzariu and A. Iftene, "Sphero – Multiplayer Augmented Game (SMAUG)," in *International Conference on Human-Computer Interaction*, (Iasi, Romania, 8-9 September 2016), 2016, pp. 46–49.

[5]https://www.kaggle.com/bogdancretu/flower299

[2] M. Chitaniuc and A. Iftene, "GeoAR-An Augmented Reality Application to Learn Geography," *Romanian Journal of Human-Computer Interaction*, vol. 11, no. 2, pp. 93–108, 2018.

[3] M.D. Minut and A. Iftene, "Creating a Dataset and Models Based on Convolutional Neural Networks to Improve Fruit Classification," in *Proceedings of the 23rd International Symposium on Symbolic and Numeric Algorithms for Scientific Computing (SYNASC)*, (Timisoara, Romania, December 7-10, 2021), 2021.

[4] G.L. Coca, S.C. Romanescu, S.M. Botez, and A. Iftene, "Crack detection system in AWS Cloud using Convolutional neural networks," in *24rd International Conference on Knowledge-Based and Intelligent Information & Engineering Systems, Procedia Computer Science, vol. 176*, 2020, pp. 400–409.

[5] C.G. Cusmuliuc, L.G. Coca, and A. Iftene, "Identifying Fake News on Twitter using Naive Bayes, SVM and Random Forest Distributed Algorithms," in *Proceedings of The 13th Edition of the International Conference on Linguistic Resources and Tools for Processing Romanian Language (ConsILR-2018)*, 2018, pp. 177–188.

[6] L.G. Coca, C.G. Cușmuliuc, and A. Iftene, "CheckThat! 2019 UAICS," in *Working Notes of CLEF 2019 – Conference and Labs of the Evaluation Forum*, (Lugano, Switzerland), September 9-12, 2019, https://dblp.org/rec/html/conf/clef/CocaCI19.

[7] J. Ran, Y. Shi, J. Yu, and D. Li, "A Multi-Feature Convolution Neural Network for Automatic Flower Recognition," *Journal of Circuits, Systems and Computers*, vol. 30, no. 15, Article No. 2150281, 2021.

[8] S. Cao and B. Song, "Visual attentional-driven deep learning method for flower recognition," *Mathematical Biosciences and Engineering*, vol. 18, no. 3, 2021, pp. 1981–1991.

[9] D.B. Chirica and A. Iftene, "Enhancing the Visit of the Botanical Garden of Iasi with the Help of Smart Devices and Augmented Re-

ality," in *Proceedings of the Conference on Mathematical Foundations of Informatics MFOI2020*, (Kyiv, Ukraine), January 12-16, 2021, pp. 78–87.

[10] C.I. Irimia, M. Matei, A. Iftene, S.C. Romanescu, M.R. Lipan, and M. Costandache, "Discover the Wonderful World of Plants with the Help of Smart Devices," in *Proceedings of the 17th International Conference on Human-Computer Interaction RoCHI 2020*, 22-23 October, 2020, pp. 73–80.

[11] D. Samantha, "List of 300 Flowers with examples," 2020. Online. Available: https://florgeous.com/types-of-flowers/.

[12] D. P. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization," *Arxiv, Computer Science, Machine Learning*, 2014. Online. Available: https://arxiv.org/abs/1412.6980.

[13] F. Chollet, *Deep Learning With Python*, Manning, 2017, 384 p. ISBN: 9781617294433.

Bogdan-Antonio Cretu[1,2], Adrian Iftene[1]

Received June 13, 2022
Accepted July 09, 2022

[1] "Alexandru Ioan Cuza" University of Iasi, Faculty of Computer Science
E–mail: adiftene@info.uaic.ro

[2] Endava SRL, Iasi, Romania
E–mail: bogdancretu405@gmail.com

Developing an early predictive system for the diagnosis of Alzheimer's disease and identifying lead biomarkers for this condition

Georgiana-Ingrid Stoleru

Abstract

Alzheimer's disease is a highly prevalent condition and most of the people suffering from it receive the diagnosis late in the process. The diagnosis is currently based on protein biomarkers in cerebrospinal fluid (CSF) and brain imaging together with cognitive tests and a medical history of the individuals. While diagnostic tools based on CSF collections are invasive, brain-imaging tools are costly. Taking these into account, my research aims to develop an early predictive system for the diagnosis of this condition, using Machine Learning, as well as to identify lead biomarkers.

Keywords: Alzheimer's disease, Biomarkers, Machine Learning, Deep Learning, Diagnosis.

1 Introduction

1.1 Background and Motivation

According to the Ulster Medical Journal[1], dementia is "a clinical diagnosis requiring new functional dependence on the basis of progressive cognitive decline"[2]. Furthermore, according to "Centers for Disease Control and Prevention", dementia is the sixth leading cause of death in the United States and the only disease in the ten leading causes

©2023 by Georgiana-Ingrid Stoleru

[1]The Ulster Medical Journal is an international general medical journal which publishes contributions to all areas of medical and surgical specialties.

[2]https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4488926/

of death in the United States, which cannot be cured, prevented or slowed[3].

The most common types of dementia include: Alzheimer's Disease (AD), Dementia with Lewy Bodies, Vascular Dementia, Frontotemporal Dementia and Mixed Dementia. While the clinical symptoms of these diseases overlap significantly, some common symptoms which may occur early before diagnosis include: memory loss, difficulties in concentration, mood changes, finding it hard to carry out familiar daily tasks and struggling to follow a conversation. These symptoms are termed as *"Mild Cognitive Impairment"* (MCI) because they are initially mild and may worsen gradually. Only a few of the patients with MCI will develop dementia in the future, exhibiting problems regarding: memory, communication, mobility, behaviour and appetite.

According to Alzheimer's Association[4], Alzheimer's Disease accounts for 60% to 80% of all the dementia cases[5]. This disease is thought to begin 20 years before symptoms arise, with small changes in the brain which are unnoticeable to the person affected. Only after multiple brain changes do patients experience noticeable symptoms, such as memory loss and language problems. However, at this point, some brain cells have been damaged or destroyed.

1.1.1 Biomarkers: Tools for Diagnosis and Advancing Research

A biomarker is a biological factor that can be measured to indicate the presence or absence of a disease, the risk of developing a disease or the disease progression. Among several factors which are currently studied as biomarkers for Alzheimer's Disease, there can be mentioned the following: the amounts of beta-amyloid and abnormal tau in the

[3]https://www.cdc.gov/dotw/alzheimers/index.html

[4]The Alzheimer's Association is an association which provides support to patients facing dementia and does research in the field of AD concerning treatment and prevention.

[5]https://www.alz.org/alzheimers-dementia/difference-between-dementia-and-alzheimer-s

brain[6], the levels of beta-amyloid and tau in cerebrospinal fluid, and the level of glucose metabolism in the brain[7]. Besides these factors, specialists need to obtain the medical history of the individuals, to collect data about changes in their behavior and to conduct several cognitive tests and neurologic examinations.

1.1.2 Prevalence, Mortality and Morbidity

Research conducted by the Alzheimer's Society[8] shows that in 2019, there were over 850,000 people with dementia in the UK[9]. At the current rate of prevalence, in 2040, there will be over 1.5 million people with dementia in the UK. According to Bright Focus Foundation[10], nearly 50 million people worldwide have Alzheimer's or related dementia. Moreover, the Centers for Disease Control and Prevention (CDC)[11] states that 121,404 people died from Alzheimer's disease in 2017[12]. The Alzheimer's Association estimates that between 2017 and 2025 every state is expected to see at least a 14% rise in the prevalence of Alzheimer's[13].

1.1.3 The Financial Cost

Because of high failure rates and long time required for clinical trials, Alzheimer's drugs cost twice the average of other drugs and seven times

[6] The accumulation of the protein fragment beta-amyloid outside neurons and the abnormal form of the protein tau inside neurons are two of the several brain changes associated with AD. Beta-amyloid plaques may contribute to cell death, while tau tangles block the transport of nutrients and other essential molecules inside neurons.

[7] The ability of the brain to metabolize glucose (its main fuel) decreases in Alzheimer's disease.

[8] The Alzheimer's Society is a UK care and research charity for people with dementia and their carers.

[9] https://www.alzheimers.org.uk/about-us/policy-and-influencing/what-we-think/demography

[10] The Bright Focus Foundation is a nonprofit organization which supports research in the field of brain and eye diseases, such as Alzheimer's disease, macular degeneration and glaucoma.

[11] The Centers for Disease Control and Prevention (CDC) is the national public health agency of the United States.

[12] https://www.cdc.gov/nchs/data/nvsr/nvsr68/nvsr68_02-508.pdf

[13] https://www.alzheimers.net/alzheimers-is-on-the-rise-in-these-states

as much as cancer drugs [1]. Finding a simple and inexpensive test to diagnose Alzheimer's would represent a major progress towards dealing with this condition.

2 Research Topic and Scope

2.1 Research Scope

2.1.1 The early diagnosis of Alzheimer's Disease

While there are several current efforts in this area, the accuracy of the developed systems is not high enough in order to be safely used in practice. The datasets contain values of clinical variables collected through means of family medicine, as well as cognitive tests and MRI images.

2.1.2 Identifying new lead biomarkers for this condition

New lead biomarkers would not only facilitate the development of an early diagnostic system, but would also represent a step forward towards reducing the amount of invasive analysis required.

2.1.3 Developing a prediction tool with high precision

The main purpose of my research is to develop a predictive system for the diagnosis of Alzheimer's Disease with higher accuracy than the state of the art. This also implies a sustained effort in the direction of identifying lead biomarkers, which would facilitate the development of such a system, using machine learning techniques.

3 Related Work

3.1 Prescription claims-based machine learning models

Nori et. al [2] used data collected between 2001 and 2015 from the OptumLabs Data Warehouse (OLDW)[14], a database containing med-

[14]The OptumLabs Data Warehouse is a comprehensive, real-world data asset, which contains longitudinal health information on enrollees and patients across the United States.

ical information about more than 125 million privately insured individuals. The dataset contains professional, facility and outpatient prescription medication claims. In the first phase, variable selection was performed, using Lasso regression algorithm. 50 important predictors were selected, which further became input for the Logistic Regression algorithm. This approach led to an average sensitivity equal to 31.9% and a specificity equal to 86.4%.

Nori et. al [3] also used a dataset provided by OLDW, which included not only medical and pharmacy claims, but also laboratory results and enrollment records for commercial and Medicare Advantage enrollees. The authors compared models obtained using three test datasets to better understand how clinical data adds information to the diagnosis. The simplest one contained claims-only. The second one added the cognitive test results and EMR diagnoses and prescriptions. As opposed to these two datasets, which used 2 years of continuous enrollment, the last one did not include continuous enrollment. Instead, it used a health care encounter in a calendar year as an indicator of a patient's ability to access health care services. By applying LightGBM, the authors obtained a sensitivity of 47% and an AUC value equal to 0.87. The top 10 features, which explained the model prediction, included codes indicating altered mental status, magnetic resonance, neurological diseases symptoms, neuropsychological testing, and psychotic, schizophrenic disorders.

3.2 A metabolite-based machine learning model

The advantage of analyzing blood metabolites is that they are easily accessible and they might act as molecular fingerprint of disease progression. Petroula Proitsi et al. [4] performed a comprehensive lipidomic analysis in an attempt to identify metabolites associated with AD status. Using a Random Forest approach, the authors identified a combination of 10 metabolites, which predicted AD with near 80% accuracy. The newly identified molecules were reduced in AD patients compared with control groups.

Daniel Stamate et al. [5] showed that plasma metabolites have the potential to match the state of the art results obtained using AD CSF biomarkers. The dataset used for this research contained metabolite data derived from blood samples from 357 participants (242 cognitively normal patients, 115 patients diagnosed with AD). Two state-of-the-art ML algorithms were employed in this study: Deep Learning and Extreme Gradient Boosting. While the Deep Learning model produced the AUC value of 0.85 with its 95% CI ranging between 0.8038 and 0.8895, the XGBoost model produced the AUC value of 0.88 with its 95% CI ranging between 0.8619 and 0.8903.

3.3 A deep learning model based on hippocampal MRI

It is a challenging task to predict when and which individuals who meet the criteria for MCI will ultimately progress to AD. Li et al. [6] developed a deep learning method based on MRI scans in order to predict MCI progression to AD dementia. A convolutional neural network with residual connections was trained to learn informative features from the MRIs for distinguishing AD from NC subjects. The output streams were then flattened and concatenated as input to a fully connected layer for building the classification model. Based on the deep learning features, there was built a model for predicting MCI to AD progression, using LASSO regularized Cox regression model. The AUC value of this model reached the value 0.813.

3.4 Discovering small-molecule biomarkers for Alzheimer's disease

Whiley et al. [7] presented several strategies for the discovery of small-molecule biomarkers in AD, providing examples of molecules of interest and their importance to the understanding of this disease. One of the factors suggested by the authors is represented by the irregularities in lipids and their metabolite concentration found in samples of AD patients. According to the authors, this area may potentially be worth a great deal of further investigation, providing important biomarkers for the disease.

4 Research Objectives

Taking these into account, the objectives set for my research are the following:

- To develop Machine Learning models for early diagnosis of Alzheimer's Disease with an accuracy close to the practicality threshold for an AD risk prediction tool.
- To identify new lead biomarkers which can be used in the development of such a model, without requiring invasive analysis or highly costly ones (such as CSF collection or brain imaging data).
- To validate the results with research teams having expertise in medicine.

5 Conclusion

This condition affects a large percentage of people worldwide. For the patients suffering from this condition, the diagnosis is made at a late stage of the disease, when the brain cells are already damaged or destroyed. Furthermore, the current available tests used for diagnosis are either expensive or invasive. The available drugs are also highly expensive and the evolution of the disease itself can not be slowed down. In conclusion, the development of an early diagnosis system using less invasive and less expensive biomarkers could considerably improve patients lives.

Acknowledgments. Data processing and analysis in this paper were supported by the Competitiveness Operational Programme Romania, under project SMIS 124759 - RaaS-IS (Research as a Service Iasi).

References

[1] J. Cummings, C. Reiber, and P. Kumarb, "The price of progress: Funding and financing Alzheimer's disease drug development," *Alzheimer's & Dementia: Translational Research & Clinical Interventions*, vol. 4, no. 1, pp. 330–343, 13 Jun 2018, doi:10.1016/j.trci.2018.04.008.

[2] V.S. Nori, C.A. Hane, D.C. Martin, A.D. Kravetz, and D.M. Sanghavi, "Identifying incident dementia by applying machine learning to a very large administrative claims dataset," *PLoS One*, vol. 14, no. 7, 5 Jul 2019, Article no. e0203246, doi: 10.1371/journal.pone.0203246, PMID: 31276468; PMCID: PMC6611655.

[3] V.S. Nori, C.A. Hane, W.H. Crown, R. Au, W.J. Burke, D.M. Sanghavi, and P. Bleicher, "Machine learning models to predict onset of dementia: A label learning approach," *Alzheimer's & Dementia: Translational Research & Clinical Interventions*, vol. 5, no. 1, pp. 918–925, 10 Dec 2019, doi: 10.1016/j.trci.2019.10.006.

[4] P. Proitsi, M. Kim, L. Whiley, et al., "Plasma lipidomics analysis finds long chain cholesteryl esters to be associated with Alzheimer's disease," *Transl Psychiatry*, vol. 5, Article no. e494, 13 Jan 2015, doi: 10.1038/tp.2014.127.

[5] D. Stamate, M. Kim, P. Proitsi, et al., "A metabolite-based machine learning approach to diagnose Alzheimer-type dementia in blood: Results from the European Medical Information Framework for Alzheimer disease biomarker discovery cohort," *Alzheimer's & Dementia: Translational Research & Clinical Interventions*, vol. 5, no. 1, pp. 933–938, 18 Dec 2019, doi: 10.1016/j.trci.2019.11.001.

[6] H. Li, M. Habes, D.A. Wolk, and Y. Fan, "A deep learning model for early prediction of Alzheimer's disease dementia based on hippocampal MRI," *Alzheimer's & Dementia*, vol. 15, no. 8, pp. 1059-1070, 2019, doi:10.1016/j.jalz.2019.02.007.

[7] L. Whiley and C. Legido-Quigley, "Current strategies in the discovery of small-molecule biomarkers for Alzheimers disease," *Bioanalysis*, vol. 3, no. 10, 18 May 2011, doi:10.4155/bio.11.62.

Georgiana-Ingrid Stoleru[1,2]

Received June 13, 2022
Accepted August 24, 2022

[1]Faculty of Computer Science, "Alexandru Ioan Cuza" University of Iasi
[2]Bitdefender, Iasi, Romania
E–mail: `ingridstoleru@gmail.com`

Decentralized Infrastructure for Digital Notarizing, Signing and Sharing Files Securely using Blockchain

Cosmin-Iulian Irimia

Abstract

Taking care of bureaucracy problems is probably one of the least enjoyable things a person can do. The issuance of paper documents is already thousands of years old and to this day it has certainly evolved, but we still rely on a very rudimentary form of trust. Besides the fact that the plain paper is very easy to destroy, the documents can be very easily forged, lost or stolen, which has created great problems throughout history. Although many security experts are still working to make physical documents more and more difficult to falsify, technology has revealed new ways to improve their security, to emphasize the trust [1] offered in an official document. In this paper, we want to express the arguments for which digital document storage solutions are clearly superior.

Keywords: blockchain, digital identities, computer vision, digital notarizing, digital file management.

1 Introduction

All systems that provide access to electronic documents [2] work independently; this comes naturally with the need to verify additional documents [3] and mainly provide access to more information than necessary. The EU COVID-19 Vaccine Certificate [4], for example, can be verified only if it is accompanied by an identity card to ensure that the person on the certificate [5] is the one presenting the document [6]. This additional process is redundant and creates another problem: it

©2023 by Cosmin-Iulian Irimia

provides access to more information than is necessary. When we present a document certifying that *we are vaccinated*, we should not provide *the address, date of birth or number and series of the identity card*.

The evolution of Blockchain technology in recent years has been a stellar one and having Ethereum working right now towards the transition from Proof-of-work to Proof-of-stake will allow us to write smart contracts that are cost and speed efficient [7], [8]. This is in addition to the already known benefits such as the high level of security [9] and the indisputable trust [10] of the existing information in the blocks.

By taking advantage of this great evolution and by combining this important problem and a state-of-the-art technological solution, this paper proposes a distributed system [11] of notarization official documents.

2 Similar Applications

All the services we found have a simple interface between a client and a physical notary and provide online consulting in this area and an official seal on the document.

2.1 Notarize[1]

Notarize's process is simple and typically takes less than five minutes. You can register on Notarize's app or website, upload your document, and verify your identity. Then you can connect with a live notary who witnesses your electronic signature, signs, and adds a notarial seal on your document. You will have digital access to your notarized document or you may send it elsewhere.

2.2 NotaryCam[2]

NotaryCam also offers a face-to-face meeting with a live notary trained through the company's own NotaryCam Academy. High-speed internet access and a desktop or laptop computer with a webcam and audio capabilities are required. NotaryCam's three-step process takes a few minutes. You can upload your document, connect with a live notary to

[1]https://www.notarize.com/
[2]https://www.notarycam.com/

confirm your identity, and they will electronically sign and apply the eNotary seal.

2.3 Nexsys[3]

Nexsys Technologies connects home buyers and closing agents through a platform called Clear Sign. The process involves a web-enabled desktop or laptop computer and mobile device for added security. Nexsys Technologies' Clear Sign may be faster than in-person transactions, but the process varies depending on how much paperwork you have. You can start by testing the audio-visual software and confirming your identity by scanning your government-issued identification. After confirming your identity, the software connects you with a live notary to walk through each page of your closing documents. The notary adds a digital stamp, and you receive digital access to the documents, along with a video recording of your meeting.

Even if all the applications seem a good solution for certain documents that we can think about at the moment, in reality, after the notary closes the call, the generated pdf can be altered, and the content changed.

3 Related Previous Work

Over the years, we tried to push things in the right direction by creating software solutions using blockchain technology to solve certain problems that did not have solutions using classical programming paradigms. We will quickly present just one of them that is kind of related to our problem. The project is called "Demochain: a solution for e-voting at large scale" [12] and it's a voting solution based on blockchain created as a dissertation thesis. The system has multiple components: the front-end part that ensures the authentication part, interaction with the system, and data visualization; the backend part that ensures secure communication and access to external services and the smart contract, where the actual data is kept.

In the first phase, we focused on ease of use: A normal user will have only two buttons: Login and Vote. The authentication is made on the

[3]https://www.nexsystech.com/

basis of two consecutive pictures: the first – with the person who wants to vote and the second – with the identity card. Once logged in, if there is an open voting session, the voter can vote for their preferred option. A strong point of our solution is that authentication is credential less so that users do not feel the pressure to remember another username and password.

Transparency was achieved by using a public ethereum blockchain, where all transactions remain recorded and nothing can be changed. Certainly, there are things we had to think about so as not to give too much access, but we tried as much as possible to maintain the transparency of the voting process. Also, the transparency of the count is indisputable because the system counts the live votes and there is not even a small chance to slip counting errors.

We offer security in several ways, first of all, the blockchain itself brings a high level of security due to the adoption of new blocks based on consensus between nodes. For communication between web client and server, we used HTTPS; and to add an extra layer of security, we implemented our own mechanism for asymmetric encryption of messages based on fingerprints generated by the device from which the login is made. In addition to security at the level of architecture [13], we also have security at the level of smart contract, so there are several types of users with more or less permissions depending on the needs of each.

A few things that are worth to be mentioned are:

- Authentication that is also based on the context of the photos, if there are several people in the picture, the authentication will not be successful or if the person shows too much anger or stress after identifying the emotions in the picture, the authentication will not pass as well. Compared to current systems, Demochain allows the addition of attachment files to the voting option and also details about them to ensure that voters can get secure information, in order to combat fake-news that affects voting.

- The system allows double identity verification, after demochain filters most of the inappropriate users who try to abuse their voting power, a suite of administrators appointed by the owner can individually verify users before they vote using the data taken at login.

- It is also worth mentioning that we ran a set of usability tests on 60 people but from different professional areas in the sense that we had 2 groups: one technical and one non-technical. The tests took place in two phases: in the first phase, the first group played the role of admin and the second group had the role of voter, after this in the second phase they changed roles to obtain more relevant data. There were three tasks for each type of actor: for the voter, users had to register in the application, then vote for one of the two preferences offered and at the end of the vote to follow the results. For the administrator, the users had to login and navigate in the management menu, to view the data of a certain user in order to approve it in the system and respectively they had to offer the actual access to the vote to a certain user. From our observation during the test sessions, all the people involved were very satisfied with the application and say that they would use the Demochain application to the detriment of the classic vote if this were an option. The participants were asked to rate their experience with a note from 1 to 5, and the results helped us to understand where we need to work more and where the experience is already great.

4 Our Proposal

4.1 Planning

Going forward from previous work, we intend to develop a decentralized infrastructure [14] and a system for the adoption, signing, and sharing of the documents securely. This system will be able to certify that a certain document that was added to the blockchain is real and identical to the original in order to be signed and used digitally to facilitate certain bureaucratic processes and to get rid of paper documents once and forever. There are four research objectives we currently identified:

1. Discovering the field, identifying the existing solutions and trying

to understand in depth the main problems that arose in their development, knowing the target group, and differentiating our solution from those already established in the market.

2. Sketching a generic architecture to solve the basic problem from where we can go further in the development of the system. Identifying the main components and technologies, comparing them, and identifying their strengths and weaknesses. Establishing an architectural design plan and defining the actors.

3. Progressively implement the system in an Agile manner in order to be able to identify possible problems with the system and to be able to fix them along the way.

4. Testing and evaluating the system in collaboration with an institution or a set of users who can use the system to better understand the problems that an end-user might encounter and to fix them.

4.2 Architecture

At this moment, the project is in the research stage and the main mention we want to bring is that the architecture will have small adjustments along the way depending on the feasibility / implementation and the new ideas that will come along the way. The application will be built on three main layers, and all of them will have sub-layers as well, but we will detail this below.

4.2.1 User Interface

We want the interface part to offer both a web application and mobile applications for the two major application distribution platforms: iOS's AppStore and Android's Google Play Store. In order to get to a Beta phase quickly, we will most likely have to write the interface using React Native to write the code once for all three platforms: Web, iOS and Android. The controls, the forms, the positioning, and the color palette are currently insignificant things, and we will decide them later in the process. The communication with the server will be done using

HTTP requests and the main functionalities that we can distinguish at this moment would be the following:

Registration, where we have two options: a short video call, where the user is manually authenticated by a moderator, or the authentication variant using pictures on the ID Card and, respectively, a short video in which the user can do certain actions that we will send on the screen (turn your head to the left, etc.). This method already exists on the market in certain banking solutions and works successfully. After authentication, the user will receive his own DID (Decentralized Identity) which he will have to keep in order to further interact with the application.

Notarizing an official document [15] itself that will require scanning it in good light conditions and together with the DID will generate a pair of data that once reached the server will generate the digital copy, notarized, and added to the blockchain.

Retrieve and share an existing document: the user chooses from the collection of notarized documents one he wants, deobfuscates the parts of interest in the document for the use-case he has (for example: showing your ID for Covid19 Green Pass verification), and shows it digitally from his phone or generates a QR code that he can share with the 3rd party.

4.2.2 Server and Computation

The main server is the one whose role is to ensure the security [16] of the access to the blockchain network and then to communicate with other micro-services for identification, processing and document signing. At this point, the two microservices we can identify are:

Sign Server The sign server is a very important module that will receive an official document and will notarize it. Using official mechanisms, the legislation and a group of lawyers and notaries will enable the process of notarization. The notarized document will be digitally signed, obfuscated ant then stored in the blockchain [17] in order to be verified and retrieved as long as necessary.

Image Processor: The Image Processor is the part that will handle the digital processing [18] of documents. Receiving a document already notarized, this system will use its facial identification and text identification submodules to be able to recognize areas that contain private information or confidential information. Both the faces and the private information (name, address, etc.) identified, but also groups of information that the user chooses as private will be obfuscated one by one, using the Zone Obfuscator Module, generating an obfuscation key for each of them, the benefit of this system being the possibility to deobfuscate specific layers depending on the user's use-case. These unlock keys will most likely be kept in a secure database, mapped, and accessed by each person's DID. We will try to briefly exemplify the steps of obfuscation and deobfuscation.

The first step is to identify the faces and the private information in the official document. The system will detect the face of the person trying to notarize this identity card, then the private information such as the person's name, sex, date of birth, CNP, series and expiration date of the document, etc. Figure 1 below is a good representation on how the work of the system will identify these things.

Figure 1. Faces and Private Information Identification

The second step is to apply the noise generated using user's DID and other unique details to obfuscate one by one all the information identified above, generating a key for each of them [19]. Figure 2 helps us get a better view of this process.

Figure 2. Obfuscation and Keys Generation

5 Conclusion

Given that the chosen field is one of great importance, where there are many problems and the progress made so far is low, the proposed solution is welcome in the industry, with huge potential to solve, at least partially, the problem of excessive bureaucracy, loss, falsification, and damage of physical documents and, especially, keeping them in a safe and easy place so that we do not have to physically carry them with us all the time.

In addition to the ease of use it proposes, the system also comes with new benefits compared to the classic system, namely the sharing of partial private information, using granular keys on each private part of the document, also the ability to share entire folders of documents without endangering the information in them and especially, proving certain things without the need for too much access by foreigners.

We believe that this system will evolve very well and that it will be a real benefit for the community.

Acknowledgments. Data processing and analysis in this paper were supported by the Competitiveness Operational Programme Romania, under project SMIS 124759 - RaaS-IS (Research as a Service Iasi).

References

[1] T. Xu, Z. Fu, M. Yu, J. Wang, H. Liu, and T. Qiu, "Blockchain Based Data Protection Framework for IoT in Untrusted Storage," in *2021 IEEE 24th International Conference on Computer Supported Cooperative Work in Design (CSCWD)*, 2021, pp. 813–818.

[2] N. I. Chesnokov, D. A. Korochentsev, L. V. Cherckesova, O. A. Safaryan, V. E. Chumakov, and A. Pilipenko, "Software Development of Electronic Digital Signature Generation at Institution Electronic Document Circulation," in *2020 IEEE East-West Design and Test Symposium (EWDTS)*, 2020, pp. 1–5.

[3] S. Mthethwa, N. Dlamini, and G. Barbour, "Proposing a Blockchain-based Solution to Verify the Integrity of Hardcopy Documents," in *2018 International Conference on Intelligent and Innovative Computing Applications (ICONIC)*, 2018, pp. 1–5.

[4] M. Hasan, A. Rahman, and M. J. Islam, "DistB-CVS: A Distributed Secure Blockchain based Online Certificate Verification System from Bangladesh Perspective," in *2020 2nd International Conference on Advanced Information and Communication Technology (ICAICT)*, 2020, pp. 460–465.

[5] A. Garba, Q. Hu, Z. Chen, and M. R. Asghar, "BB-PKI: Blockchain-Based Public Key Infrastructure Certificate Management," in *2020 IEEE 22nd International Conference on High Performance Computing and Communications; IEEE 18th International Conference on Smart City; IEEE 6th International Conference on Data Science and Systems (HPCC/SmartCity/DSS)*, 2020, pp. 824–829.

[6] I. T. Imam, Y. Arafat, K. S. Alam, and S. A. Shahriyar, "DOC-BLOCK: A Blockchain Based Authentication System for Digital Documents," in *2021 Third International Conference on Intelli-*

gent Communication Technologies and Virtual Mobile Networks (ICICV), 2021, pp. 1262–1267.

[7] M. Alizadeh, K. Andersson, and O. Schelén, "Efficient Decentralized Data Storage Based on Public Blockchain and IPFS," in *2020 IEEE Asia-Pacific Conference on Computer Science and Data Engineering (CSDE)*, 2020, pp. 1–8.

[8] J. Cheng, N. Lee, C. Chi, and Y. Chen, "Blockchain and smart contract for digital certificate," in *2018 IEEE International Conference on Applied System Invention (ICASI)*, 2018, pp. 1046–1051.

[9] P. Haveri, U. B. Rashmi, D. G. Narayan, K. Nagaratna, and K. Shivaraj, "EduBlock: Securing Educational Documents using Blockchain Technology," in *2020 11th International Conference on Computing, Communication and Networking Technologies (ICCCNT)*, 2020, pp. 1–7.

[10] A. G. Wang, B. White, and R. L. Cottrell, "A Blockchain-Based Decentralized Data Storage and Access Framework for PingER," in *2018 17th IEEE International Conference On Trust, Security And Privacy In Computing And Communications/ 12th IEEE International Conference On Big Data Science And Engineering (TrustCom/BigDataSE)*, 2018, pp. 1303–1308.

[11] R. Kumar and R. Tripathi, "Implementation of Distributed File Storage and Access Framework using IPFS and Blockchain," in *2019 Fifth International Conference on Image Information Processing (ICIIP)*, 2019, pp. 246–251.

[12] C.I. Irimia, A. Iftene, and D. Gîfu, "A Large-Scale E-voting System Based on Blockchain," in *J29th International Conference on Information Systems Development (ISD2021)*, (September 8-10 2021, Valencia, Spain), Universitat Politècnica de València, 2021.

[13] Q. Lu, M. Staples, H. O'Connor, S. Chen, and A. Guabtni, "Software Architecture for Blockchain-based Trade Certificate Systems," in *2020 IEEE International Conference on Blockchain and Cryptocurrency (ICBC)*, 2020, pp. 1–3.

[14] M. Shah, M. Shaikh, V. Mishra, and G. Tuscano, "Decentralized Cloud Storage Using Blockchain," in *2020 4th International*

Conference on Trends in Electronics and Informatics (ICOEI) (48184), 2020, pp. 384–389.

[15] H. Magrahi, N. Omrane, O. Senot, and R. Jaziri, "NFB: A Protocol for Notarizing Files over the Blockchain," in *2018 9th IFIP International Conference on New Technologies, Mobility and Security (NTMS)*, 2018, pp. 1–4.

[16] R. Pise and S. Patil, "Enhancing Security of Data in Cloud Storage using Decentralized Blockchain," in *2021 Third International Conference on Intelligent Communication Technologies and Virtual Mobile Networks (ICICV)*, 2021, pp. 161–167.

[17] I. Afrianto, and Y. Heryanto, "Design and Implementation of Work Training Certificate Verification Based On Public Blockchain Platform," in *2020 Fifth International Conference on Informatics and Computing (ICIC)*, 2020, pp. 1–8.

[18] R. J. Kutty and N. Javed, "Secure Blockchain for Admission Processing in Educational Institutions," in *2021 International Conference on Computer Communication and Informatics (ICCCI)*, 2021, pp. 1–4.

[19] S. Noh and K. -H. Rhee, "Implicit Authentication in Neural Key Exchange Based on the Randomization of the Public Blockchain," in *2020 IEEE International Conference on Blockchain (Blockchain)*, 2020, pp. 545–549.

Cosmin-Iulian Irimia[1,2]

Received June 13, 2022
Accepted July 26, 2022

[1] "Alexandru Ioan Cuza" University of Iasi, Faculty of Computer Science
E–mail: `irimia.cosmin@gmail.com`

[2] Deloitte Tehnologie Romania
E–mail: `cirimia@deloittece.com`

Timed Modal Logic for rTiMo

Bogdan Aman, Gabriel Ciobanu

Abstract

rTiMo is a process calculus that uses timeouts to describe the communication and migration taking place in distributed real-time systems. In order to describe properties of such computations we devise a timed modal logic and study its expressiveness by constructing characteristic formulas for rTiMo processes.

Keywords: timed process calculus, timed modal logic, migration, communication.

1 Introduction

Process calculi are formal frameworks that provide high-level descriptions for the behaviour of agents/processes in concurrent systems. A process calculus has several features: (i) in order to communicate agents/processes use message-passing along shared channels; (ii) there exists few operators and primitives that are used to describe the systems; (iii) behavioural equivalence and equational reasoning can be used to manipulate the agents/processes. The existence of a parallel composition operator in process calculi sets them apart from other models of computation; this feature allows to put in parallel small modules in order to model large systems.

In [1],[2] we introduced and studied a formalism called rTiMo (real-Time Mobility) that uses timeouts to describe the migration and communication taking place in distributed real-time systems. Coordination among agents present in real-time distributed systems is performed by imposing that the communication and migration take place within given timed constraints. Overall rTiMo has several advantages that make it useful in describing real-time systems: (i) it is able to explicitly model distributed locations and the migration of agents between

©2023 by Bogdan Aman, Gabriel Ciobanu

them; (ii) the agents execute in parallel, and interact only locally to communicate; (iii) due to the parallel operator, it is natural to create complex systems from the smaller ones; (iv) behavioural equivalence can be used to replace parts of a system with equivalent ones.

A formal framework to model systems and reason about their behaviours is given by (temporal) modal logics [3], [4]. Reasoning about specific spatial and timing requirements of real-time systems described in rTiMo , requires that we introduce an expressive logic, that includes spatial and time operators inspired from [5], [6].

The structure of the paper is as follows: rTiMo process calculus is presented in Section 2, while Section 3 contains the logic we define in this paper, together with several results regarding the expressiveness of our logic: we consider some logical formulas for capturing some primitives of the rTiMo calculus. The conclusion and references end the article.

2 rTiMo Calculus

In rTiMo calculus, a system consists of several distributed locations and agents residing in them. Agents are able to change their current locations by migrating to other locations. The coordination among the agents of real-time distributed systems is performed by imposing that the communication and migration take place within given timed constraints. In rTiMo , the agents perform local communication and migration between locations in parallel, while the passage of time in the entire system is modelled using a global clock.

The timeout of a migration action specifies how many time units need to pass before a process is able to continue executing at a new location. Agents execute in parallel, and are able to communicate if some conditions are satisfied: they use shared channels, reside at the same location at the same instance of time. We use labelled transitions in order to give the semantics of rTiMo ; the labels are created by using the multisets of actions performed in parallel in the same time framework.

We use $^{\Delta t}$ to denote a timeout of length $t \in \mathbb{R}_+$. A timer $^{\Delta 3}$ appearing in the migration action of a process $go^{\Delta 3} l$ then P marks the fact that process P cannot execute at the current location for 3 time units, but will be able to execute at location l after the 3 time units have passed. A timer $^{\Delta 5}$ appearing in the output communication action of a process $a^{\Delta 5}!\langle z\rangle$ then P else Q marks the fact that the value z can be communicated along channel a for at most 5 time units. In a similar manner, a timer $^{\Delta 4}$ appearing in the input communication action of a process $a^{\Delta 4}?(x)$ then P else Q marks the fact that the value x can be updated with the value received along channel a in at most 4 time units. In both cases, input and output actions, if the communication takes place before the timer expires, then the execution proceeds with process P; otherwise, the execution proceeds with process Q.

Table 1 contains the syntax of rTiMo , where the following assumptions are made:

- there exist three sets *Loc*, *Chan* and *Id* containing the locations l, the communication channels a and the process identifiers id, respectively;
- each process identifier $id \in Id$ has an arity m_{id} that represents the number of distinct parameters u_i appearing in the unique process definition $id(u_1, \dots, u_{m_{id}}) \stackrel{def}{=} P_{id}$;
- t, u and v represent a positive real-timeout, a variable and an expression, respectively; the expressions are recursively made out of variables, values and operations.

Processes	P, Q	$::=$	$a^{\Delta t}!\langle v\rangle$ then P else Q $\mid$	(output)
			$a^{\Delta t}?(u)$ then P else Q $\mid$	(input)
			$go^{\Delta t} l$ then P $\mid$	(move)
			0 $\mid$	(termination)
			$id(v)$ $\mid$	(recursion)
			$P \mid Q$	(parallel)
Located Processes	L	$::=$	$l[[P]]$	
Systems	N	$::=$	L $\mid$ $L \mid N$	

Table 1. rTiMo Syntax

A variable u is bound only if it appears in the input action of a process $a^{\Delta t}?(u)$ then P else Q; note that variable u is not bound within process Q, but only within process P. The sets containing the free variables appearing in a network N and in a process P are denoted by $fv(N)$ and $fv(P)$, respectively. A process definition is correct only if the inclusion $fv(P_{id}) \subseteq \{u_1, \ldots, u_{m_{id}}\}$ holds. A process P in which all variables $u \in fv(P)$ are instantiated with the value v is denoted by $\{v/u, \ldots\}P$; avoiding clashes done by performing alpha-conversion on some names appearing in P might be needed.

The mobility of processes is achieved by using the migration capability $go^{\Delta t}l$ then P that marks the fact that process P cannot execute at the current location for t time units, but will be able to execute at location l after the timeout expires. In order to have a flexible migration the name l can be a variable; it is replaced by a location value received from another processes by local communication. This ensures that a process does not have a predefined migration pattern, but is able to adapt it based on the interactions performed with other processes. Besides migration and communication actions, the processes can also contain the parallel composition operator $\mid$ and the terminated process 0. A process can communicate or migrate only if it resides inside a location l, and thus a network N is built from such located processes denoted by $l[[P]]$. If $fv(N) = \emptyset$, then the network is said to be well-formed.

Operational Semantics. The structural equivalence $\equiv$ is needed to rearrange a network such that its located processes can migrate or communicate by means of the operational semantic rules given in Table 3. The $\equiv$ relation is the smallest congruence that also holds for the equalities from Table 2. The equalities from Table 2 ensure that for any system N there exist an equivalent one $l_1[[P_1]] \mid \ldots \mid l_n[[P_n]]$, in which the processes P_i cannot be decomposed into parallel processes; this means that for each P_i there is no Q_i and R_i such that $P_i \equiv Q_i \mid R_i$.

Table 3 contains the operational semantics rules of rTiMo . A labelled transition $N \xrightarrow{\Lambda} N'$ shows how a network N is transformed into a network N' by applying a multiset of actions Λ. If $\Lambda = \{\lambda\}$ we simply write $N \xrightarrow{\lambda} N'$ to indicate that $N \xrightarrow{\{\lambda\}} N'$. On the other hand, a labelled transition $N \overset{t}{\rightsquigarrow} N'$ shows how a network N is transformed into a network N' by allowing t time units to pass.

Table 2. rTiMo Structural Congruence

$$N \mid l[[0]] \equiv N$$
$$N \mid N' \equiv N' \mid N$$
$$(N \mid N') \mid N'' \equiv N \mid (N' \mid N'')$$
$$l[[P \mid Q]] \equiv l[[P]] \mid l[[Q]]$$

The rule (Move0) models how a process $go^{\Delta 0} l'$ then P changes its current location from l to l' in order to allow process P to execute. The rule (Com) models how two processes $a^{\Delta t}!\langle v\rangle$ then P else Q and $a^{\Delta t}?(u)$ then P' else Q' residing at the same location l communicate on the shared channel a; a successful communication means that the second process will instantiate its variable u with the received value v. There is no change of location for any of the processes during the communication, and the two processes continue their execution at the current location l as P and $\{v/u\}P'$, respectively. The rules (Put0) and (Get0) model how the processes $a^{\Delta 0}!\langle v\rangle$ then P else Q and $a^{\Delta 0}?(u)$ then P else Q choose to remove the output and input action, respectively, and continue as Q at the current location.

Rule (Call) models the unfolding of a recursive process $id(u)$ by replacing it with the process definition of id in which the parameters u are instantiated with the values v. The use of the structural congruence $\equiv$ is achieved by using the rules (Equiv) and (DEquiv). The behaviour of a network of parallel located process is modelled by rule (Par) that collects the actions applied on the components of the network.

The passage of time is modelled by using the rules with names starting with D. The notation $N_1 \mid N_2 \not\xrightarrow{\lambda}$ used in rule (DPar) marks that there exists no action λ and no network $N_1' \mid N_2'$ such that $N_1 \mid N_2 \xrightarrow{\lambda} N_1' N_2'$ by applying one of the rules: (Move0), (Com), (Put0), (Get0) or (Call). The timeout t appearing in rule (DPar) marks the fact that for at least t time units none of the rules (Move0),

Table 3. rTiMo Operational Semantics

(DStop)	$l[[0]] \stackrel{t}{\rightsquigarrow} l[[0]]$
(DMove)	if $t \geq t'$ then $l[[go^{\Delta t} l' \text{ then } P]] \stackrel{t'}{\rightsquigarrow} l[[go^{\Delta t-t'} l' \text{ then } P]]$
(Move0)	$l[[go^{\Delta 0} l' \text{ then } P]] \xrightarrow{l \rhd l'} l'[[P]]$
(Com)	$\dfrac{t > 0 \text{ and } t' > 0}{l[[a^{\Delta t}!\langle v\rangle \text{ then } P \text{ else } Q \mid a^{\Delta t'}?(u) \text{ then } P' \text{ else } Q']] \xrightarrow{\{v/u\}@l} l[[P \mid \{v/u\}P']]}$
(DPut)	$\dfrac{t \geq t' > 0}{l[[a^{\Delta t}!\langle v\rangle \text{ then } P \text{ else } Q]] \stackrel{t'}{\rightsquigarrow} l[[a^{\Delta t-t'}!\langle v\rangle \text{ then } P \text{ else } Q]]}$
(Put0)	$l[[a^{\Delta 0}!\langle v\rangle \text{ then } P \text{ else } Q]] \xrightarrow{a!^{\Delta 0}@l} l[[Q]]$
(DGet)	$\dfrac{t \geq t' > 0}{l[[a^{\Delta t}?(u) \text{ then } P \text{ else } Q]] \stackrel{t'}{\rightsquigarrow} l[[a^{\Delta t-t'}?(u) \text{ then } P \text{ else } Q]]}$
(Get0)	$l[[a^{\Delta 0}?(u) \text{ then } P \text{ else } Q]] \xrightarrow{a?^{\Delta 0}@l} l[[Q]]$
(DCall)	$\dfrac{l[[\{v/u\}P_{id}]] \stackrel{t}{\rightsquigarrow} l[[P'_{id}]] \text{ and } id(u) \stackrel{def}{=} P_{id}}{l[[id(v)]] \stackrel{t}{\rightsquigarrow} l[[P'_{id}]]}$
(Call)	$\dfrac{l[[\{v/u\}P_{id}]] \xrightarrow{id@l} l[[P'_{id}]] \text{ and } id(u) \stackrel{def}{=} P_{id}}{l[[id(v)]] \xrightarrow{id@l} l[[P'_{id}]]}$
(DPar)	$\dfrac{N_1 \stackrel{t}{\rightsquigarrow} N'_1\ ,\ N_2 \stackrel{t}{\rightsquigarrow} N'_2 \text{ and } N_1 \mid N_2 \not\xrightarrow{\lambda}}{N_1 \mid N_2 \stackrel{t}{\rightsquigarrow} N'_1 \mid N'_2}$
(Par)	$\dfrac{N_1 \xrightarrow{\Lambda_1} N'_1 \text{ and } N_2 \xrightarrow{\Lambda_2} N'_2}{N_1 \mid N_2 \xrightarrow{\Lambda_1 \cup \Lambda_2} N'_1 \mid N'_2}$
(DEquiv)	$\dfrac{N \equiv N', N' \stackrel{t}{\rightsquigarrow} N'' \text{ and } N'' \equiv N'''}{N \stackrel{t}{\rightsquigarrow} N'''}$
(Equiv)	$\dfrac{N \equiv N', N' \xrightarrow{\Lambda} N'' \text{ and } N'' \equiv N'''}{N \xrightarrow{\Lambda} N'''}$

(Com), (Put0), (Get0), and (Call) is applicable. The use of negative premises allows to separate the passage of time and the application of actions, while assuring that the set of rules remains consistent [7].

The fact that applying time rules does not lead to nondeterministic behaviours and does not skip time is illustrated by the next results.

Proposition 1. *For all networks N, N', and N'', it holds that:*

1. $N \overset{t}{\rightsquigarrow} N'$ *together with* $N \overset{t}{\rightsquigarrow} N''$ *imply that* $N' \equiv N''$;
2. $N \overset{(t+t')}{\rightsquigarrow} N'$ *iff exists* N'' *with* $N \overset{t}{\rightsquigarrow} N''$ *together with* $N'' \overset{t'}{\rightsquigarrow} N'$.

3 The rTiMo Logic

Table 4 presents the syntax of the logical formulas used in this section.

Table 4. Syntax of logical formulas

φ, ψ	$::=$	$\top$	(true)
	$\mid$	$\neg\varphi$	(negation)
	$\mid$	$\varphi \vee \psi$	(disjunction)
	$\mid$	$\forall x.\varphi$	(universal quantification over names)
	$\mid$	$\mathbf{0}$	(void)
	$\mid$	$\varphi \mid \psi$	(composition)
	$\mid$	$\varphi \rhd \psi$	(guarantee)
	$\mid$	$l[\varphi]$	(location)
	$\mid$	$\varphi@l$	(localisation)
	$\mid$	$\langle\lambda\rangle\varphi$	(existential quantification over actions)
	$\mid$	$\langle t\rangle\varphi$	(existential quantification over time)

The logic has the classical propositional connectives, $\top$, $\neg\varphi$, $\varphi \vee \psi$, together with a universal quantification on names $\forall x.\varphi$. Drawing inspiration from [6], the connectives $\mathbf{0}$, $l[\varphi]$, and $\varphi \mid \psi$, are the logical formulas used to model processes and networks, while the adjuncts of the formulas $l[\varphi]$ and $\varphi \mid \psi$ are the formulas $\varphi@l$ and $\varphi \rhd \psi$, respectively.

The logical formulas $\langle\lambda\rangle\varphi$ and $\langle t\rangle\varphi$ are the formulas corresponding to the existential quantification over action and time transitions [5].

The satisfaction relation $P \vDash \varphi$ means that the closed process P satisfies the closed formula φ, while the satisfaction relation $N \vDash \varphi$ marks the fact that the closed network N satisfies the closed formula φ. These relations are defined inductively in Definition 1.

Definition 1. *The satisfaction relations $P \vDash \varphi$ and $N \vDash \varphi$ are defined inductively as:*

$$
\begin{array}{lcl}
P \vDash \top & \stackrel{def}{=} & \textit{always true} \\
P \vDash \neg\varphi & \stackrel{def}{=} & \textit{not } P \vDash \varphi \\
P \vDash \varphi \vee \psi & \stackrel{def}{=} & P \vDash \varphi \textit{ or } P \vDash \psi \\
P \vDash \forall x.\varphi & \stackrel{def}{=} & \textit{for any name } n,\ P \vDash \varphi\{n/x\} \\
P \vDash \mathbf{0} & \stackrel{def}{=} & P \equiv \mathbf{0} \\
P \vDash \varphi \mid \psi & \stackrel{def}{=} & \exists P_1, P_2 \textit{ such that } P \equiv P_1 \mid P_2,\ P_1 \vDash \varphi \textit{ and } P_2 \vDash \psi \\
P \vDash \varphi \blacktriangleright \psi & \stackrel{def}{=} & \forall R,\ R \vDash \varphi \textit{ implies } P \mid R \vDash \psi \\
P \vDash \varphi@l & \stackrel{def}{=} & \exists N \textit{ such that } N \equiv l[[P]] \textit{ and } N \vDash \varphi \\
N \vDash \neg\varphi & \stackrel{def}{=} & \textit{not } N \vDash \varphi \\
N \vDash \varphi \vee \psi & \stackrel{def}{=} & N \vDash \varphi \textit{ or } N \vDash \psi \\
N \vDash \varphi \mid \psi & \stackrel{def}{=} & \exists N_1, N_2 \textit{ such that } N \equiv N_1 \mid N_2,\ N_1 \vDash \varphi \textit{ and } N_2 \vDash \psi \\
N \vDash \varphi \blacktriangleright \psi & \stackrel{def}{=} & \forall N',\ N' \vDash \varphi \textit{ implies } N \mid N' \vDash \psi \\
N \vDash l[\varphi] & \stackrel{def}{=} & \exists P \textit{ such that } N \equiv l[[P]] \textit{ and } P \vDash \varphi \\
N \vDash \langle\lambda\rangle\varphi & \stackrel{def}{=} & \exists N' \textit{ such that } N \xrightarrow{\lambda} N' \textit{ and } N' \vDash \varphi \\
N \vDash \langle t\rangle\varphi & \stackrel{def}{=} & \exists N' \textit{ such that } N \stackrel{t}{\rightsquigarrow} N' \textit{ and } N' \vDash \varphi
\end{array}
$$

By definition, the satisfaction relation given in Definition 1 is closed under the structural congruence:

Proposition 2. *$N \equiv N'$ and $N \vDash \varphi$ imply that $N' \vDash \varphi$.*

In what follows we list some connectives that can be derived from those presented in Table 4.

$$\begin{array}{lcl} \bot & \stackrel{def}{=} & \neg\top \\ \varphi \wedge \psi & \stackrel{def}{=} & \neg(\neg\varphi \vee \neg\psi) \\ \exists x.\varphi & \stackrel{def}{=} & \neg\forall x.\neg\varphi \\ \langle\lambda\rangle\varphi & \stackrel{def}{=} & \neg\langle\lambda\rangle\neg\varphi \\ \langle t\rangle\varphi & \stackrel{def}{=} & \neg\langle t\rangle\neg\varphi \\ \varphi \to \psi & \stackrel{def}{=} & \neg\varphi \vee \psi \end{array}$$

The following result considers a formula that is true for a network only if that network contains a single location.

Proposition 3. *There exist l and P such that $N \equiv l[[P]]$ iff*

$$N \vDash \exists x.x[\top].$$

The following result considers a formula that is true for a process that is able to execute a migration after some time units.

Proposition 4. *There exist l, P', and P'' such that $P \equiv go^{\Delta t} l'$ then P'', $P' \equiv go^{\Delta 0} l'$ then P'', $l[[P]] \overset{t}{\rightsquigarrow} l[[P']] \xrightarrow{\lambda} l'[[P'']]$ and $P' \vDash \varphi$ iff*

$$\begin{array}{ll} P \vDash & \exists y.(\langle t\rangle\langle y \rhd l'\rangle l'[\varphi])@y \\ & \wedge\neg\mathbf{0} \\ & \wedge\neg(\neg\mathbf{0} \mid \neg\mathbf{0}) \\ & \wedge\neg\exists x.x[\top] \end{array}$$

We define the logical equivalence between two networks.

Definition 2. *For two closed networks N and N', $N =_L N'$ denotes that for any closed formula φ*

$$N \vDash \varphi \textit{ holds iff } N' \vDash \varphi \textit{ holds.}$$

The bisimulation is a concept that was introduced independently in three different research areas [8]: computer science [9], [10]; set theory [11], [12]; and modal logic [13]. We previously used bisimulation to equate the timed behaviour of networks [2], [14]. In what follows, we consider the following bisimulation between networks:

Definition 3. *A binary relation $\mathcal{R}$ is called a* **strong timed bisimulation** *if $(N_1, N_2) \in \mathcal{R}$ implies that:*

- *if* $N_1 \xrightarrow{\lambda} N_1'$*, then* $\exists N_2'$*,* $N_2 \xrightarrow{\lambda} N_2'$ *and* $(N_1', N_2') \in \mathcal{R}$*;*

- *if* $N_1 \overset{t}{\rightsquigarrow} N_1'$*, then* $\exists N_2'$*,* $N_2 \overset{t}{\rightsquigarrow} N_2'$ *and* $(N_1', N_2') \in \mathcal{R}$*;*

- *if* $N_2 \xrightarrow{\lambda} N_2'$*, then* $\exists N_1'$*,* $N_1 \xrightarrow{\lambda} N_1'$ *and* $(N_1', N_2') \in \mathcal{R}$*;*

- *if* $N_2 \overset{t}{\rightsquigarrow} N_2'$*, then* $\exists N_1'$*,* $N_1 \overset{t}{\rightsquigarrow} N_1'$ *and* $(N_1', N_2') \in \mathcal{R}$*,*

where the above hold for all $\lambda \in \{id@l, \{v/u\}@l,\ go^{\Delta 0}@l,\ a?^{\Delta 0}@l,\ a!^{\Delta 0}@l\}$ *and* $t \in \mathbb{R}_+$*.*

Two networks N_1 *and* N_2 *are* **strongly timed bisimilar***, written* $N_1 \sim N_2$*, if there exists a strong timed bisimulation* $\mathcal{R}$ *such that* $(N_1, N_2) \in \mathcal{R}$*. Formally:*

$$\sim = \bigcup\{\mathcal{R} \mid \mathcal{R} \text{ is a strong timed bisimulation}\}.$$

Having these two relations between networks it holds that:

Theorem 1. $N_1 =_L N_2$ *iff* $N_1 \sim N_2$*.*

4 Conclusion

TiMo (Timed Mobility) was initially defined in [15]; over the years several variants were defined: PerTiMo [16] that considers access permissions, rTiMo [1] that considers real-time timeouts, knowTiMo [17] that considers an explicit knowledge for each agent. Also the combination between the bigraphs [18] and TiMo was considered in [19], where the BigTiMo calculus was defined and studied.

This paper considers the real-time extension for the process calculus TiMo . In order to be able to reason about specific spatial and timing requirements of real-time systems described in rTiMo , in this paper, we introduced an expressive logic that includes spatial and time operators. We presented several results regarding the expressiveness of our logic: we considered some logical formulas for capturing some primitives of the rTiMo calculus.

References

[1] B. Aman and G. Ciobanu, "Real-time migration properties of rTiMo verified in Uppaal," in *The 11th International Conference on Software Engineering and Formal Methods – SEFM 2013 Proc. LNCS, vol. 8137*, 2013, pp. 31–45.

[2] B. Aman and G. Ciobanu, "Verification of critical systems described in real-time TiMo," *International Journal on Software Tools for Technology Transfer*, vol. 19, no. 4, pp. 395–408, 2017.

[3] E. A. Emerson, "Temporal and modal logic," in *Handbook of Theoretical Computer Science, Volume B: Formal Models and Semantics*, 1990, pp. 995–1072.

[4] Z. Manna and A. Pnueli, *The temporal logic of reactive and concurrent systems - specification*, Springer, 1992.

[5] P. Bouyer, F. Cassez, and F. Laroussinie, "Timed modal logics for real-time systems – specification, verification and control," *Journal of Logic, Language and Information*, vol. 20, no. 2, pp. 169–203, 2011.

[6] L. Cardelli and A. D. Gordon, "Anytime, anywhere: Modal logics for mobile ambients," in *The 27th ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages – POPL 2000*, 2000, pp. 365–377.

[7] J. F. Groote, "Transition system specifications with negative premises," *Theoretical Computer Science*, vol. 118, no. 2, pp. 263–299, 1993.

[8] D. Sangiorgi, "On the origins of bisimulation and coinduction," *ACM Transactions on Programming Languages and Systems*, vol. 31, no. 4, pp. 15:1–15:41, 2009.

[9] A. Ginzburg, *Algebraic Theory of Automata*, 1st ed., Academic Press, 1998.

[10] R. Milner, "An algebraic definition of simulation between programs," in *The 2nd International Joint Conference on Artificial Intelligence*, 1971, pp. 481–489.

[11] M. Forti and F. Honsell, "Set theory with free construction principles," *Annali della Scuola Normale Superiore di Pisa, Classe di Scienze 4e série*, vol. 10, no. 3, pp. 493–522, 1983.

[12] R. Hinnion, "Extensional quotients of structures and applications to the study of the axiom of extensionality," *Bulletin de la Société Mathmatique de Belgique,* vol. XXXIII, no. II (B), pp. 173–206, 1981.

[13] J. van Benthem, *Modal Logic and Classical Logic*, Bibliopolis, 1985.

[14] B. Aman, G. Ciobanu, and M. Koutny, "Behavioural equivalences over migrating processes with timers," in *The International Conference on Formal Techniques for Distributed Systems – Joint 14th IFIP WG 6.1 FMOODS 2012 and 32nd IFIP WG 6.1 FORTE 2012 Proc. LNCS, vol.7273*, 2012, pp. 52–66.

[15] G. Ciobanu and M. Koutny, "Modelling and verification of timed interaction and migration," in *The 11th International Conference on Fundamental Approaches to Software Engineering - FASE 2008 Proc. LNCS, vol. 4961*, 2008, pp. 215–229.

[16] G. Ciobanu and M. Koutny, "Timed migration and interaction with access permissions," in *The 17th International Symposium on Formal Methods – FM 2011 Proc. LNCS, vol. 6664*, 2011, pp. 293–307.

[17] B. Aman and G. Ciobanu, "Knowledge dynamics and behavioural equivalences in multi-agent systems," *Mathematics*, vol. 9, no. 22, Article No 2869, 26 p., 2021.

[18] R. Milner, *The Space and Motion of Communicating Agents*, Cambridge University Press, 2009.

[19] W. Xie, H. Zhu, M. Zhang, G. Lu, and Y. Fang, "Formalization and verification of mobile systems calculus using the rewriting engine Maude," in *The 2018 IEEE 42nd Annual Computer Software and Applications Conference – COMPSAC 2018, Proc. IEEE Computer Society, vol. 1*, 2018, pp. 213–218.

Bogdan Aman, Gabriel Ciobanu

Received June 13, 2022
Accepted July 08, 2022

Institute of Computer Science, Romanian Academy, Iaşi, Romania
E–mail: {bogdan.aman,gabriel.ciobanu}@iit.academiaromana-is.ro

Punctilog Compared to Dependency Grammar and Constituency Grammar

Victoria Bobicev, Tudor Bumbu, Victor Didic,
Dumitru Prijilevschi, Gheorghe Morari

Abstract

Punctilog, yet another model of sentence structure representation, was invented with the intention to represent the meaning encoded in the natural language text. In this paper, we compared punctilog markup with two syntactic grammars most used in computational linguistics: constituency and dependency grammar. Both grammars have been used in computational linguistics for many years; there are numerous annotated resources and tools working with these formalisms. Our study analyses similarities and differences between the grammars and the new model in order to use existing resources and tools for punctilog markup.

Keywords: punctilog, sentence structure, dependency grammar, constituency grammar, computational linguistics.

MSC 2020: 03B65, 68T50.

1 Introduction

Meaning is the most essential aspect of natural language text but also the most difficult to formalize for automatic processing. Various formalisms have been invented in order to capture the meaning of the natural language text; however, this problem still has no optimal solution.

Grammars, widely used in the domain of computational linguistics partially fulfill the need of formal sentence representation but still do not represent the meaning directly. In this paper, we discuss a novel methodology of sentence meaning formalization called punctilog [3]. It

©2023 by V. Bobicev, T. Bumbu, V. Didic, D. Prijilevschi, G. Morari

is using words of the sentence as units and special symbols for their relationship representation.

Sentence structure created by punctilog is partially similar to the structures created by other notations, namely grammars, developed in the domain of computational linguistics. In this work, we compare three types of sentence structures: created by constituency grammar, created by dependency grammar, and created by puctilog. Our final goal is an algorithm of transformation from one formalism to another.

The paper is organized in the following sections: Section 2 introduces some related work; the proposed formalism named punctilog is described in Section 3; Sections 4 and 5 contain a short overview of constituency and dependency grammars, respectively; Section 6 is dedicated to the comparison of the above described formalisms; Section 7 describes our effort to transform described grammar structures into punctilog formalism; the paper concludes with discussion and future work.

2 Related Work

The problem of the logical representation of sentence meaning is one of the central problems in computational linguistics. Various approaches have been developed over years for the sentence meaning representation [7] (chapter 15). The differences between these approaches were substantial but all of them represented meaning as a structure composed of a set of symbols from a vocabulary. Mostly, the symbols were natural language words that constituted the initial sentence. Structures, on the other hand, were dependent considerably on the approach. One of the first and most widely used was First-Order Logic [5]. String representation of the sentence meaning was one of the advantages; the others were: a small set of instruments (predicates, logical connectors, quantifiers) and comparatively simple rules of representation which stipulated that meaning representation consisted of objects, properties of objects, and relations among objects. However, a small set of instruments was also a disadvantage as it had difficulties presenting meanings of various natural language sentences.

Another well known method was Abstract Meaning Representation

[1] that formed rooted, labeled graphs from English sentences based on notions of arguments and relations. The syntactic structure of the sentence was transformed significantly and it was impossible to restore the original structure of the meaning representation.

Punctilog is a more straightforward representation that keeps word order in the sentence and marks relations among these words by introducing special punctuation marks.

3 Punctilog

The punctuation symbolism for semantic markup called PML (punctuation markup language) was introduced in [3] with the ultimate intention to disambiguate natural language texts. Initially, it worked only within a sentence. The markup is based on the compositionality principle stating that the meaning of an expression is a function of meanings of its constituents. In our case the constituents are words that form the sentence.

The formalism used punctuation marks for the semantic markup. Round brackets were used to indicate constituent structure [2]: connection of two elements as constituents is a binary operation and it is called association operation. The colon mark indicated a "modifier": an element that modified another element in a sentence. The angular brackets indicate that the enclosed expression should be treated as idiomatic or "multi-word expression" which means that it should not be treated compositionally but rather perceive all this combination of words as one concept. The formal definition of the model is described in [3]. Some details we skip in this paper as less significant for our comparison. An example of full punctilog annotation is presented in Figure 1, and the tree structure of this sentence is presented in Figure 2.

The authors of [12] investigated the ways of structural disambiguation experimenting with automate punctuation markup language annotation. The results were promising but there still were problems that had yet to be investigated.

In [4], the PML was renamed to punctilog as the abbreviation PML had been used for Prague Markup Language before punctilog had been introduced. I.Drugus et al. [4] mostly concentrated on association op-

"Swimming!" called Ion Freckle, but his team, Gicu and Mihai, was up to it already.

([Swimming!] :((called: (Ion ::Freckle))), (((his: <team>): <Gicu, Mihai>) :(((was :<up to>): it) :already))

Figure 1. An example of annotation using all punctilog punctuation marks from [3]

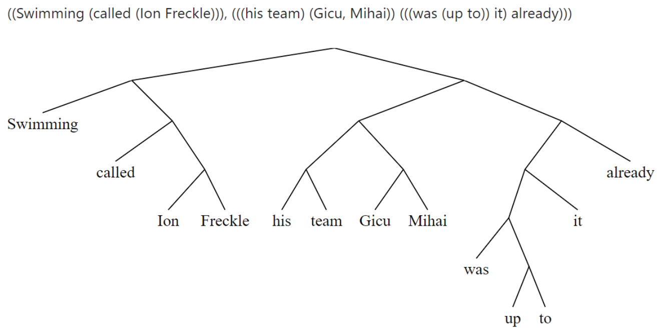


Figure 2. A tree structure based on association operations for the sentence presented in Figure 1

eration and the algorithms developed for the transformation of dependency relations in association operations. In this paper, we analyze the similarities and differences between associations, dependencies and constituent structures.

4 Constituency Grammar

The most widely used in computational linguistics formal system for modeling constituent structure of English and other natural language sentences is the Context-Free Grammar. Syntactic constituency is the idea that groups of words can behave as single units [7] (chapter 12), or constituents. Thus, grouping words in the sequence, one can define meaning indicating their relationships.

Since linguists started to group words long before the computational grammars appeared, their denotations were adopted. In constituent grammars, there are multiple groups named "noun phrase",

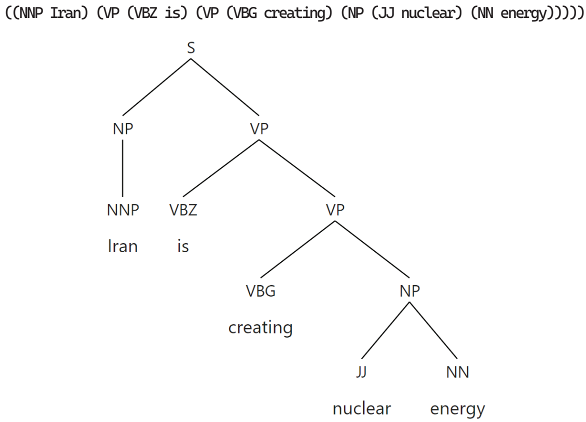


Figure 3. An example of constituency grammar annotation in a brackets and graphical form using neural constituency parser[1]

"verb phrase", "prepositional phrase", etc. Noun phrases include a noun and all words closely related to it as article, adjective, and others. The main element in a verb phrase is the main verb; auxiliary verbs, adverbs, and other closely related words are included in this group.

Structures formed by these grammars can be presented in a brackets form or in the form of a tree. Figure 3 presents an example of brackets representation of the constituent structure and graphical representation in a form of the tree for the sentence "Iran is creating nuclear energy". Notations: S is a sentence, NP is a noun phrase, VP is a verb phrase, other abbreviations denote parts of speech (NN – noun, JJ – adjective, etc.).

5 Dependency Grammar

Constituent grammars are widely used in computational linguistics; however, multiple problems appear while applying this formalism to languages with free order of words in a sentence. For such languages,

[1]https://parser.kitaev.io

dependency grammar was a much better solution. One of the first implementations of the idea of word dependencies was link grammar [11]. In this formalism, the syntactic structure of a sentence is presented as directed labeled grammatical relations between the words [7] (chapter 14). Relations among the words are labeled edges directed from the head word to its dependents. One word is considered a root word, it is not dependent on any other word in a sentence and, by convention, it is the main predicate of the sentence. Thus, as there is one root and other words, which are dependent on the root or other words, it is possible to represent the dependency structure of a sentence in the form of a tree as in both above described conventions (see Figure 4).

One of the largely known project dedicated dependency grammar annotations is Universal Dependencies [9].

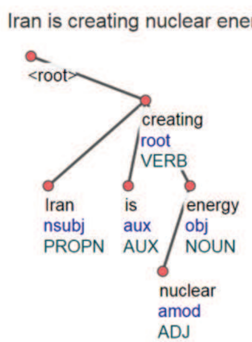


Figure 4. An example of dependency grammar annotation graphical representation in a form of the tree[2]

6 Comparison of the Formalisms

All three described formalisms perceive a text written in natural language as a set of finite strings (sentences) composed as sequences of the symbols (words) and form their structures adding specific markup

[2]https://universaldependencies.org/conllu_viewer.html

to these sequences taking into consideration the words' specific properties (morphological, syntactic and semantic). As it is seen from the example in Section 4, constituency structure may be introduced by adding parentheses into the initial sentence. This annotation is quite similar to the punctilog association operation and might be straightforwardly used for punctilog markup. The only difficulty is that present constituents are formed from more than two elements, whilst punctilog allows only strictly binary associations. From this point of view, dependency relations seem to be more suitable as they are strictly between two elements: head and dependent word.

In this work, we represent all three structures in a form of tree for better visibility. Tree representation of the sentence structure is a widely acceptable convention; the most well-known grammars create tree-form structures of sentences that encode syntactic and semantic relations among the words; though, their tree structures are quite different. There are works that demonstrate the possibility to transform one structure into another [6] but details on both sides can make the transformation complicated. In our work, we compare constituent structures produced by Berkeley Neural Parser [8], dependency structures of Universal Dependencies parser Stanza [10], and Punctilog conventions. Table 1 summarizes the similarities and differences between tree structures of these formalisms.

Transformation from one convention to another can be quite useful considering multiple linguistic resources annotated in various formalisms. Having a reliable algorithm of transformation makes it possible to obtain large volumes of text annotated in a convention we need.

This work analyses the transformation from constituency and dependency grammars to punctilog markup.

Comparing the trees we see that constituency and punctilog structures are more similar than punctilog and dependency trees. In dependency structures, all nodes are words, whilst in constituency grammar and punctilog, words are only terminal nodes. However, the main difficulty in transformation of grammar structure in punctilog notation presents multiple ramifications from one head node to several child nodes while punctilog notation accepts only binary ramifications (any parent node can be connected to exactly two child nodes).

Table 1. Comparison between constituency grammar, dependency grammar and punctilog graphical representations in forms of tree

Property	constituency grammar	dependency grammar	punctilog
form of the graph	tree with several ramifications and one root	tree with several ramifications and one root	tree with several ramifications and one root
graph nodes	terminal nodes are words, non-terminal nodes are syntactic groups: constituents, NP, VP, PP, etc.	all nodes are words of the sentence	terminal nodes are words, non-terminal nodes are not named in any way
root node	Sentence Node: S	main predicate of the sentence	just a node where the last two edges meet
graph edges	are not labeled, not directed	are directed and labeled	are not labeled, not directed
ramification	several nodes can be connected to one head node	several nodes can be connected to one head node	only two nodes can be connected to one head node

7 Transformation Problems

There are multiple tools and resources for constituency and for dependency grammars: parsers and already annotated corpora; thus we consider that the best way of similar resources creation for punctilog is the reuse of already existing tools. We developed deterministic algorithms of constituency and dependency structures transformation into punctilog [4]. The algorithms transform the constituency and dependency structures connections into association operations of punctilog.

Figures 5 and 6 present the initial constituent structure of the sentence "In many ways, the prerequisites differ from one Member State to another." and the punctilog structure transformed from the constituency tree. The same operation for dependency structure of the same sentence is presented in Figures 7 and 8. While Figures 5 and 7 are naturally very different, it is interesting how different are Figures 6 and 8 which code the same sentence in the same punctilog convention.

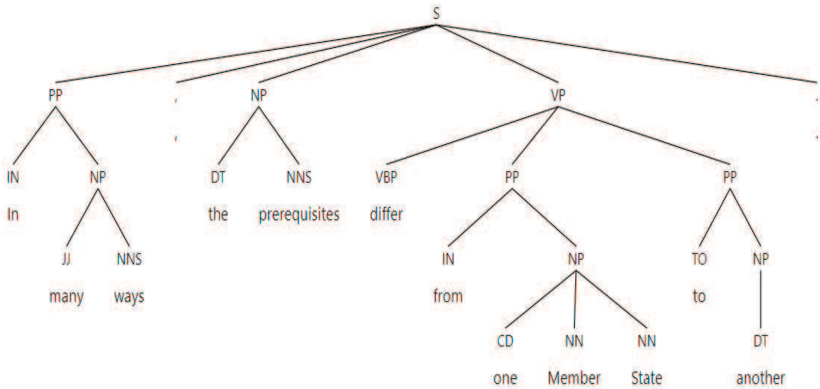


Figure 5. Syntactic tree of the sentence formed by constituency parser

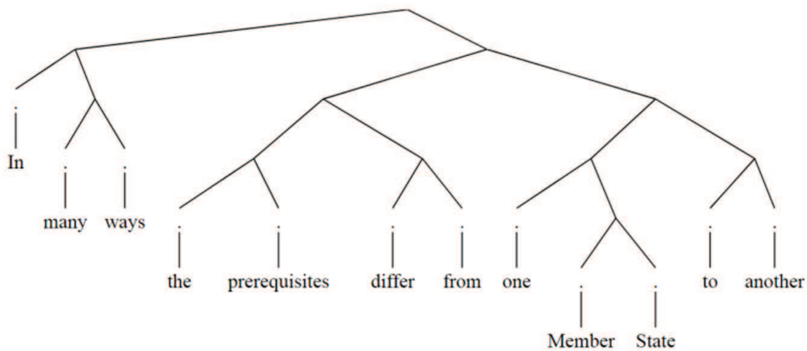


Figure 6. Syntactic tree of the sentence for punctilog convention transformed from constituency structure

The main problem in this transformation is the interpretation of

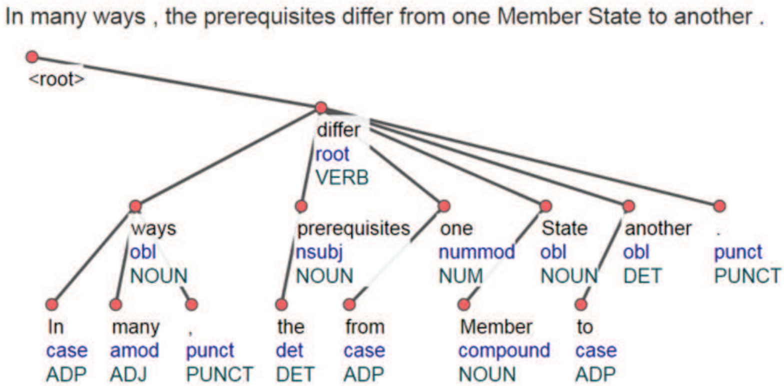


Figure 7. Syntactic tree formed by dependency parser

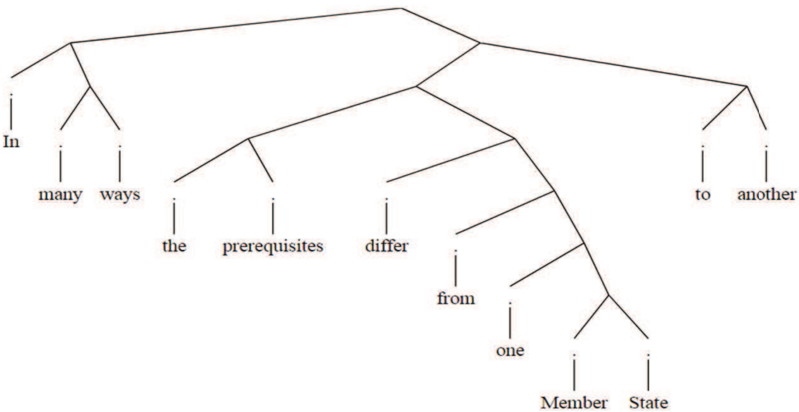


Figure 8. Syntactic tree for punctilog convention transformed from the dependency structure

multiple connections which we have to convert into a sequence of binary connections. In the example in Figures 5 and 6, structures are comparatively similar in terms of connections but in Figure 7, almost all words are connected to the root. The transformation algorithms were described in [4], and they analyzed the connections and decided which types of dependencies to connect first. However, in some cases the decision was difficult; for example, if there were multiple preposition groups connected in a sequence one after another. In the above example, there are two prepositional groups "from one Member State" and "to another" that can be connected in various ways. Figure 9 demonstrates an alternative sequence of connections for the part "the prerequisites differ from one Member State to another" that contains two prepositional phrases in a sequence.

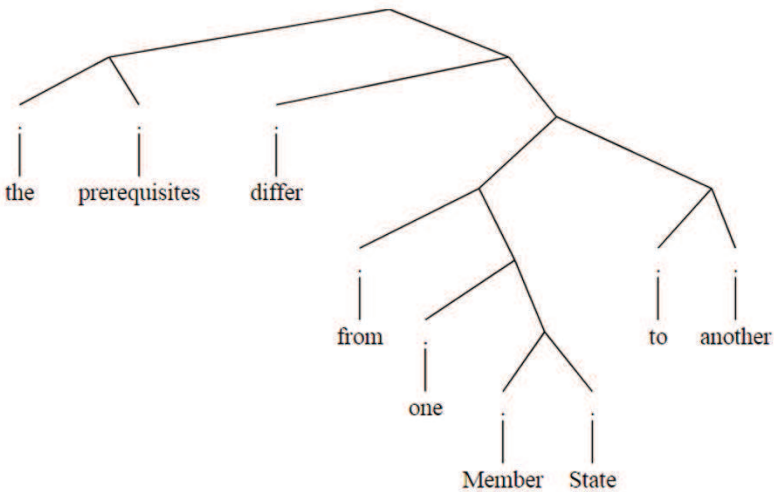


Figure 9. An alternative structure of connections while transforming from the dependency structure into punctilog convention

Figure 10 demonstrates one more difficult example of the sentence "First, we must educate people on how to protect themselves better online.", where multiple words are connected to one head.

The obtained after the transformation punctilog structure is pre-

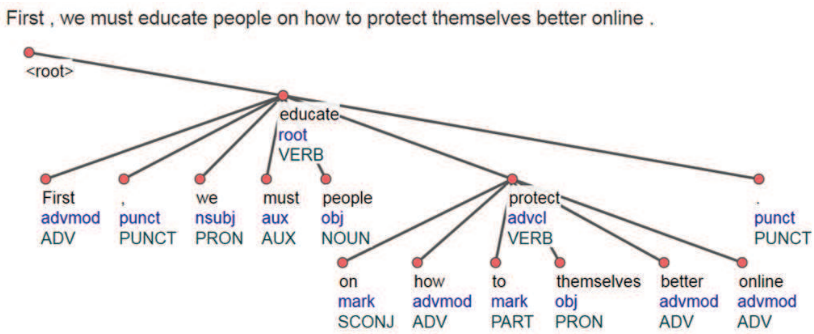


Figure 10. Dependency structure of the sentence “First, we must educate people on how to protect themselves better online”

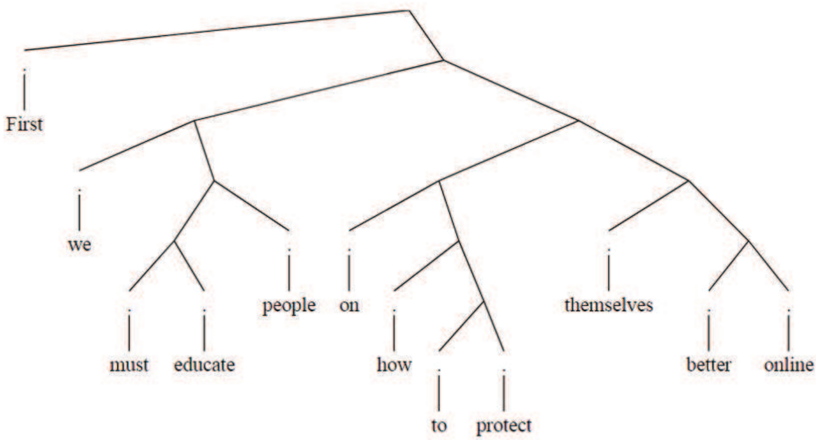


Figure 11. Punctilog structure obtained from dependency structure from Figure 10

sented in Figure 11. Most of the sentence has an adequate structure; the problematic one is the last part “to protect themselves better online”. Here, the last three words are connected semantically to the head verb “protect” but binary associations connect “better” to “online” before the connection to the verb.

8 Discussion and Conclusion

In this paper, we compare several representations of sentence structure, namely constituency and dependency structures with a new method of sentence meaning representation called punctilog. We studied similarities and differences between these two grammars and the new model in order to use existing resources and tools for punctilog markup. We developed rule based deterministic algorithms of transformation from the constituency and dependency structures to punctilog markup.

The main problem of the algorithms is the interpretation of multiple connections to one parent node that have to be converted into a sequence of binary connections. In such cases, several indicators are taken into consideration. First of all, the distance between words; the closest words are connected first. Next, morphological information is used in the algorithm’s rules. The words that are considered closer from morpho-syntactic point of view are also connected before other connections. For example, adjectives, and articles are connected to the closest noun; adverbs are connected to the verb or adjective next to which they are.

The connection order problem becomes more difficult while connecting large parts of long sentences. Long distance semantic relations present a difficult task that has no optimal solution yet.

Our future plans include development of the algorithm for transformation of multiple connections into binary and transformation of syntactic treebanks into punctilog annotated corpus.

References

[1] J. Bos, “Expressive Power of Abstract Meaning Representations,” *Computational Linguistics*, vol. 42, no. 3, 2016.

[2] A. Carnie, *Constituent Structure*, Oxford University Press, 2nd ed., 2010, 320 p.

[3] I. Drugus, "PML: A Punctuation Symbolism for Semantic Markup," in *The 11th International Conf. "Linguistic Resources and Tools for Processing the Romanian Language" Proc.*, 2015, pp. 79–92.

[4] I. Drugus, T. Bumbu, V. Bobicev, V. Didic, A. Burduja, A. Petrachi, and V. Alexei, "Punctilog: a New Method of Sentence Structure Representation," in *Proceedings of the Workshop on Intelligent Information Systems WIIS2021*, 2021, pp. 118–129.

[5] J. P. E. Hodgson, *First Order Logic*, Saint Joseph's University, Philadelphia, 1995.

[6] R. Johansson and P. Nugues, "Extended constituent-to-dependency conversion for English," in *Proceedings of the 16th Nordic Conference of Computational Linguistics (NODALIDA 2007)*, 2007.

[7] Dan Jurafsky and James H. Martin, *Speech and Language Processing*, Prentice Hall Series in Artificial Intelligence, 2008, 971 p.

[8] N. Kitaev, S. Cao, and D. Klein, "Multilingual Constituency Parsing with Self-Attention and Pre-Training," in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 2019, pp. 3499–3505.

[9] Marie-Catherine de Marneffe, Christopher Manning, Joakim Nivre, and Daniel Zeman, "Universal Dependencies," *Computational Linguistics*, vol. 47, no. 2, pp. 255–308, 2021.

[10] Peng Qi, Yuhao Zhang, Yuhui Zhang, Jason Bolton, and Christopher D. Manning, "Stanza: A Python Natural Language Processing Toolkit for Many Human Languages," in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, 2020. DOI: 10.18653/v1/2020.acl-demos.14.

[11] D. D. Sleator and D. Temperley, "Parsing English with a link grammar," in *Proceedings of the Third International Workshop on Parsing Technologies IWPT-93*, 1993, pp. 277–292.

[12] S. Stratulat, D. Prijilevschi, G. Morari, and T. Bumbu, "A Disambiguation Model for Natural Language Processing," in *Proceedings of Conference on Mathematical Foundations of Informatics MFOI-2020*, (Taras Shevchenko National University of Kyiv January 12-16, Kyiv, Ukraine), 2021, pp. 361–381.

Victoria Bobicev[1], Tudor Bumbu[2],
Victor Didic[3], Dumitru Prijilevschi[4],
Gheorghe Morari[5]

Received June 13, 2022
Accepted August 03, 2022

Victoria Bobicev
[1]Technical University of Moldova
E–mail: victoria.bobicev@ia.utm.md

Tudor Bumbu
[2]Vladimir Andrunachievici Institute of Mathematics and Computer Science
E–mail: tudor.bumbu@math.md

Victor Didic
[3]Technical University of Moldova
E–mail: didic.victor@iis.utm.md

Dumitru Prijilevschi
[4]Technical University of Moldova
E–mail: dumitru.prijilevschi@isa.utm.md

Gheorghe Morari
[5]Technical University of Moldova
E–mail: morari.gheorghe@isa.utm.md

Ocular Disease Recognition

Ioana Madalina Tugui, Adrian Iftene

Abstract

Ocular Diseases can cause irreversible vision loss. There are a few symptoms that appear as soon as needed, and a clear diagnostic made early can save someone's eye sight. That's why an automatic and precise algorithm for this task is a must in preventing something that the human eye can miss. The purpose of the project is to create a model that can identify an ocular disease based on fundus photographs.

Keywords: ocular disease, data augmentation, LeNet5, AlexNet.

1 Introduction

Artificial intelligence (AI) has begun to be used successfully in the medical field, increasing doctors' confidence in using automated systems to help them at various stages of their daily activities. Successful examples include both rapid imaging (X-Ray, CT) and predictions of disease progression or proposals for effective treatment. The applicability of AI is more and more diverse in this field lately [1], including the classification of teeth [2], the monitoring of patients [3], prediction in chronic kidney disease [4], [5], personalized therapies for cancer patients [6], stroke detection [7], [8], tuberculosis detection [9], [10], etc.

Fundus photographs are ocular documentation that record the appearance of a patient's retina[1]. These are also used to document abnormalities of disease process affecting the eye, and/or to follow up on the progress of the eye condition/disease such as diabetes or age-macular degeneration (AMD). The ocular disease recognition is a difficult task, as each pathology, especially in an early stage, when we have similar

©2023 by Ioana Madalina Tugui, Adrian Iftene

[1]https://en.wikipedia.org/wiki/Fundus_photography/

patterns. Even when there is a severe case, the diagnostic can't be observed because of small details.

In the majority of existing studies we found, ocular disease recognition using deep learning focuses on only one abnormality. In the paper [11], were used multiple architectures for different issues, and the obtained results are quite good. The authors used accessible architectures such as AlexNet [12] or other custom CNN's. Thus, the diseases such as diabetic retinopathy, cataract, glaucoma are well handled, with an accuracy better than 90% for each task. AlexNet was used successfully in other similar task like flower and fruit classification [15].

After Section 2 with the presentation of the Data set from Kaggle, which contains six main abnormalities, Section 3 comes, which is related to experiments with different architectures and the obtained results. The last section is related to conclusions and future work.

2 Data set

For this project, we will use the data set provided by Kaggle[2]. In the data set from Kaggle[3], we can find the next abnormalities: *Diabetes, Glaucoma, Cataract, Related Macular Degeneration, Myopia, Hypertensive Retinopathy* and another ones. In the next sections, we will investigate how these diseases manifest and more important, statistics made based on age and gender, that may help us later.

2.1 Diabetes

Diabetes is a chronic disease that occurs either when the pancreas does not produce enough insulin or when the body cannot effectively use the insulin it produces[4]. Diabetic retinopathy is an important cause of blindness, and occurs as a result of long-term accumulated damage to the small blood vessels in the retina. Diabetes is the cause of 2.6%

[2] https://www.kaggle.com/andrewmvd/ocular-disease-recognition-odir5k/
[3] https://www.kaggle.com/andrewmvd/ocular-disease-recognition-odir5k
[4] https://www.who.int/news-room/fact-sheets/detail/diabetes/

of global blindness[5]. As shown in Figure 1, it's caused by damage to the blood vessels of the light-sensitive tissue at the back of the eye (retina). According to World Health Organization, there are about 422 million people worldwide who have diabetes[6]. Also, according to Diabetes Research and Clinical Practice, in 2019, a total of 463 million people are estimated to be living with diabetes, representing 9.3% of the global adult population (20–79 years). The increase of diabetes prevalence with age leads to a prevalence of 19.9% (111.2 million) in people aged 65–79 years[7].

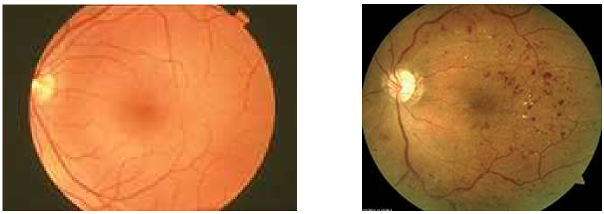


Figure 1. Normal Eye vs. Diabetic Eye

2.2 Glaucoma

Glaucoma is a group of eye conditions that damage the optic nerve, the health of which is vital for good vision. This damage is often caused by an abnormally high pressure in your eye. It is one of the leading causes of blindness for people over the age of 60. It can occur at any age but is more common in older adults. Having certain medical conditions, such as diabetes, heart disease, high blood pressure and sickle cell anemia are some of the risk factors[8]. Here are some fundus images that represent how it manifests (see Figure 2[9]). According to

[5] https://www.who.int/news-room/fact-sheets/detail/diabetes/

[6] https://www.who.int/news-room/fact-sheets/detail/diabetes/

[7] https://www.diabetesresearchclinicalpractice.com/article/S0168-8227(19)31230-6/fulltext/

[8] https://www.mayoclinic.org/diseases-conditions/glaucoma/symptoms-causes/syc-20372839

[9] https://www.researchgate.net/figure/Sample-fundus-images-of-normal-and-glaucoma-classes-from-our-private-database_fig2_321383942

World Health Organization, glaucoma is the second leading cause of blindness globally[10]. Some statistics presented by American Academy Of Ophthalmology are that the number of people (aged 40–80 years) with glaucoma worldwide will be increasing to 111.8 million in 2040[11].

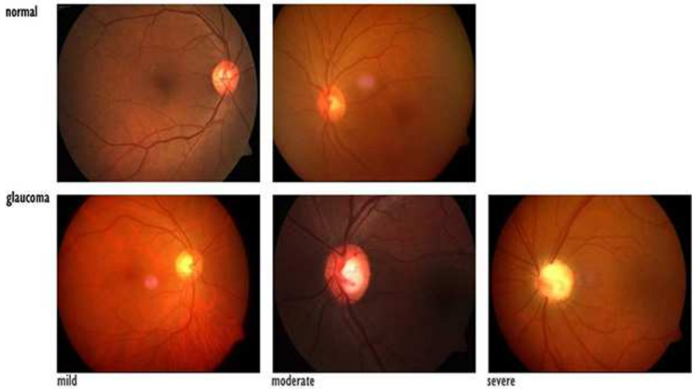


Figure 2. Normal eye versus Glaucoma eye

2.3 Cataract

A cataract is a clouding of the normally clear lens of your eye[12]. Most cataracts develop slowly and don't disturb your eyesight early on. But with time, cataracts will eventually interfere with your vision[13]. It is one of the most easy abnormalities one can recognize, because of the white film it creates. In Figure 3, we can observe how a life-like representation of cataract looks like on fundus images[14]. It is very easy to observe how cloudy the lens is.

[10]https://www.who.int/bulletin/volumes/82/11/feature1104/en/
[11]https://www.aaojournal.org/article/S0161-6420(14)00433-3/abstract
[12]https://www.healthline.com/health/cataract
[13]https://www.mayoclinic.org/diseases-conditions/cataracts/symptoms-causes/syc-20353790
[14]https://www.researchgate.net/figure/On-left-the-patients-left-eye-has-no-cataract-and-all-structures-are-visible-On-right_fig2_258811697

Figure 3. Normal Eye versus Cataract Eye

2.4 Related Macular Degeneration

Age-related macular degeneration (AMD) is an eye disease that can blur the sharp, central vision you need for activities like reading and driving. “Age-related” means that it often happens in older people. “Macular” means it affects a part of your eye called the macula[15]. In Figure 4, we have an explanation for how AMD looks like. It occurs when the delicate cells of the macula – the small, central part of the retina responsible for the center of our field of vision – become damaged and stop working.

Figure 4. Normal eye vs. amd

[15]https://www.nei.nih.gov/learn-about-eye-health/eye-conditions-and-diseases/age-related-macular-degeneration

2.5 Myopia

Myopia occurs when the eye grows too long from front to back. About 41.6 percent of Americans are nearsighted, up from 25 percent in 1971[16]. Figure 5 shows differences between a normal eye and a myopic eye. As you can see, the myopic eye (right) has a more stretched retina and thin blood vessels. Projections suggest that almost 50 percent of the world will be myopic by 2050[17].

Figure 5. Normal eye vs Myopic Eye

2.6 Hypertensive Retinopathy

Acute blood pressure elevation typically causes reversible vasoconstriction in retinal blood vessels[18]. You can observe that HR is not such a common disease. In Figure 6, we can observe how Hypertensive Retinopathy manifests. The Bhaktapur Retina Study is a population-based, cross-sectional study to estimate the prevalence of vitreoretinal diseases among subjects 60 years and above residing in the Bhaktapur district of Nepal. This study was done on 1000 people[19].

[16] https://www.nei.nih.gov/about/news-and-events/news/myopia-close-look-efforts-turn-back-growing-problem

[17] https://www.nei.nih.gov/about/news-and-events/news/myopia-close-look-efforts-turn-back-growing-problem

[18] https://www.msdmanuals.com/professional/eye-disorders/retinal-disorders/hypertensive-retinopathy

[19] https://www.researchgate.net/figure/Prevalence-of-retinal-disease-among-the-study-subjects-n-1000_fig1_288842588

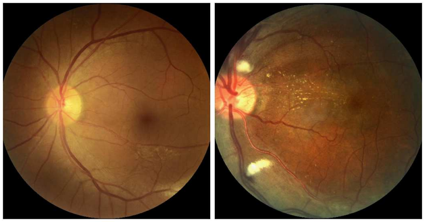


Figure 6. Normal eye vs Hypertensive Retinopathy

2.7 Statistics

Using Jupyer[20] and Pandas[21], we could create histograms that will help us identify the data that we are working with. We can observe in Figure 7 that the statistics shown before, are indeed, present in our data set.

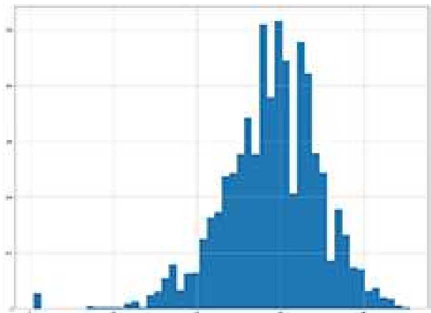


Figure 7. Histogram base on age

Somewhere around age of 40 to 75 people are more prognoses to develop an eye abnormality. Data is stored in a csv file. There are 6392 rows, each entity being described by an ID, patient age, patient sex, filename of the left eye and the right eye, and, for each photo, what diagnostic it has. After inspecting the number of photos of the left eye of each label we have, we can see that data is not well distributed. This

[20]https://jupyter.org/
[21]https://pandas.pydata.org/

caused problems with already existing good CNN architectures. The model was always overfitting.

2.8 Data Augmentation

This problem can be solved by augmentation applied on the data we are using. Our first approach was by using this ImageDataGenerator from keras_preprocessing.image[22]:

$data\_generator = ImageDataGenerator(rotation\_range = 40, shear\_range = 0.2, zoom\_range = 0.1)$

At the end, some images were misleading:

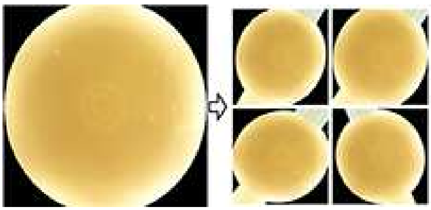


Figure 8. Raw photo vs. augmentation

It can be seen that images have some new surfaces. The problem starts with the *rotation_range* that was too high. This bug could have been caused also by the *shear_range*, but in this case, lowering the *rotation_range* was enough. The *zoom_range* also had some bad influence, as the white line was more visible when zooming out. By minimizing the *rotation_range* and *zoom_range*, we added a new argument, *brightness_range*. After refactoring the dataset, we started the augmentation process. By augmentation, we want to bring the number of images for each label to be closer to the number of maximum number of images from a label. In our case, we want to bring each label closer to the "Normal" one, with 1469 images. Our strategy is to set a certain constant for each diagnostic of images that needs to be created.

[22]https://keras.io/api/preprocessing/image/

3 Proposed Architecture

This project was written in Python 3.7.7, using frameworks and libraries such as TensorFlow[23], Keras[24], Scikit-Learn[25] and Pandas[26].

3.1 Multilayer Perceptron

On our first attempt, we tried using a multi-layer perceptron[27,28] with grey-scaled images. After trying many formats of images, we went for a scale of 70×70 pixels. The neural network was implemented from scratch. It has 4900 input nodes, 1500 hidden nodes, and 8 output nodes. We tried a learning rate of 0.5 with 100 epochs at first. We have implemented 2 activation functions: ReLU and Sigmoid. The model was not performing very well, as it was using gray images in a task where color was important in detecting the label of the image, and it was not using any filters. For this architecture, the code was inspired by the book [13].

3.2 CNN Architectures

3.2.1 LeNet5

LeNet5[29] CNN architecture is made up of 8 layers [14], which consist of 3 convolutional layers, 2 subsampling layers, 2 fully connected layers, and 1 flattened layer. All layers use tanh activation function, except one fully connected layer which uses softmax. After training the model, we can observe that it overfits.

After testing the model with a new dataset, the performance was about 73%. After scaling up the images to 100×100 pixels, the model needs more than 10 epochs for training.

[23] https://www.tensorflow.org/
[24] https://keras.io/
[25] https://scikit-learn.org/stable/
[26] https://pandas.pydata.org/
[27] "Make your own neural network" - Tarig Rashid
[28] https://www.goodreads.com/en/book/show/29746976
[29] https://www.datasciencecentral.com/profiles/blogs/lenet-5-a-classic-cnn-architecture

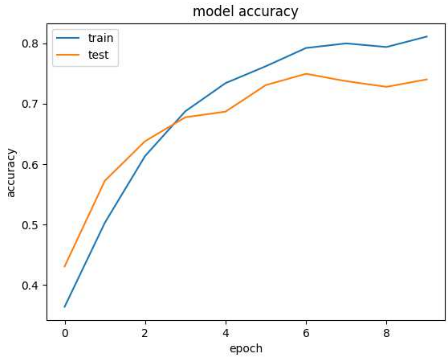


Figure 9. LeNet-5 Accuracy 70×70

Next, we tried 2 adapted models of the LeNet. In the first one, we added 2 dropouts, one of value 0.1 and one of value 0.5. We added more filters and made it more dense. The performance of this model on the new dataset was about 83% on both 70×70 and 100×100 image scales.

3.2.2 AlexNet

AlexNet[30] has 3 convolutional layers, 3 max-pooling layers, 2 normalization layers, 2 fully connected layers, and 1 softmax layer [14]. Each layer has convolutional filters and a nonlinear activation function, ReLU. For the AlexNet, we consider a CNN architecture with an input shape of (70,70,3). After 50 epochs, the history of accuracy for training looks like in Figure 10.

After testing the model with a total new dataset made up with 5225 images, the performance was of 82%. At some point, the test set has a really big drop in the testing phase. We changed the size of photos and made them to be scaled up to 100×100 and adapt a little more the model. After testing again the model on the dataset mentioned before, the performance was of 84%.

[30]https://www.mygreatlearning.com/blog/alexnet-the-first-cnn-to-win-image-net

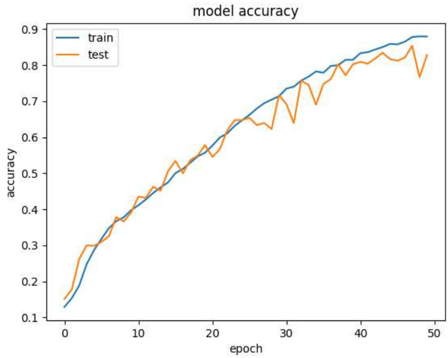


Figure 10. AlexNet Accuracy History 70×70

3.2.3 Custom models

For better results, we tried to combine these two models into one. The all obtained results are in Table 1.

4 Conclusion

In conclusion, for a task that works better when each diagnostic is treated isolated, an accuracy of 91-93% where all pathologies are put together is an improvement.

We tried testing each model on images found by us on the Internet. AlexNet performs better in recognition of myopia, cataract, and a normal eye. For the other diseases, there was not a positive answer. Glaucoma was one of the hardest to recognize. For it, in the final results, there was not even a small probability that it might actually be glaucoma. For a hypertensive eye, there was a small probability, but it was not the biggest found in the graphic.

The second custom model, which is made out of the first custom model with more filters added, performs similar to AlexNet, but with a small improvement. For the normal eye, the recognition is not as good, but in some cases it still predicts right. An improvement is brought on the diabetic eye, which now, on new images, is always predicted right.

Table 1. The results for the considered architectures

Results						
CNN Architecture	Acc-uracy	Valid. Accur.	Over-fitting	Timing (h:m)	Image size	Epochs
Orig.LeNet-5	73%	0	Yes	00:05	70×70	10
Orig.LeNet-5	15%	0	Yes	00:10	100×100	10
LeNet5Model2	86%	83%	No	02:03	70×70	50
LeNet5Model2	86%	82%	Yes	04:27	100×100	50
LeNet5Model3	84%	84%	Yes	02:09	70×70	50
LeNet5Model3	84%	83%	No	04:40	100×100	50
AlexNet	87%	82%	Yes	01:04	70×70	50
AlexNet	87%	84%	Yes	01:38	100×100	50
Combination1	90%	90%	No	08:45	70×70	45
Combination1	92%	92%	No	16:53	100×100	45
Combination2	90%	90%	No	11:37	70×70	45
Combination2	95%	93%	Yes	23:24	100×100	45

Acknowledgments. This work was supported by project REVERT (taRgeted thErapy for adVanced colorEctal canceR paTients), Grant Agreement number: 848098, H2020-SC1-BHC-2018-2020/H2020-SC1-2019-Two-Stage-RTD. Data processing and analysis in this paper were supported by the Competitiveness Operational Programme Romania, under project SMIS 124759 - RaaS-IS (Research as a Service Iasi).

References

[1] A. Burlacu, A. Iftene, E. Busoiu, D. Cogean, and A. Covic, “Challenging the supremacy of evidence-based medicine through artificial intelligence: the time has come for a change of paradigms,” *Nephrology Dialysis Transplantation*, vol. 35, no. 2, 7 November, 2019, pp. 191–194, https://doi.org/10.1093/ndt/gfz203.

[2] C.E. Iftinca, A. Iftene, and L.G. Coca, “Artificial Intelligence in Dentistry: Teeth Classification,” in *Proceedings of the Workshop*

on Intelligent Information Systems WIIS2021, (Chisinau, Republic of Moldova), October 14-15, 2021, pp. 143–155.

[3] A.E. Gistescu, C.M. Milut, T. Proca, and A. Iftene, "MedPlus - a cross-platform application that allows remote patient monitoring," in *25rd International Conference on Knowledge-Based and Intelligent Information & Engineering Systems, vol. 192*, (Szczecin, Poland), 8-10 September, 2021, pp. 3751–3760.

[4] A. Burlacu, A. Iftene, I.V. Popa, R. Crisan-Dabija, C. Brinza, and A. Covic, "Computational Models Used to Predict Cardiovascular Complications in Chronic Kidney Disease Patients: A Systematic Review," *Medicina (Kaunas)*, vol. 57, no. 6, Article Number 538, 2021, E-ISSN: 1648-9144. DOI: 10.3390/medicina57060538.

[5] A. Burlacu, A. Iftene, D. Jugrin, I.V. Popa, P.M. Lupu, C. Vlad, and A. Covic, "Using Artificial Intelligence Resources in Dialysis and Kidney Transplant Patients: A Literature Review," *BioMed Research International*, article ID 9867872, 2020, 14 pages. DOI: 10.1155/2020/9867872.

[6] A.J. Banegas-Luna, A. Pena-Garcia, A. Iftene, F. Guadagni, P. Ferroni, N. Scarpato, F.M. Zanzotto, A. Bueno-Crespo, and H. Perez-Sanchez, "Towards the Interpretability of Machine Learning Predictions for Medical Applications Targeting Personalised Therapies: A Cancer Case Survey," *International Journal of Molecular Sciences*, vol. 22, no. 9, Article Number 4394, 2021. E-ISSN: 1422-0067. DOI: 10.3390/ijms22094394.

[7] V. Pulbere and A. Iftene, "Using F.A.S.T. Test to Detect Stroke," in *Proceedings of the Conference on Mathematical Foundations of Informatics MFOI2020*, (Kyiv, Ukraine), January 12-16, 2021, pp. 346–360.

[8] M. Rezmerita, I. Cercel, and A. Iftene, "Stroke Detector - An Application that applies the F.A.S.T. Test to Identify Stroke," in *Proceedings of the 17th International Conference on Human-Computer Interaction RoCHI 2020*, 22-23 October, 2020, pp. 39–47.

[9] A. Hanganu, C. Simionescu, L.G. Coca, and A. Iftene, "UAIC2021: Lung Analysis for Tuberculosis Classification," in *Proceedings of the Working Notes of CLEF 2021 - Conference and*

Labs of the Evaluation Forum, vol. 2936, (Bucharest, Romania), September 21-24, 2021, pp. 1253–1263.

[10] L.G. Coca, A. Hanganu, C.G. Cușmuliuc, and A. Iftene, "UAIC2020: Lung Analysis for Tuberculosis Detection," in *Working Notes of CLEF 2020 - Conference and Labs of the Evaluation Forum, vol.2696*, (Thessaloniki, Greece), 22-25 September, 2020.

[11] Y. Elloumi, M. Akila, and H. Boudegga, "Ocular Diseases Diagnosis in Fundus Images using a Deep Learning: Approaches, tools and Performance evaluation," arXiv:1905.02544, 2019. [Online] Available: https://arxiv.org/ftp/arxiv/papers/1905/1905.02544.pdf.

[12] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks," *Communications of the ACM*, vol. 60, no. 6, pp. 84–90, 2017.

[13] T. Rashid, *Make Your Own Neural Network*, 1st ed., North Charleston, SC, USA: CreateSpace Independent Publishing Platform, 2016, 222 p. ISBN-10: 1530826608. ISBN-13: 978-1530826605.

[14] A. Géron, *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems,* 2nd. ed., O'Reilly Media, Inc., 2019, 856 p. ISBN-10: 1492032646. ISBN-13: 978-1492032649.

[15] M.D. Minut and A. Iftene, "Creating a Dataset and Models Based on Convolutional Neural Networks to Improve Fruit Classification," in *Proceedings of the 23rd International Symposium on Symbolic and Numeric Algorithms for Scientific Computing (SYNASC)*, (Timisoara, Romania), December 7-10, 2021.

Ioana Madalina Tugui[1,2], Adrian Iftene[1]

Received June 14, 2022
Accepted August 24, 2022

[1]"Alexandru Ioan Cuza" University of Iasi, Faculty of Computer Science
[2]Bytex Technologies, Iasi, Romania
E–mail: {ioana.tugui,adiftene}@info.uaic.ro

Using Machine Learning and Sound Processing Techniques to Improve Patient Health

Mihai-Andrei Costandache

Abstract

Health is one of the most important aspects of life. Medical experts have always made efforts to prevent, diagnose, and treat diseases. Diseases may be hard to identify and/or they may adapt rapidly, therefore, to stand a chance against them, medical experts have to use all kinds of machines/devices. Besides the specific medical equipment, there are devices with unrelated purposes (e.g., entertainment), that become genuine medical assistants by using various apps and websites. We propose using machine learning and sound processing techniques in the collaborative effort of medicine and computer science against diseases.

Keywords: machine learning, sound processing, diagnosis, treatment.

1 Introduction

We propose a research topic with two parts: *diagnosis* and *treatment*. In this way, we manage to consider the two issues independently, but at the same time, we can integrate them into a real context that follows the usual workflow (find out the patient's illness, then take action accordingly). Our intention is to develop a system with multiple features from each research direction.

The automated systems help the medics in the *diagnosis* by "reading" samples in various forms (e.g., sounds, images) and making predictions, that simply put are presence/absence of a disease. This approach resembles the way humans would perform the analysis. In this context of our research, machine learning and sound processing are mandatory.

©2023 by Mihai-Andrei Costandache

The data will be in audio form, including sounds from patient actions such as breathing and coughing.

From the *treatment* point of view, specialized systems help the medics and patients make the best decisions, based on the health evolution. There are multiple possible features: the medic receives a notification that the patient needs to be checked on immediately, the patient and medic get information about medication, the patient gets a reminder to take his/her medication or a warning that he/she is doing something against the treatment, etc. In this context, machine learning and sound processing are optional. The data (for machine learning purposes and not only) will be in text form, including various reports about the patient or medication.

We need to make the distinction between machine learning systems that require human-annotated datasets and those that are self-supervised. It is sometimes hard to label data manually, mostly due to large amounts of information; therefore, self-supervised methods are welcome. Another important issue regarding datasets is the patient confidentiality, health information is difficult to obtain, even if the data is de-identified in most cases anyway. The possible diseases that our system will approach are various, we consider the following categories: *respiratory*, *heart*, *sleep*, and *mental/speech* problems.

The rest of the paper is organized as follows: Section 2 presents the existing work in the context of the paper, focusing on the projects employing machine learning and/or sound processing. Section 3 describes the two parts of the system we will create, diagnosis and treatment. We are aware of the features we need to provide and what properties (mainly in a computer systems sense) our system should have. Section 4 shows our experience in the context of the paper, focusing on sound processing. Section 5 presents with whom we can collaborate (medical experts and computer science colleagues) and contains a plan of the work we intend to do, before drawing the conclusion in the last section.

2 Existing Work

2.1 Diagnosis

There are various medical tasks and existing approaches to them. In the context of our research, we are interested in disease detection techniques that employ sound processing and machine learning. Besides getting familiar with prediction methods, by studying the papers we get a baseline to compare our system with. There are also several datasets available for researchers to use, and if the datasets are associated with papers that describe machine learning models, we can have a fair comparison between our work and other researchers' work, by processing the same datasets. Obtaining medical data is difficult, due to privacy reasons, therefore finding a way to collect data properly from health institutions or anonymous potential patients online, or searching for existing datasets is a challenge in itself.

We have collected relevant papers and/or datasets for the disease categories we are interested in. Each of them has listed the sounds that are used:

- Respiratory [1]–[5] – breathing, coughing, sneezing, phonation of sustained vowels, numbers;
- Heart [6]–[8] – S1 ("lub") and S2 ("dub") sounds;
- Sleep [8], [9] – breathing;
- Mental/Speech [8], [10] – phonation of sustained vowels, numbers, words, short sentences.

2.2 Treatment

The most important types [11], [12], [14] of medical treatment (or just monitoring) products are:

- Tracker – It monitors the number of steps someone makes, how much sleep he/she gets, how much water he/she drinks, his/her diet, etc. and makes recommendations accordingly (sometimes including achievements in order to better establish a daily routine).

- Questionnaire – It gets information by "asking" the patient directly, this solution may require less effort to develop.
- Medic-patient remote communication – It provides medic-patient video/audio calls and text chats. When the medic and the patient are far from each other, or when social distancing is applied during pandemics, remote communication is greatly used.

There are several recommendations [11], [14] regarding such products, that are listed below. Security and compliance with specific regulations [14] can be considered mandatory.

- Patient-centric UX & UI;
- Low battery and memory consumption;
- Offline mode;
- Location awareness.

We have found several examples of medical treatment/monitoring products in [11]–[16]. Given that they are generally designed for mobile devices, they can be obtained and used easily by anyone who is familiar with smartphones.

3 Proposed System

3.1 Diagnosis

This part of our project is represented by machine learning techniques that state the presence/absence of a disease, from audio data. We identify four main properties this component should have, each of them is explained in detail in the following paragraphs:

- Correctness;
- Speed;
- Transparency;
- Extensibility.

The outcome of the system should be reliable, but given there are no perfect systems, we need to carefully analyze metrics such as precision and recall in the training process and search for a balance (possibly adjusted to the context). In the following, "positive" represents the presence of a disease, while "negative" represents its absence. Obviously, both a false positive, "scaring" a person into thinking he/she is ill and a false negative, saying that an ill patient is healthy, are dangerous.

Medical decisions should be made fast sometimes, therefore our system will return a result in a prompt fashion. The problem here is the size of the data. For one patient, we will most likely have more than just a recording, as the patient needs to be observed for a long enough period. The training process is definitely affected by the large datasets, but it is performed only occasionally, when we add/update models, as the models are serialized. The main concern should be the speed of the prediction process. It will be performed at each user request, therefore bottlenecks at this level are unacceptable.

The stakeholders will get enough details in order to understand and trust the outcome that is determined automatically. The system will output the probability of the presence of a disease and what were the main clues that led to a certain conclusion (e.g., *the audio data X recorded at time Y is suspicious*), and so on. A prediction must have an explanation in order to be valuable for two reasons: the medic and the patient need to trust the outcome, and the patient has the right to know his/her status in detail.

We will decide upon the initial diseases taken into consideration by the system after we will discuss with our partners. Besides these ones, "new" diseases should easily be added as the system is already available to the user, without modifying our system too much. Only some low-level data reading procedures and models should be altered.

The machine learning techniques we will use the most will be neural networks and random forests.

3.2 Treatment

Our system will obtain health condition information by "calling" the methods implemented in the diagnosis part (and not only), provide logs, notify the stakeholders, offer the patient treatment guidance, and facilitate the medic-patient communication. The notifications will mainly have different degrees: basic, warning, alert. This part of our project should be configurable to some extent. For example, an elder is more vulnerable and requires stricter observation, therefore the medic should get data about him/her more frequently. However, the patients' priority must not be modified too much by the medic, and the patient must not be able to "mute" some of the notifications about his/her health. The treatment guidance will offer the patient the chance to be more informed about his/her medication in terms of side effects, other users' experience, etc., and will offer him/her medication suggestions. Obviously, the patient must still stay in touch with the medic and must not take any important decision on his/her own, as it could put his/her health at risk. Our system is a health assistant, not a substitute for the medic.

4 Sound Processing

4.1 Sound Preprocessing

In [17], we created a machine learning system for identifying bird species from audio recordings, in the context of Conference and Labs of the Evaluation Forum (CLEF), LifeCLEF Lab, Task 2 - BirdCLEF. The identification technique was based on neural networks and mel-spectrograms. The spectrogram [18] is a visual representation of audio data, with time on the X-axis, frequency on the Y-axis, amplitude coded by color, and the Mel scale [19] is a scale of pitches. Before creating and using the machine learning models, we had to work on the preprocessing. Fig. 1 presents the main preprocessing approach, while Fig. 2 and Fig. 3 show two examples of mel-spectrograms.

We will adapt the preprocessing methods to our project. Most of the modules/tools will be used not only for the preprocessing step, prior to the model training/prediction, but also for processing the audio data in the context of sound de-identification (presented in the next subsection). Mainly, we intend to use the same modules, with their roles as follows:

- LibROSA [20] - audio file input functions, mel-spectrogram data generation, various processing features;
- FFmpeg [21] - support for several audio codecs;
- Matplotlib [22] - mel-spectrogram image I/O;
- NumPy [23] - numerical computing and n-dimensional arrays.

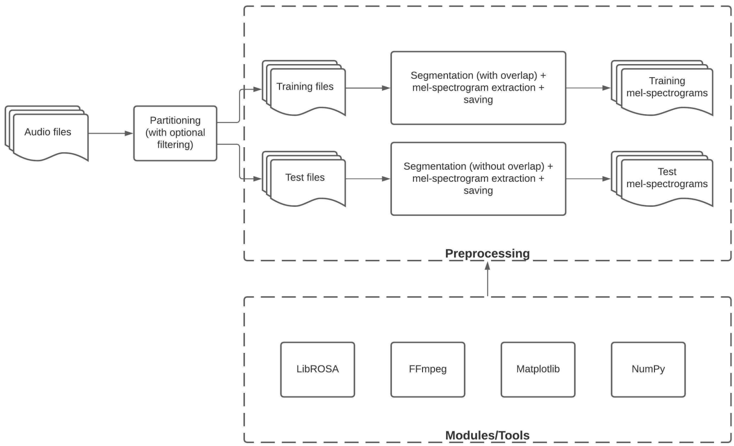


Figure 1. Preprocessing (adapted from [17]).

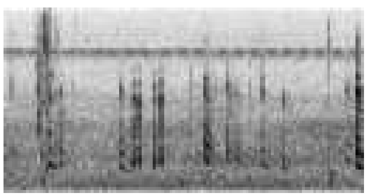


Figure 2. Spectrogram example 1 [17].

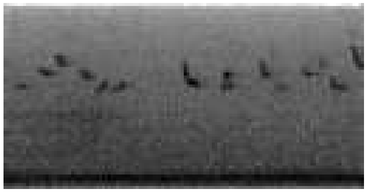


Figure 3. Spectrogram example 2 [17].

4.2 Sound De-identification

The privacy of the patients providing audio samples (either for model training or for prediction) is essential. In [24], we approached speaker de-identification, that is the task of hiding the identities of the people talking in a given context. We altered the audio data, by changing the voice pitch and adding noise, therefore the characteristics used in speaker recognition were greatly diminished or removed. The most important components of the system were a sound processing component and an API. Besides the de-identification functionality, there was also the reconstruction feature, modifying the voice back. Fig. 4 shows the architecture of the main components. SoundFile [25] was used for audio file output, Flask [26] was required for building the API.

We will adapt the system we created to our current project, as now we are not working only with spoken words (e.g., breathing sounds).

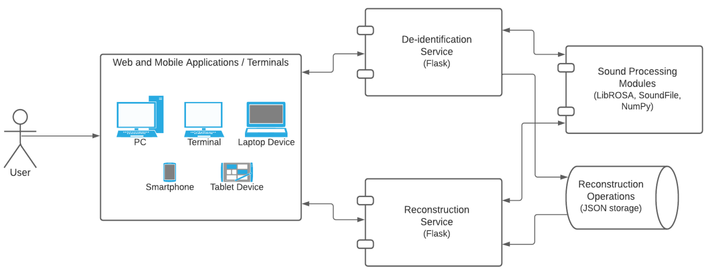


Figure 4. Sound processing component and API architecture [24].

5 Next Work

5.1 Partners

The nature of our project requires us to collaborate with medical experts, but we would like to work with other computer scientists as well. Our faculty has been involved in medical projects with the following institutions/groups:

- "Grigore T. Popa" University of Medicine and Pharmacy;
- IMAGO-MOL Cluster (North East Regional Innovative Cluster for Structural and Molecular Imaging).

We consider that the expertise of the people from both fields of study, medicine and computer science, will be a great addition to our project.

5.2 Objectives

We created a plan for our project that covers the main steps one should perform when doing research and proposing his/her own solution to a particular problem. Mainly, one should learn about the existing approaches, and then he/she should bring a valuable contribution to the topic (i.e., offering new approaches and/or improving the existing ones).

The plan is as follows:

1. Discussion with our partners about the requirements;
2. Survey(s) on relevant works;
3. Proposal of a system (e.g., architecture, features) and its refinement according to our partners' ideas;
4. Design and implementation of several algorithms based on sound processing and machine learning;
5. Implementation of the overall system.

6 Conclusion

The collaboration between experts from medicine and computer science leads to life-saving products that help prevent, diagnose, and treat diseases. We propose the research topic called "Using Machine Learning and Sound Processing Techniques to Improve Patient Health", with the goal of developing a system capable of diagnosis and treatment. We intend to work closely with medical experts and people at our faculty.

Acknowledgments. The PhD Program of the Faculty of Computer Science, UAIC Iași, Iasi, Romania, supported part of the research for this paper. Data processing and analysis in this paper were supported by the Competitiveness Operational Program Romania, under project SMIS 124759 - RaaS-IS (Research as a Service Iasi).

References

[1] L. Orlandic, T. Teijeiro, and D. Atienza, "The COUGHVID crowdsourcing dataset: A corpus for the study of large-scale cough analysis algorithms," *Scientific Data Journal*, no. 8, Article 156, 2021, DOI:10.1038/s41597-021-00937-4. [Online]. Available: https://paperswithcode.com/dataset/coughvid.

[2] A. Muguli, L. Pinto, R. Nirmala, N. Sharma, P. Krishnan, P.K. Ghosh, R. Kumar, S. Bhat, S.R. Chetupalli, S. Ganapathy, S. Ramoji, and V. Nanda, "DiCOVA Challenge: Dataset, task, and baseline system for COVID-19 diagnosis using acoustics," - derived from Project Coswara, of almost the same authors, 2021. [Online]. Available: https://paperswithcode.com/dataset/dicova.

[3] Kaggle (vbookshelf), "Respiratory Sound Database," 2019. [Online]. Available: https://www.kaggle.com/vbookshelf/respiratory-sound-database.

[4] OSF, "Dataset of sounds of symptoms associated with respiratory sickness," 2018. [Online]. Available: https://osf.io/tmkud/wiki/home/.

[5] Open Data Stack Exchange, "Data request. Voice audio dataset labeled with influenza-like illness or not," 2020. [Online]. Available: https://opendata.stackexchange.com/questions/18071/voice-audio-dataset-labeled-with-influenza-like-illness-or-not.

[6] Kaggle (kinguistics), "Heartbeat Sounds," 2016. [Online]. Available: https://www.kaggle.com/kinguistics/heartbeat-sounds.

[7] P. Bentley, G. Nordehn, M. Coimbra, and S. Mannor, "The PASCAL Classifying Heart Sounds Challenge 2011 (CHSC2011) Results," 2012. [Online]. Available: http://www.peterjbentley.com/heartchallenge/.

[8] PhysioNet, "Databases," [Online]. Available: https://www.physionet.org/about/database/.

[9] G. Korompili, A. Amfilochiou, L. Kokkalas, S.A. Mitilineos, N.A. Tatlas, M. Kouvaras, E. Kastanakis, C. Maniou, and S.M. Potirakis, "PSG-Audio, a scored polysomnography dataset with simultaneous audio recordings for sleep apnea studies," *Scientific Data Journal*, no. 8, Article 197, 2021, DOI:10.1038/s41597-021-00977-w. [Online]. Available: https://www.nature.com/articles/s41597-021-00977-w.

[10] GitHub (beamandrew), "Medical Data for Machine Learning," 2020. https://github.com/beamandrew/medical-data.

[11] Care Innovations, "What is Telehealth? What is Remote Patient Monitoring? How are They Different?," [Online]. Available: https://news.careinnovations.com/blog/what-is-telehealth-what-is-remote-patient-monitoring-how-are-they-different.

[12] Insider Intelligence, "Remote Patient Monitoring Trends & Health Devices in 2021," [Online]. Available: https://www.insiderintelligence.com/insights/remote-patient-monitoring-industry-explained/.

[13] Capterra, "Best Remote Patient Monitoring Software 2021 | Reviews of the Most Popular Tools and Systems," [Online]. Available: https://www.capterra.com/remote-patient-monitoring-software/.

[14] Topflight Apps, "Remote Patient Monitoring App Development Guide: Best Practices & Tips," [Online]. Available: https://topflightapps.com/ideas/how-to-develop-a-remote-patient-monitoring-app/.

[15] TaiDoc, "Health App – TaiDoc provides apps to self-monitoring/analysis to your health," [Online]. Available: https://taidoc.com/health-app/.

[16] CareCloud, "7 Best FDA Approved Health Apps – Continuum," [Online]. Available: https://www.carecloud.com/continuum/7-best-fda-approved-health-apps/.

[17] M.-A. Costandache, "LifeCLEF Lab, Task 2-BirdCLEF," in *The 10th International Conference and Labs of the Evaluation Forum (CLEF)*, (University of Lugano, Switzerland), 09-12 September, 2019.

[18] PNSN, "What is a spectrogram?," [Online]. Available: https://pnsn.org/spectrograms/what-is-a-spectrogram.

[19] S.S. Stevens, J. Volkmann, and E.B. Newman, "A scale for the measurement of the psychological magnitude pitch," *Journal of the Acoustical Society of America*, vol. 8, no 3, pp. 185–190, 1937, DOI:10.1121/1.1915893.

[20] B. McFee, V. Lostanlen, A. Metsai, M. McVicar, et al., "librosa/librosa: 0.8.0," DOI:10.5281/zenodo.3955228. [Online]. Available: https://zenodo.org/record/3955228#.YLfL9YXitPY.

[21] "FFmpeg," [Online]. Available: https://www.ffmpeg.org/.

[22] "Matplotlib," [Online]. Available: https://matplotlib.org/.

[23] C. R. Harris, K. J. Millman, S. J. van der Walt, et al., "Array programming with NumPy," *Nature Journal*, vol. 585, pp. 357–362, 2020, DOI:0.1038/s41586-020-2649-2. https://www.nature.com/articles/s41586-020-2649-2.

[24] M.-A. Costandache, A. Iftene, and D. Gîfu, "A Speaker De-identification System based on Sound Processing," in *Proceedings of the 29th International Conference on Information Systems Development (ISD)*, (Polytechnic University of Valencia, Spain), 8-10 September, 2021. https://aisel.aisnet.org/isd2014/proceedings2021/security/1/

[25] "SoundFile," [Online]. Available: https://pypi.org/project/SoundFile/.

[26] "Flask," [Online]. Available: https://flask.palletsprojects.com/en/1.1.x/.

Mihai-Andrei Costandache[1,2]

Received June 14, 2022
Accepted September 1, 2022

[1]Faculty of Computer Science, UAIC Iași, Iasi, Romania
E–mail: mihai.costandache@info.uaic.ro

[2]Codefy Software, Iasi, Romania

Uniqueness Violations

Paola Cattabriga

Abstract

This article presents a development of the author's investigation in self-reference procedures, considered from the point of view of logical inquiry regarding mathematical definitions and rules for applying substitutions. As already shown, this offers an explicative guide about how to show that Russell's antinomy does not really affect Frege's system, and, similarly, that the so-called Cantor theorem, affirming the nondenumerability of the set of all the subsets of the set of natural numbers, is not actually acceptable in an axiomatic framework of first-order set theory [3], [5], [7], [8]. We briefly resume both the demonstrations, displaying that they point out to the same sort of uniqueness violation, namely the neglect of a restriction that derives from the Criterion of Non-creativity. This article applies then the same explicative analysis to the so-called "Diagonalization Lemma", core of the whole Gödel's 1931 incompleteness argument, drawing accordingly for the first time clear evidence and explanation why it results to be a uniqueness violation in Peano Arithmetic.

Keywords: logic in informatics, theoretical aspects of automated reasoning, diagonalization, self-reference procedures, theory of definition.

1 Introduction

Results here exposed rise from experiences in theorem proving, declarative programming and computability extended in depth by the author to a quest about the reasons behind self-referring. These investigations necessarily required historical researches about the so-called "crisis of the foundations" of mathematics, a well-known phase of rethinking started with the discovery of paradoxes, mainly Russell's antinomy

©2023 by Paola Cattabriga

in 1902. Cantor's diagonalization argument for reaching more-than-denumerable sets and Russell's antinomy are logically connected and for the historical perspective the latter rises from the former. We are going to show that they are also connected by the same form of uniqueness violation as highlighted by Extensionality. The restriction on uniqueness is a condition established by the rules for definitions. With the *Theory of Definition* we refer to the field of logical researches of ancient tradition, which in the last century has seen noteworthy contributions from J.C.C. Mckinsey, A. Tarsky, E. W. Beth, A. Robinson, W. Craig. It is also worth mentioning that the modern theory of definition was at its beginning with Frege and Peano. But, our research, even if proposing a completely new and different *way out* from Russell's antinomy, recognizes the value of the contributions by S. Lešniewski and A. Padoa, for their inquires about the role of new definitions within the whole complexity of a logical system [11] [13]. All the well-known sorts of self-reference are always based on some introduction of a new set or predicate symbol into the system. For example, Russell's antinomy *defines* the set of those sets not belonging to themselves, and Cantor's theorem supposes the existence of a one-to-one correspondence between a set and the set of its parts and *defines* a set of those elements not belonging to such correspondence. It is therefore advantageous, from the explanatory point of view, to consider the requirements these definitions must satisfy to preserve the consistency of the system itself. The point of introducing a new symbol is to facilitate deductive investigation from the structure of the theory, but not to add to that structure. The rules for definitions, established by the criterions of eliminability and non-creativity, state the conditions for proper equivalence, giving some basic restrictions to prevent superimpositions and circularity. By the Criterion of Eliminability, a defined symbol should always be eliminable from any formula of the theory, and by the Criterion of Non-Creativity, a new definition does not permit the proof of relationships among the old symbols which were previously unprovable, namely it does not function as a creative axiom. By the rule for defining a *new operation symbol* (or a new individual constant, i.e., an operation symbol of rank zero), an equivalence like

$$O(x_1 \dots x_n) = y \leftrightarrow \psi$$

introducing a new operation symbol O is a proper definition only if the formula $\exists! y\psi$ is derivable from the axioms and preceding definitions of the theory. If the restriction on the uniqueness is dropped, then a contradiction can be derived [16, pp 151-173] [14] [5], [7], [8].

2 Uniqueness by Extensionality

In Zermelo-Fraenkel set theory, *ZF*, when defining a set, the uniqueness for the comprehension axiom schema is ensured by the axiom of extensionality. The self-referring characteristic of Russell's antinomy turns out to be accordingly a procedure that forces a set to be twice defined: we are dealing with a set whose elements belong and do not belong to the set itself, dropping thus the above uniqueness condition. Le us briefly resume how Russell's antinomy argument does not hold in a first order system based on both *Comprehension* and *Extensionality*, because it violates the uniqueness embodied by the latter [5].

$$\textit{Comprehension} \quad \forall z_1 ... \forall z_n \exists y \forall x (x \in y \leftrightarrow \varphi(x)),$$

where $\varphi(x)$ is any formula in *ZF*, $z_1, ..., z_n$ are the free variables of $\varphi(x)$ other than x, and y is not a free variable of $\varphi(x)$, and

$$\textit{Extensionality} \quad \forall x \forall y [\forall z (z \in x \leftrightarrow z \in y) \leftrightarrow x = y].$$

Extensionality, without any additional axioms, implies that for every condition $\varphi(x)$ on x (in *Comprehension*) there exists "at most" one set y that contains *exactly* those elements x that fulfill the condition $\varphi(x)$, in other words, if there is a set y such that $\forall x(x \in y \leftrightarrow \varphi(x))$, y is unique. It can be shown as follows. If y' is also such, i.e., $\forall x(x \in y' \leftrightarrow \varphi(x))$, then we have, obviously $\forall x(x \in y' \leftrightarrow x \in y)$, and then by *Extensionality*, $y' = y$.

Usually Russell's antinomy argumentation is presented as follows.

RA *There exists no set which contains exactly those elements which do not contain themselves, in symbols* $\neg\exists y \forall x(x \in y \leftrightarrow x \notin x)$.

Proof. By contradiction. Assume that y is a set such that for every element x, $x \in y$ if and only if $x \notin x$. For $x = y$, we have

$y \in y$ if and only if $y \notin y$. Since, obviously, $y \in y$ or $y \notin y$, and as we saw, each of $y \in y$ and $y \notin y$ implies the other statement, we have both $y \in y$ and $y \notin y$, which is a contradiction.

This proof holds for sure as a first order theorem, and it can affect comprehension principle when it is regarded individually or in itself, but it does not hold in any system which applies both *Comprehension* and *Extensionality*. Indeed, we can show that Russell's antinomy does not affect a first-order set theory, since, by *Extensionality*, the above proof RA cannot be accomplished. If, as an example of *Comprehension*, we define

$$x \in y \text{ if and only if } x \notin x,$$

then, by *Extensionality*, we obtain always

$$(x \in y \leftrightarrow x \notin x) \rightarrow \neg(x = y), \tag{$*$}$$

so it is never the case that $x = y$. We can consider this as an applicative case of the uniqueness condition. Indeed, a special case, because it involves negation and, therefore, complementation. For it, the above inference RA can not be accomplished since "$x = y$" can not be assumed and "$y \in y$ if and only if $y \notin y$" is not derived. In other terms, for a simple first-order rule from

$$\forall x(x \in a \leftrightarrow x \notin x)$$

we can yield

$$(a \in a \leftrightarrow a \notin a),$$

but this rule is not applicable within a reasoning in a context of Set theory, where, by *Extensionality*, the formula

$$(x \in a \leftrightarrow x \notin x) \rightarrow \neg(x = a) \tag{$**$}$$

can always be derived, it always holds. Whenever $x \in a \leftrightarrow x \notin x$, then $(**)$.

In accordance, Russell's antinomy argument does not hold in a first order system based on both *Comprehension* and *Extensionality*, since it violates the restriction on the uniqueness established by extensionality.

In defining a set containing exactly those elements which do not contain themselves we define a set which members can not be considered to be identical to the set itself, either we should have a set defined twice, as a set which contains and does not contain itself. This would be to define a set and its complement as being exactly the same identical set, and in detail to violate the criterion of non-creativity and its consequent criterion of relative consistency [16, p 155].

From the logical point of view, Russell's antinomy turns out to be just a partial argument. Although set theory disposes of the symbols to mention, or to express, the existence of sets containing exactly those elements which do not contain themselves, yet, when the argument reaches its own whole representation, no contradiction can be derived. All that turns out to be amazing especially in Frege's Grundgesetze [5], [8].

Likewise, Cantor's diagonalization turns out to be a violation of the restriction on uniqueness too [3]. For the so-called Cantor theorem, each set in ZF has cardinality properly less than its power set. The argument supposes by absurd the existence of a one-to-one correspondence between a set and the set of its parts, $\mathsf{g} : A \mapsto P(A)$ and reaches contradiction, defining a set B of those elements of the set not belonging to g, $B = \{x \in A | x \notin \mathsf{g}(x)\}$. B is a subset of A and g is a surjection, so *there must exist some* $b \in A$ *such that* $B = \mathsf{g}(b)$ (*diagonalization*), and (as for each $x \in A$) either $b \in B$ or $b \notin B$. As it is well-known, both cases should lead to contradiction [3]. Except that the definition of the relative complement,

$$\overline{B} = \{x \in A | x \notin B\},$$

is always possible in ZF by the Subsets Axiom Schema, and that leads to

$$B = \mathsf{g}(b) \Leftrightarrow \overline{B} \neq \mathsf{g}(b).$$

Hence, thanks to the Axiom of Extensionality,

$$(b \in \overline{B} \iff b \in \mathsf{g}(b)) \Longrightarrow \overline{B} = \mathsf{g}(b),$$

and consequently

$$B \neq \mathsf{g}(b).$$

In accordance, diagonalization is false and cannot be inferred within Cantor's argument. Its proof by contradiction on the two cases $b \in B$ and $b \notin B$ is not to be concluded. It is also fairly certain that Cantorian diagonalization is an argument out of the context of an axiomatic framework [3].

3 Uniqueness violation in Gödel's 1931

The development of Gödel's 1931 argument by the so-called "Diagonalization Lemma" enable us to display within Gödel's incompleteness how to trace down anew a uniqueness violation. The central idea of the Diagonalization Lemma like a general self-referential statement is credited by Gödel to Carnap [1].

> It is even possible, for any methamathematical property f which can be expressed in the system, to construct a proposition which says of itself that it has this property. For suppose that $F(z_n)$ means that n is the number of a formula that has the property f. Then, if $F(S(w,w))$ has the number p, $F(S(z_p, z_p))$ says that it itself has the property f. [10, pp 362-363]

As it is well-known, Gödel numbering is the number theoretical means by which self-referring propositions are permitted in Peano Arithmetics, *PA*. This method of associating numbers with symbols, expressions and sequences of expressions was originally devised by Gödel in order to *aritmethize* metamathematics [9, pp 163-171]. In the following, for any detail on Gödel numbering the reader is referred to [12, pp 190-197].

We recall the Diagonalization Lemma in a short elegant version due to [15, p 6].

Let us define the function of substitution $sb(\ulcorner \phi(v) \urcorner, \overline{n}) = \ulcorner \phi(\overline{n}) \urcorner$, which gives us the Gödel number of the result of replacing v by the n-th numeral in $\phi(v)$ (see the corresponding $Sb(x^v_y)$ and $Sb[x^a_{\chi(y)}]$ in Gödel 1931). Let $\phi(v)$ be given and let us call $\beta(v)$ the formula $\phi(sb(v,v))$.

Let $m = \ulcorner\beta(v)\urcorner$ and $\delta = \beta(\overline{m})$. We shall show that δ is the sentence we were looking for. To this purpose, we notice that in *PA* they hold the following equivalences

$$\begin{aligned}
\vdash \delta &\iff \beta(\overline{m}) && \text{by definition} && (1)\\
&\iff \phi(sb(\overline{m},\overline{m})) && \text{by definition} && (2)\\
&\iff \phi(sb(\overline{\ulcorner\beta(v)\urcorner},\overline{m})) && \text{since } m = \ulcorner\beta(v)\urcorner && (3)\\
&\iff \phi(\overline{\ulcorner\beta(\overline{m})\urcorner}) && \text{definition of } sb && (4)\\
&\iff \phi(\overline{\ulcorner\delta\urcorner}) && \text{by definition.} && (5)
\end{aligned}$$

Diagonalization Lemma For any formula ϕ with only the variable v free, there is a sentence δ such that

$$\vdash_{PA} \delta \iff \phi(\overline{\ulcorner\delta\urcorner}).$$

As it is well known, Gödel's 1931 argument begins here taking $\phi(v)$ as $\forall x\,\neg Pf(x,v)$ and obtaining

$$\vdash_{PA} \delta \iff \forall x\,\neg Pf(x,\overline{\ulcorner\delta\urcorner}).$$

The reader can refer to the definition of the concepts involved to [12, pp 203-207]. Gödel's incompleteness argumentation states: (a) if *PA* is consistent, not $\vdash_{PA} \delta$, (b) if *PA* is ω-consistent, not $\vdash_{PA} \neg\delta$. Thence, if *PA* is ω-consistent, δ is an undecidable sentence of *PA*. The argument is as follows. Let q be the Gödel number of δ.

(a) Assume $\vdash_{PA} \delta$. Let r be the Gödel number of a proof in *PA* of δ. Then $\mathtt{Pf}(r,q)$. Hence, $\vdash_{PA} Pf(\overline{r},\overline{q})$, that is $\vdash_{PA} Pf(\overline{r},\overline{\ulcorner\delta\urcorner})$. We already have $\vdash_{PA} \delta \iff \forall x\,\neg Pf(x,\overline{\ulcorner\delta\urcorner})$. By Biconditional Elimination, $\vdash_{PA} \forall x\,\neg Pf(x,\overline{\ulcorner\delta\urcorner})$. By Rule A4 (Particularization Rule), $\vdash_{PA} \neg Pf(\overline{r},\overline{\ulcorner\delta\urcorner})$. Therefore, *PA* is inconsistent.

(b) Assume *PA* is ω-consistent and $\vdash_{PA} \neg\delta$. Since $\vdash_{PA} \delta \iff \forall x\,\neg Pf(x,\overline{\ulcorner\delta\urcorner})$, by Biconditional Elimination, we obtain $\vdash_{PA} \neg\forall x\,\neg Pf(x,\overline{\ulcorner\delta\urcorner})$ which abbreviates to $(*)$ $\vdash_{PA} \exists x\,Pf(x,\overline{\ulcorner\delta\urcorner})$.

On the other hand, since *PA* is ω-consistent, *PA* is consistent. But, $\vdash_{PA} \neg\delta$. Hence, not $\vdash_{PA} \delta$; that is, there is no proof in *PA* of δ. So $\texttt{Pf}(n, q)$ is false for every natural number n and, therefore, $\vdash_{PA} \neg Pf(\overline{n}, \overline{\ulcorner\delta\urcorner})$ for every natural number n. (Remember that $\overline{\ulcorner\delta\urcorner}$ is $\overline{q}$.) By ω-consistency, not $\vdash_{PA} \exists x\, Pf(x, \overline{\ulcorner\delta\urcorner})$, contradicting $(*)$.

Calling back our attention to the Diagonalization Lemma, we note that it holds for any formula ϕ with only the variable v free. In other terms, ϕ can be replaced by any formula with only one free variable. Accordingly, we look at Diagonalization Lemma in a different light, drawing from it the following consequences. Let us replace $\phi(v)$ with $\forall x\, Pf(x, v)$, and let n be the Gödel number of a proof of δ, i.e. $\vdash_{PA} \delta$.

$$\vdash_{PA} \delta \iff \forall x\, Pf(x, \overline{\ulcorner\delta\urcorner}) \qquad \text{Replacement} \qquad (6)$$

$$\iff Pf(\overline{n}, \overline{\ulcorner\delta\urcorner}) \qquad \text{Particularization} \qquad (7)$$

$$\iff Pf(\overline{n}, \overline{\ulcorner\beta(\overline{m})\urcorner}) \qquad \delta = \beta(\overline{m}) \qquad (8)$$

$$\iff Pf(\overline{n}, \overline{\ulcorner\forall x\, Pf(x, sb(\overline{m}, \overline{m})\urcorner}) \qquad \beta(\overline{m}) \text{ is } \phi(sb(\overline{m}, \overline{m})) \qquad (9)$$

$$\iff Pf(\overline{n}, \overline{\ulcorner\forall x\, Pf(x, sb(\overline{\ulcorner\beta(v)\urcorner}, \overline{m}))\urcorner}) \qquad m = \ulcorner\beta(v)\urcorner \qquad (10)$$

$$\iff Pf(\overline{n}, \overline{\ulcorner\forall x\, Pf(x, \overline{\ulcorner\beta(\overline{m})\urcorner})\urcorner}). \qquad \text{definition of } sb \qquad (11)$$

We can then observe that although these consequences can be deduced by the rules of first-order calculus, they do not truly hold in *PA*. Arithmetization is a one-to-one function, so that every string of symbols is matched to a unique number. By virtue of the *uniqueness* of the factorization of integers into primes, different sequences of expressions have different Gödel numbers. A proof in *PA* is a determined finite sequence of expressions composed of strings of symbols, and, therefore, each proof has its own Gödel number, which is *unique*. This can clearly be noticed through the definitions of the `Pf` and `Prf` predicates [12, pp 190-198] ([9, pp 162-176]).

$\texttt{Prf}(x)$: x is the Gödel number of a proof in *PA*:

$\exists u_{u<x}\, \exists v_{v<x}\, \exists z_{z<x}\, \exists w_{w<x}\, ([x = 2^w \wedge \mathtt{Ax}(w)]\vee$
$[\mathtt{Prf}(u) \wedge \mathtt{Fml}((u)_w) \wedge x = u * 2^v \wedge \mathtt{Gen}((u)_w, v)]\vee$
$[\mathtt{Prf}(u) \wedge \mathtt{Fml}((u)_z) \wedge \mathtt{Fml}((u)_w) \wedge x = u * 2^v \wedge \mathtt{MP}((u)_z, (u)_w, v)]\vee$
$[\mathtt{Prf}(u) \wedge x = u * 2^v \wedge \mathtt{Ax}(v)]$.

$\mathtt{Pf}(x, v)$: x is the Gödel number of a proof in PA of the formula with Gödel number v:

$$\mathtt{Prf}(x) \wedge v = (x)_{lh(x)\dot{-}1}.$$

Pf and Prf are primitive recursive, as the relations obtained from recursive relations by means of propositional connectives and bounded quantifiers are also primitive recursive, and they are expressible in PA, respectively, by Pf and Prf. According to their whole recursive codings, the consequences of the Diagonalization Lemma from (9) to (11) do not hold all together in PA. As we can notice, n cannot be the Gödel number of each one of the different proofs in PA of $\forall x\, Pf(x, sb(\overline{m}, \overline{m}))$, $\forall x\, Pf(x, sb(\ulcorner\overline{\beta(v)}\urcorner, \overline{m}))$ and $\forall x\, Pf(x, \ulcorner\overline{\beta(\overline{m})}\urcorner)$, since each one of them has its own Gödel number. There is no doubt that for their own symbolic composition each one of these formulas, $\ulcorner\forall x\, Pf(x, sb(\overline{m}, \overline{m}))\urcorner$, $\ulcorner\forall x\, Pf(x, sb(\ulcorner\overline{\beta(v)}\urcorner, \overline{m}))\urcorner$, and $\ulcorner\forall x\, Pf(x, \ulcorner\overline{\beta(\overline{m})}\urcorner)\urcorner$, is a different Gödel number. As displayed by the definitions of $\mathtt{Prf}(x)$ and $\mathtt{Pf}(x, v)$ and by the uniqueness of the factorization of the integers into primes, the proofs of these formulas can be generated only with different Gödel numbers, let us say q_1, q_2, and q_3. Each single $Pf(x_1, \overline{q_1})$, $Pf(x_2, \overline{q_2})$, and $Pf(x_3, \overline{q_3})$ is a different sequence of strings and is a unique Gödel number. It is, therefore, excluded that they would be all based on the same $Prf(\overline{n})$. In other terms, it is never the case that $Pf(\overline{n}, \overline{q_1})$, $Pf(\overline{n}, \overline{q_2})$, and $Pf(\overline{n}, \overline{q_3})$ in PA. n could be the Gödel number of only one of the proofs of q_1, q_2, and q_3, while the consequences of the Diagonalization lemma from (9) to (11) state, on the contrary, that they hold all together, from the same $Prf(\overline{n})$. This is clearly a uniqueness violation nested in the introduction of δ in PA on the basis of its definition by the function of substitution and the number-theoretical self-referring. By reason of that, the so called Diagonalization Lemma appears to be unacceptable in PA [2], [4], [6], [7].

Ending note

From the historical point of view, all the theorems that limit the expressive power of the logical mathematical thinking turn out to be characterized by some form of self-reference, often nestled beneath an apparently perfectly functioning framework of symbolic computation. This article is aimed to highlight how, even in the most extraordinary achievements of mathematical logic, self-contradicting inferential cores can be encoded, then widespread to the various areas of research, yielding limitative results, like incompleteness, undecidability and uncomputability. It also calls attention on how the Theory of Definition appears to offer a way to overcome these limitations.

References

[1] R. Carnap, *Logische Syntax der Sprache,* Wien: Springer, 1934.

[2] P. Cattabriga, "Beyond Undecidable," in *Proceedings of The International Conference on Artificial Intelligence, IC-AI-2000, Volume III*, (Las Vegas, Nevada, USA), June 26-29, 2000, CSREA Press, pp. 1475–1481, `arXiv:math/0606713`.

[3] P. Cattabriga, "Beyond Uncountable," in *Symposium pour le Développement de la Théorie des Ensembles accessibles*, (Institut Henri Poincaré, Paris), 3-4 september 2003, Revue Bio-Math N 160, 2003, `arXiv:math/0312360`.

[4] P. Cattabriga, "Observations concerning Gödel's 1931," *Bulletin of Symbolic Logic*, vol. 13, no. 3, pp. 382, 2007, `arXiv:math/0306038`.

[5] P. Cattabriga, "How to release Frege system from Russell's antinomy," *Bull. of Symb. Logic*, vol. 13, no. 2, pp. 269, 2007, `arXiv:0705.0901`.

[6] P. Cattabriga, "A note on Türing 1936," [Online]. Available: `arXiv:1308.0497`, 2013.

[7] P. Cattabriga, "Diagonalization as a Definition," in *InterSymp 2019 - International Conference on Systems Research, Informations and Cybernetics, Proceedings BB-IIAS19.*

[8] P. Cattabriga, "Paradox free," in *WIL2021 - 5th Workshop on "Women in Logic"*, 27th of June, 2021, Poster.

[9] K. Gödel, *On formally undecidable proposition of Principia mathematica and related systems I, 1931*, (Collected Works, Vol. I, Publications 1929-1936), New York: Oxford University Press, 1986, pp. 145–195.

[10] K. Gödel, *On undecidable propositions of formal mathematical systems, 1934*, (Collected Works, Vol. I, Publications 1929-1936), New York: Oxford University Press, 1986, pp. 346–371.

[11] S. Lešniewski, "Über Definitionen in der sogenannten Theorie der Deduktion," *Comptes rendus des sèances de la Societé des Sciences et des Lettres de Varsovie*, vol. 24, pp. 289–309, 1931.

[12] E. Mendelson, *Introduction to Mathematical Logic*, 4th ed., Chapmann & Hall, 1997.

[13] A. Padoa, "Logical introduction to any deductive theory," 1900. in *From Frege to Gödel. A Source Book in Mathematical Logic, 1879-1931*, Cambridge, Massachusetts: Harvard University Press, 1967, pp. 118–123.

[14] R. Rogers, *Mathematical logic and formalized theories*, North-Holland Pub. Company, 1971.

[15] C. Smoryński, *Self-Reference and Modal Logic*, Springer-Verlag, New York Inc., 1985.

[16] P. Suppes, *Introduction to Logic*, Dover Publicatios, Inc., New York, 1999.

Paola Cattabriga

Received June 14, 2022
Accepted August 04, 2022

University of Bologna
`https://unibo.academia.edu/PaolaCattabriga`

Appetite – Nutritionist Assistant

Nicoleta-Tatiana Trifan, Adrian Iftene

Abstract

Currently, we see how on TV, on social networks, in the press, on blogs, etc., famous people and advertising promote a healthy lifestyle and proper nutrition. These things have become especially prevalent during the pandemic when everyone has been isolated at home, and because of stress and sedentary lifestyle people have either gained unwanted pounds or lost too much weight. The current paper describes the Appetite application starting with the motivation and similar applications, continuing with the architecture and details about the main functionalities.

Keywords: nutrition, mobile application, augmented reality.

1 Introduction

In recent years, mobile phones have become an important part of our lives, with each of us devoting a lot of time to the applications on them. Whether it's apps that help us solve our work tasks or to learn [1], [2], [3], whether it's apps that allow us to connect with coworkers, friends, or family members [4], apps that relax or entertain us [5], [6], or utility applications that help us to have a better life [7], [8], [9], they have become indispensable.

The idea of the Appetite application was born during a pandemic when we heard from more and more people that they added extra kilograms and fail to return to their normal shape. We have known people who preferred to lose weight unhealthily and had lasting consequences for their health. We often thought about how we could integrate into a single application a lot of functionalities that would be useful in order to follow a correct diet. That's how we came up with the idea to create and develop the *Appetite* application.

©2023 by Nicoleta-Tatiana Trifan, Adrian Iftene

Appetite is a phone application that aims to promote and provide all the information and functionality needed to follow a healthy lifestyle. After testing similar applications that exist on the market, we noticed which of the options are more necessary, what users need, and what we need to put in this application. In this application, users will be able to take into account the calories and macro-nutrients consumed daily. They will also be able to track their progress, find out information about food by scanning only the bar-code. We also intend to bring, in addition to the practical connotation of the application, something informative, so users can interactively learn important things about their diet. The application is addressed to all people who want to make a change in their lives and do not know where to start or those who want to have some control over what they consume daily. There are no age limits or negative promotions that have a pro-anorexia or pro-obesity tendency. Similar approaches use Augmented Reality (AR) to recognize ingredients [10] and promote local or traditional recipes [11], [12].

2 Similar applications

Similar applications can be found on the online stores specific to each phone. In the following, we will describe the advantages and disadvantages offered by the software products already registered on the online market.

2.1 Yazio

The YAZIO[1] brand offers a calorie calculator that allows the management of food consumed for a day, the control of physical activities, the pursuit of food challenges, etc. It has its origins in Germany, founded by Florian Weissenstein and Sebastian Weber (see Figure 1).

Advantages of the application are as follows: (1) It has interesting free challenges for users; (2) It is possible to synchronize with a multitude of fitness applications or existing assistants, such as *Fit-Bit,*

[1] https://play.google.com/store/apps/details?id=com.yazio.android&hl=ro

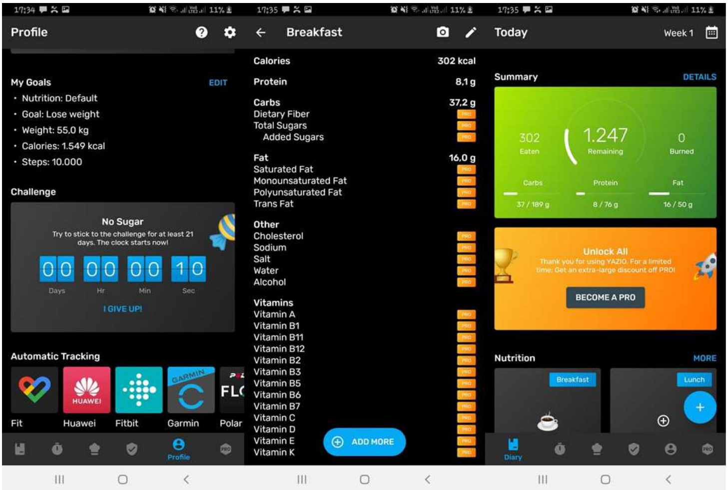


Figure 1. Examples from Yazio application

S-Health, Garmin, etc.; (3) The application has an attractive and intuitive design for users, which makes it easy to use; (4) When registering in the application, the personalized purpose is necessary to be set: weight loss, muscle development or weight maintenance; (5) The level of macronutrients consumed daily, calories, and other data of interest to the user can be easily tracked.

The disadvantages of the application are as follows: (1) Unfortunately, without an upgrade, the application remains only at the computer stage, which makes it less beneficial than other free applications that are on the market; (2) There are extremely few information resources in the free version; (3) Users are more encouraged to maintain their calorie count, not to adopt a healthy lifestyle with the help of appropriate information and assistance.

2.2 FatSecret

Mainly FatSecret[2] is characterized by computation of calories, but this application also comes with a wide range of other options. It was

[2] https://play.google.com/store/apps/details?id=com.fatsecret.android&hl=ro

created in 2007 by Leny Moses and Rodney Moses (see Figure 2).

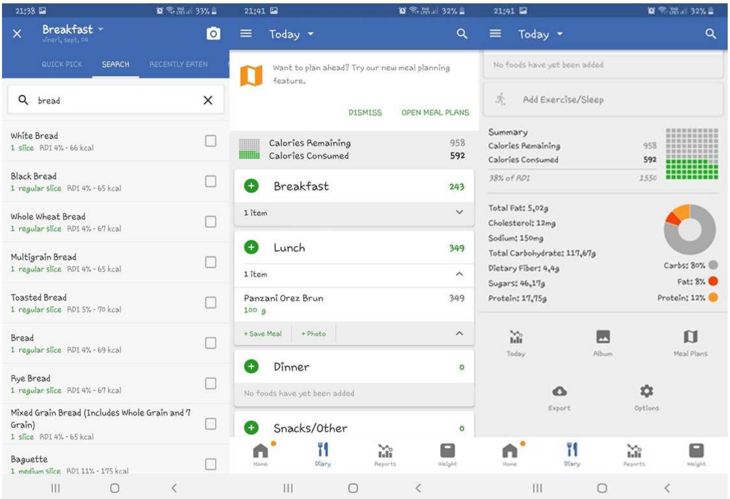


Figure 2. Examples from FatSecret application

Advantages of the application are as follows: (1) There are a variety of foods from which we can choose daily; (2) Provides a fairly successful approximate caloric calculation; (3) We can observe a very intuitive interface; (4) An important functionality would be the creation of reports on the user's success/ changes in his body over a period of time, based on recorded data; (5) The section, where we add exercises, can also be synchronized with any applications as *Google Fit* or *Samsung Health*; (6) Even if a diet plan enters Premium version with payment, we can access the chat, where several users of the application communicate and share their own recipes or ways to lose/build muscle mass.

The disadvantages of application are as follows: (1) Even if FatSecret offers a wide range of foods, sometimes we fail to find some products very often used. That's why we have to improvise: look for something similar or enter manually, looking for necessary data (amount of carbohydrates, fats, and proteins); (2) Unfortunately, it does not provide the absolutely correct level of sodium and protein in food; (3) The application significantly depreciates the level of sodium and protein in food, which could have negative consequences on the body.

2.3 LifeSum

LifeSum[3] is a Swedish application, created by Martin Wählby and Toue Westlund and launched on May 1, 2013. It is a startup that aims to help customers adopt a healthier way of life through the use of technology and applied psychology.

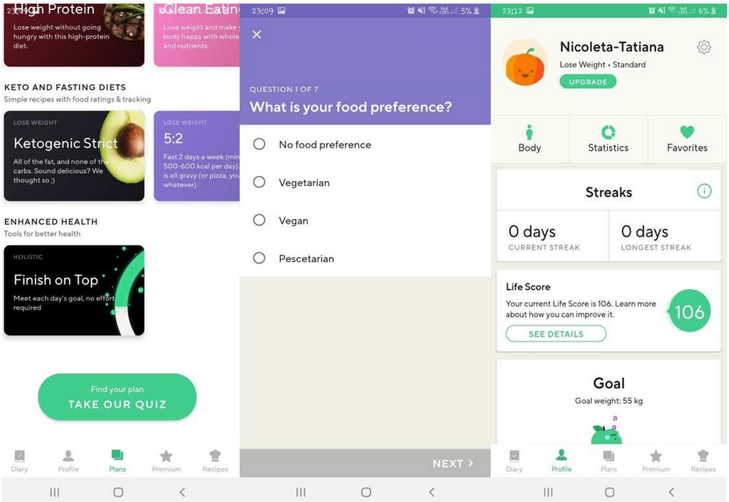


Figure 3. Examples from LifeSum application

The advantages of the application are as follows: (1) Due to the fact that the application has been added to Google Assistant, we can add, for example, food from the table, using voice, thus facilitating the option to enter data; (2) It offers a lot of interesting questionnaires that help when recommending a diet plan according to the client's preferences and needs; (3) Compared to other applications, they have exceeded the calorie calculation stage and extend their mission to promote a healthy lifestyle (this can be seen in the challenges proposed in

[3]https://play.google.com/store/apps/details?id=com.sillens.shapeupclub&hl=ro&gl=US

the application: 3 sugar-free weeks, 3 weeks of intensive sports, etc.); (4) Very nice and interactive design and animations that increase the attractiveness for customers.

3 Proposed solution

The application was structured in two scenes that assign a logical division of the application. The first of these is *SampleScene* which contains all the sections in the application, such as recording/logging, reports, recording daily consumption, etc., with which the user interacts. The augmented reality part, which refers to barcode scanning, is called the *AR option* and offers all the functionalities to scan the barcode and add this food to a certain mass (see Figure 4).

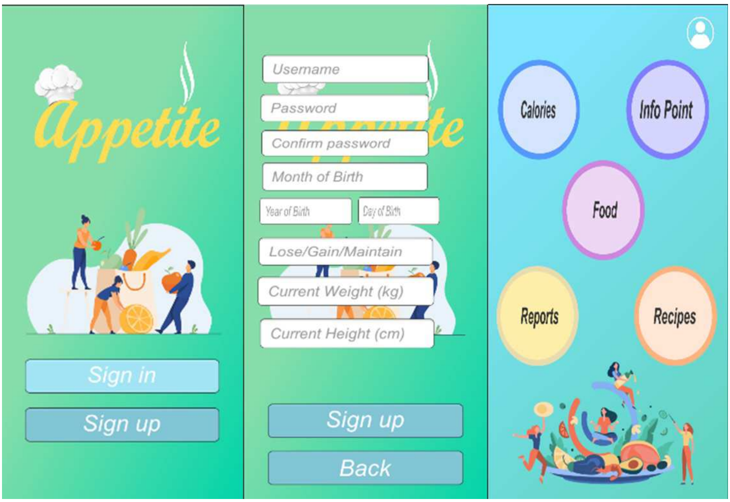


Figure 4. Appetite: Sign in, Register, and Main menu options

The first page with which the user interacts is the login page, where it is necessary to choose one of the listed options: *Sign in* or *Sign up* (registration in the application as a new user).

The next page in the application that you are going to interact within the application is the one that contains the main menu. After the connection is successful, the user is shown this page where they can choose what they want to do next. There are six potential actions: recording consumed foods (*Calories* button), consulting the information section (*Info Point* button), scanning the food barcode (*Food* button), consulting recipe recommendations (*Recipes* button), consulting reports on personal evolution (*Reports* button), and entry in the profile section (white icon in the upper right corner).

To get to the profile page, the button in the top right corner, which looks like a little white man, needs to be pressed from the main menu (see Figure 4 on the right). Within this page, you can change the personal information related to *height, weight, the numerical value of the desired body mass* (see Figure 5 on the left).

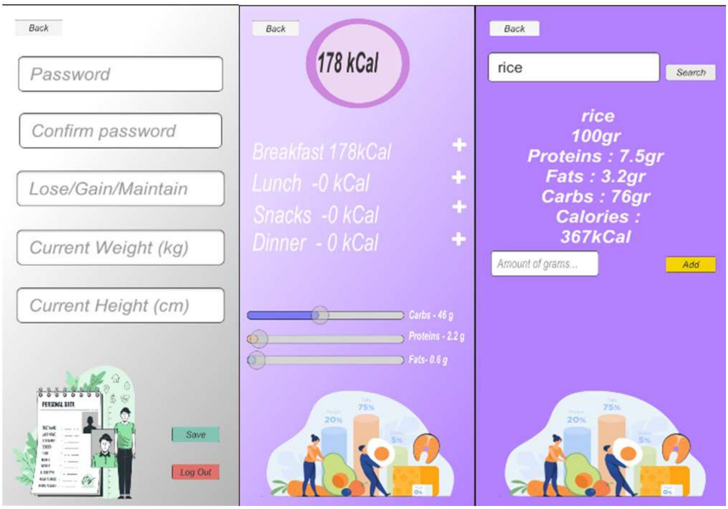


Figure 5. Appetite: options from the application

It is also possible to change the *login password*. When data is recorded in the Current Weight box, insertions are made in the updates table, so those correct calculations are made on the reports side. If a new value is entered in the box representing the desired body mass, the changes occur at the *user_data* table level. The Appetite app has a calorie calculator on one of the pages (see Figure 5 in the middle). At the top of the page, you can see the total amount of calories consumed that day. Also listed below are the calories consumed at each of the meals. These values are changed when the buttons for adding consumed products are pressed (plus button). At the same time, in this section, you can see the total amounts of carbohydrates, proteins, and fats consumed that day. If the user presses the button to add products, he has the possibility to change the numerical value of the calories at a certain mass.

What actually happens is that the user is redirected to a new page where he has a search box and can write the product that interests him (see Figure 5 on the right). After pressing the search button, the nutritional information about that food production will be displayed, indicating the weight for which it is displayed. Additionally, the possibility is ensured for the user to register this food for one of the meals, mentioning the quantity consumed.

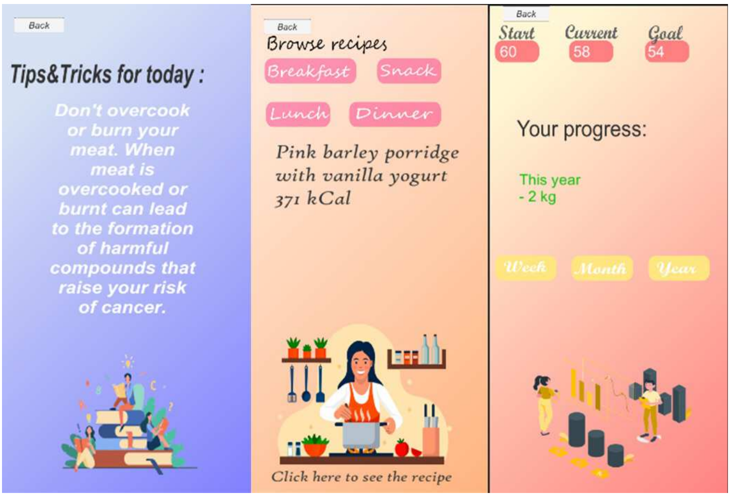


Figure 6. Appetite: Tips&Tricks, Browse recipes, and Progress options

The *Info Point* page displays information extracted from the application database (see Figure 6 on the left). These are a set of nutrition tips, and each time the user accesses this option one of them is displayed at random. Regarding the recipes, there is a separate button for each type of meal, thus sorting the recommended recipes (see Figure 6 in the middle). Each time the user presses one of the four buttons, a new title is displayed, to which the caloric value of the food is added. If the user is interested in any of the recipes displayed on the screen, the bottom of the page has a button that he can click and is redirected to a web page, where there are all the ingredients and cooking instructions.

In the *Reports page*, we have indicated at the top the information about the purpose of body mass, current value, and weight at the beginning of using the application (see Figure 6 on the right). The three buttons, Week, Month, and Year, calculate the progress for that term. If the person has lost weight, the information appears in green on the screen, otherwise, it is red. When the user accesses the *Food option*, the AR scene is loaded. In it, we find the page with instructions on how we should scan the barcode of the food product (see Figure 7 on the left).

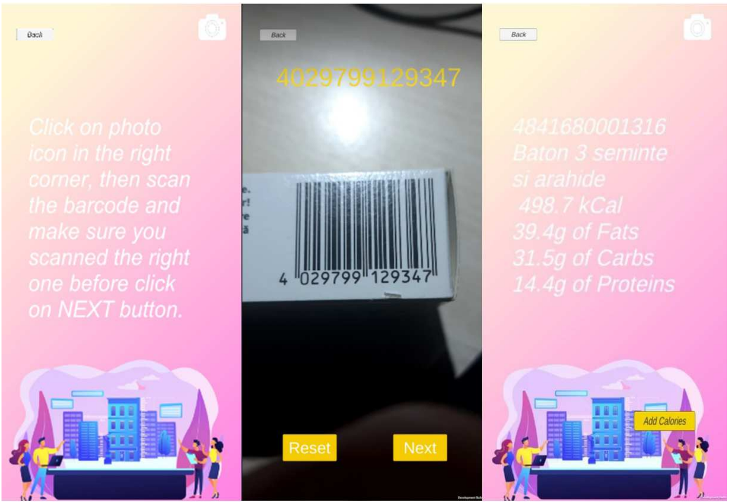


Figure 7. Appetite: AR option

Next, activate the card menu components, respectively turn on the camera, assume the correct positioning of the phone on the barcode, and scan. Later, the code read is displayed on the screen, if the user notices that it is wrong then he can choose the reset button and scan again or give next to see the information about that product (see Figure 7 in the middle) (similar with [11]). The product information is displayed in the same way as in the calorie calculator section (see Figure 7 on the right). It is also possible to add this product to the application database. The actual scanning takes place with the help of a library - QR Code/Barcode Scanner and Generator - Cross-Platform (Pro)[4]. It comes with all the necessary scripts to correctly identify the numbers on the label. Details such as the size of the dial-in which the code is analyzed or the code displayed on the screen have been edited and added by us.

4 Conclusion

The application comes in response to a current problem that is increasingly common – physical appearance. Control over our diet makes an important contribution to our physical and mental state [13]. By saying this, we mean avoiding certain food intolerances, frequent mood swings, or storing energy in larger quantities [14]. The application has an interface that can be easily interacted with. It has combined the features that are most popular and used in the process of weight loss or weight gain. It is based on a database large enough to demonstrate its purpose and workflow. We have described in the previous chapters what options it offers.

Compared to other applications in terms of idea and architecture, it can be said that it comes with an interesting look and content. Add new stuff to existing apps, such as recipe tips or recommendations, so it can be among the favorite apps for users of all ages.

From our point of view, the Appetite application would have some possibilities for updating and development due to its implementation architecture. The first aspect that could be improved would be the customization of some details in the application. For example, at the time of registration, calculate the BMI (body mass index) and recommend

[4]https://assetstore.unity.com/packages/tools/integration/qr-code-barcode-scanner-and-generator-cross-platform-pro-56083

to the user how many calories he should eat in a day. The second aspect that could provide an easier interaction between the user and the application would be the addition of the password reset option or the addition of a graph in the reports page that would show the continuous evolution, not just the numerical value.

Acknowledgments. This work was supported by project REVERT (taRgeted thErapy for adVanced colorEctal canceR paTients), Grant Agreement number: 848098, H2020-SC1-BHC-2018-2020/H2020-SC1-2019-Two-Stage-RTD. Data processing and analysis in this paper were supported by the Competitiveness Operational Programme Romania, under project SMIS 124759 - RaaS-IS (Research as a Service Iasi).

References

[1] C.G. Stativă, A. Iftene, and C.M. Miluț, "Assessment of Pronunciation in Language Learning Applications," in *Proceedings of the 2021 International Conference on Speech Technology and Human-Computer Dialogue (SpeD), IEEE Catalog Number: CFP2155H-ART*, (Bucharest, Romania), Virtual Conference, October 13-15, 2021, pp. 114–119.

[2] A. Simion, A. Iftene, and D. Gîfu, "An Augmented Reality Piano Learning Tool," in *Proceedings of the 18th International Conference on Human-Computer Interaction RoCHI 2021*, (Bucharest, Romania), 16-17 September, 2021, pp.134–141.

[3] I. Cercel and A. Iftene, "Planetarium - An Augmented Reality Application," in *Proceedings of the Conference on Mathematical Foundations of Informatics MFOI2020*, (Kyiv, Ukraine), January 12-16, 2021, pp. 62–77.

[4] A. Sirițeanu and A. Iftene, "MeetYou - Social Networking on Android," in *Proceedings, of 11th International Conference RoEduNet IEEE: Networking in Education and Research.*, (Sinaia, Romania), O. Rusu, Ed., 17-19 January, 2013, pp. 228–233.

[5] M.A. Cioată and A. Iftene, "Ear Trainer for Guitar Chords – An Android Application for Recognition of Guitar Chords," *International Journal of User-System Interaction*, vol. 13, no. 2, pp. 89–109, 2020.

[6] D.B. Chirica and A. Iftene, "Enhancing the Visit of the Botanical Garden of Iasi with the Help of Smart Devices and Augmented Reality," in *Proceedings of the Conference on Mathematical Foundations of Informatics MFOI2020*, (Kyiv, Ukraine), January 12-16, 2021, pp. 78–87.

[7] L.G. Coca, C.G. Cușmuliuc, and A. Iftene, "Automatic tarmac crack identification application," in *25th International Conference on Knowledge-Based and Intelligent Information & Engineering Systems*, vol. 192, (Szczecin, Poland), 8-10 September, 2021, pp. 478–486.

[8] A. E. Gîștescu, C.M. Miluț, T. Proca, and A. Iftene, "MedPlus – a cross-platform application that allows remote patient monitoring," in *25th International Conference on Knowledge-Based and Intelligent Information & Engineering Systems*, vol. 192, (Szczecin, Poland), 8-10 September, 2021, pp. 3751–3760.

[9] M. Rezmeriță, I. Cercel, and A. Iftene, "Stroke Detector - An Application that applies the F.A.S.T. Test to Identify Stroke," in *Proceedings of the 17th International Conference on Human-Computer Interaction RoCHI 2020*, 22-23 October, 2020, pp. 39–47.

[10] M. Chițaniuc, B. Miron, and A. Iftene, "FoodAR - An Augmented Reality Application used in Gastronomy," in *Proceedings of the 15th Conference on Human Computer Interaction - RoCHI 2018*, (Cluj-Napoca, Romania), 3-4 September, 2018, pp. 18–23.

[11] A. Iftene, D. Trandabăț, and V. Rădulescu, "Eye and Voice Control for an Augmented Reality Cooking Experience," in *24th International Conference on Knowledge-Based and Intelligent Information & Engineering Systems, 16-18 September*, Procedia Computer Science, vol. 176, 2020, pp. 1469–1478.

[12] M. Albayrak and E. Gunes, "Traditional foods: Interaction between local and global foods in Turkey," *African Journal of Business Management*, vol. 4, no. 4, pp. 555–561, 2010.

[13] A. Cavicchi and K.C. Stancova, "Food and gastronomy as elements of regional innovation strategies," Project: EU Joint Research Center – Agrifood Platform, Report number: JRC99987, Institute for Prospective Technological Studies, Spain, EUR 27757 EN, 2016, doi:10.2791/284013.

[14] M. B. Livingstone and L. K. Pourshahidi, "Portion size and obesity," *Advances in Nutrition*, vol. 5, no. 6, pp. 829–34, 2014.

Nicoleta-Tatiana Trifan[1,2,3], Adrian Iftene[1]

Received June 14, 2022
Accepted August 24, 2022

[1]"Alexandru Ioan Cuza" University of Iasi, Romania, Faculty of Computer Science
[2]Jysk, Aarhus, Denmark
[3]Aarhus University, Denmark
E–mail: {nicoleta.trifan,adiftene}@info.uaic.ro

Tourist Trip Design Problem – A Practical Application

Sergiu Amihăesei, Cristian Frăsinaru

Abstract

The last years have seen a rise in interest in the automation of itineraries. The problem consists in planning a route for tourists interested in visiting various points of interest (POIs). Although there exist a plethora of applications that tackle this objective, many of them are still rudimentary in nature, offering only predefined itineraries or being case studies on specific cities. This paper takes a new approach to solving the *Tourist Trip Design Problem*, using the constraint programming paradigm and external services that offer real-world data about points of interest.

Keywords: tourist trip design problem, web application, constraint programming

1 Introduction

In this paper we propose a solution for automatic itinerary creation, starting from the known *Orienteering Problem* [5]. The aforementioned consists of finding an order to visit a subset of nodes so that the total score is maximized and the allocated time is not exceeded.

Naturally, a number of variants have been derived from the *Orienteering Problem* [5]. Among the most recent applications is the *Tourist Trip Design Problem* (TTDP), which involves generating tourist routes per day that visit a set of points of interest, respecting the constraints provided by the tourist and the characteristics of the locations visited.

Multiple applications were developed over the years that introduced different variations in solving the *Orienteering Problem* [5]. In the paper *A Personalized Tourist Trip Design Algorithm For Mobile Tourist*

©2023 by Sergiu Amihăesei, Cristian Frăsinaru

Guides [4], Bruce L. Golden, et al. present a combined artificial intelligence and Guided Local Search meta-heuristic approach to solve the *TTDP*. More recently, Damianos Gavalas, et al. published *A Personalized Multimodal Tourist Tour Planner* [2], where they describe a mobile application that considers the option of using public transit and which incorporates time dependency to minimize time at transit stops. The survey *Orienteering Problem: A Survey of Recent Variants, Solution Approaches and Applications* [1] presents the aforementioned papers, together with their algorithmic approaches, and analyzes their performance in relation to other solutions.

In the following, we will describe an implementation of the *Tourist Trip Design Problem* in the form of a web application. To build the application, we turned to the *Spring* ecosystem for the server component, while for the client, we opted for the *Tailwind* and *Alpinejs* frameworks. Since the *Orienteering Problem* [5] is difficult to solve computationally, in the implementation of the algorithm we employed the use of the constraint-based programming paradigm, using the dedicated *Choco-Solver* [6] library.

In order to provide the user with relevant tourist locations, we adopted the use of the API provided by *Foursquare*[1]. The distance between objectives, as well as other geographical data, was fetched from the *Mapbox API*[2].

The paper is organized as follows: Section 2 introduces the formal model of the problem, Section 3 describes our solution as a constraint satisfaction problem, Section 4 shows the overall application, and Section 5 presents our conclusions.

2 Problem Description

The *Orienteering Problem* [5] was first introduced in the paper with the same name published in 1987. It starts from the premise of finding a minimum cost path that visits a subset of nodes, the objective being to maximize the total score obtained from the selected nodes and

[1]https://foursquare.com/
[2]https://docs.mapbox.com/api/overview/

to respect the time constraints. Thus, the *Orienteering Problem* [5] is combinatorial in nature, being a perfect match for the constraint programming paradigm.

We can define the *Orienteering Problem* [5] as follows:

Definition 1. [1] *Let $N = \{1, ..., |N|\}$ be a set of nodes, where each node $i \in N$ is assigned a score $S_i > 0$. The start and end nodes are fixed on node 1 and, respectively, node $|N|$. The objective is to find a path that visits a subset of N nodes in a given time T_{max}, maximizing the total score. It is assumed that the scores of the nodes can be summed, and each node can be visited at most once. The time to get from node i to node j is represented as t_{ij}.*

The *Orienteering Problem* [5] can also be formulated as an integer optimization problem:

Definition 2. [1] *Let the following decision variables: $X_{ij} = 1$ if a visit to node i is followed by a visit to node j, 0 otherwise. The variables u_i will be used in the subtour elimination constraints and will allow determining the position of the visited nodes in the path. Consider the following constraints:*

$$\text{Maximize} \sum_{i=2}^{|N|-1} \sum_{j=2}^{|N|} S_i X_{ij}. \tag{1}$$

The objective function (1) is used to maximize the total score of the visited nodes.

$$\sum_{j=2}^{|N|} X_{ij} = \sum_{i=1}^{|N|-1} X_{i|N|} = 1. \tag{2}$$

Constraints (2) ensure that the path starts at node 1 and ends at node $|N|$.

$$\sum_{i=1}^{|N|-1} X_{ik} = \sum_{j=2}^{|N|} X_{kj} \leq 1; \ \forall k = 2, ..., (|N| - 1). \tag{3}$$

Constraints (3) ensure path connectivity and guarantee that each node is visited at most once.

$$\sum_{i=1}^{|N|-1} \sum_{j=2}^{|N|} t_{ij} X_{ij} \leq T_{max}. \quad (4)$$

Constraint (4) limits the total travel time to fit within the maximum allowed time T_{max}.

$$2 \leq u_i \leq |N|;\ \forall i = 2, ..., |N|; \quad (5)$$

$$u_i - u_j + 1 \leq (|N| - 1)(1 - X_{ij});\ \forall i = 2, ..., |N|. \quad (6)$$

The combination of constraints (5) and (6) prevents subtours.

Since the emergence of the *Orienteering Problem* [5], multiple variants of it have been introduced, such as the *Tourist Trip Design Problem* (TTDP), which has been used as a reference point in this paper. TTDP is defined as a route planning problem for tourists interested in visiting multiple points of interest. The simplest version of the TTDP corresponds to the *Orienteering Problem* [5].

3 Algorithm Description

Using the *Choco-solver* [6] library, two algorithms have been developed for creating itineraries. In the following section, we will describe the second algorithm as an integer optimization problem. The first algorithm is being omitted for brevity, since it's a particular case of the one described below. Time is expressed in minutes by the designated integer variables, being the most meaningful unit of time for this purpose.

3.1 Unrestricted Generator

As previously stated, this algorithm is an extension of the first one, in which it is not mandatory for the user to assign the day of the visit for a point of interest. The day will be assigned by the algorithm, in order to minimize the total time per the entire schedule. Thus, certain days might be removed if deemed unnecessary.

Definition 3. *Let us consider the following variables provided by the front-end application: n, m representing the number of POIs, respectively days. The arrays $dayNumbers_i$, $daysStart_i$, and $daysEnd_i$ contain the data about days, namely the index of a day in the week and the time windows. Information about POIs is stored in the matrices $openingTimes_{i,j}$, $closingTimes_{i,j}$, and in the array $visitDurations_i$, which represent the time windows when a POI is open, respectively the time a user allocated for visiting a POI. The variable accommodation contains the index of the accommodation in the array of POIs, while the matrix $timeCost$ specifies the travel cost between each pair of POIs.*

Definition 4. *Using the Choco-Solver [6] library, we defined the following integer arrays: $visitTimesSt_i$ and $visitTimesEn_i$ which will provide the allocated time slot for a POI, ord_i that will give the index of a POI in the generated schedule, day_i which will give the day of a POI, $surplus_i$ used for memorizing the additional cost from the accommodation to the first visited POI, respectively $succCost_i$ that will store the travel cost to the next POI. Variable $totalTimeCost$ is a sum of every $succCost_i$, and will be used as the minimization criterion in the objective function.*

$$visitTimesSt_i, visitTimesEn_i \in [0, 1440], \forall i \in [0, n-1]. \tag{7}$$

Variables (7) will take values between the hours 00:00 and 24:00.

$$ord_i \in [0, n-1], day_i \in [0, m-1], \forall i \in [0, n-1]. \tag{8}$$

Variables (8) will take values according to the number of POIs, respectively days.

$$\begin{aligned} surplus_i &\in [0, 1440], \forall i \in [0, m-1]; \\ succCost_i &\in [0, 1440], \forall i \in [0, n-1]. \end{aligned} \tag{9}$$

The values of variables (9) cannot surpass the number of minutes in a day.

$$totalTimeCost \in [0, 1440 \cdot m]. \tag{10}$$

The values of variable (10) cannot surpass the total number of minutes between all the days.

We define the following constraints over the variables described above:

$$day_i = j, \forall i \in [0, n-1], \forall j \in [0, m-1], j \text{ user-specified.} \tag{11}$$

Constraint (11) will limit the placement of a POI to a day specified by the user, otherwise the day will be determined by the algorithm.

$$\begin{aligned}
max_{visitTimesSt_i} &= \max(daysStart_k, openingTimes_{i,dayNumbers_k})\\
min_{visitTimesSt_i} &= \min(daysEnd_k, closingTimes_{i,dayNumbers_k} -\\
&\quad - visitDurations_i)\\
max_{visitTimesEn_i} &= \max(daysStart_k, openingTimes_{i,dayNumbers_k}) +\\
&\quad + visitDurations_i\\
min_{visitTimesEn_i} &= \min(daysEnd_k, closingTimes_{i,dayNumbers_k})\\
\text{if } &\begin{cases} day_i = k \end{cases}\\
\text{then } &\begin{cases} max_{visitTimesSt_i} \le visitTimesSt_i \le min_{visitTimesSt_i} \\ \text{and} \\ max_{visitTimesEn_i} \le visitTimesEn_i \le min_{visitTimesEn_i} \end{cases}\\
&\forall i \in [0, n-1], \forall k \in [0, m-1].
\end{aligned} \tag{12}$$

Constraint (12) will place the POIs visit time window according to their schedules for the day that they were added to.

$$
\begin{aligned}
succCost_{term} =& surplus_k + timeCost_{i,accommodation} + visitDurations_i \\
dayEnd_{term} =& visitTimesSt_i + timeCost_{i,accommodation} + \\
& + visitDurations_i \\
\text{if } & \left\{ ord_i = n-1 \text{ and } day_i = k \right. \\
\text{then } & \begin{cases} succCost_i = succCost_{term} \\ \qquad\qquad \text{and} \\ dayEnd_{term} \leq daysEnd_k \end{cases} \\
& \forall i \in [0, n-1], \forall k \in [0, m-1],\ accommodation \neq -1.
\end{aligned}
\tag{13}
$$

Constraint (13) is used to determine the time from the accommodation to the first POI and from the last POI back to the accommodation, on the last day of the schedule. It also ensures that the time from the last POI to the accommodation does not exceed the day's time window.

$$
\begin{aligned}
max_{wait} =& \max(0, openingTimes_{i,dayNumbers_k} \\
& - daysStart_k + timeCost_{accommodation,i}) \\
\text{if } & \left\{ ord_i = 0 \text{ and } day_i = k \right. \\
\text{then } & \begin{cases} visitTimesSt_i = max_{wait} + \\ \qquad + timeCost_{accommodation,i} + daysStart_k \\ \qquad\qquad \text{and} \\ surplus_k = visitTimesSt_i - daysStart_k \end{cases} \\
& \forall i \in [0, n-1], \forall k \in [0, m-1],\ accommodation \neq -1.
\end{aligned}
\tag{14}
$$

Constraint (14) ensures that the time from the accommodation to the first POI in the schedule is taken into account. The waiting time for the opening of the POI is also taken into account.

$$
\begin{aligned}
max_{wait} =& \max(0, openingTimes_{j,dayNumbers_k} - \\
& - visitTimesEn_i + timeCost_{i,j}) \\
\text{if } & \left\{ ord_i + 1 = ord_j \text{ and } day_i = k \text{ and } day_i = day_j \right. \\
\text{then } & \left\{ succCost_i = max_{wait} + timeCost_{i,j} + visitDurations_i \right. \\
& \forall i, j \in [0, n-1],\ i \neq j,\ \forall k \in [0, m-1].
\end{aligned} \tag{15}
$$

Constraint (15) is used to set the visiting hours for a POI based on its successor in the day. It will take into account the waiting time for a POI to open, the travel cost, and the duration of visiting a POI.

$$
\begin{aligned}
\text{if } & \left\{ ord_i + 1 = ord_j \text{ and } day_i = k \text{ and } day_i \neq day_j \right. \\
\text{then } & \begin{cases} succCost_i = surplus_k + timeCost_{i,accommodation} + \\ \qquad + visitDurations_i \\ \qquad\qquad \text{and} \\ daysEnd_k \geq visitTimesSt_i + timeCost_{i,accommodation} + \\ \qquad + visitDurations_i \end{cases} \\
& \forall i, j \in [0, n-1],\ i \neq j,\ \forall k \in [0, m-1],\ accommodation \neq -1.
\end{aligned} \tag{16}
$$

Constraint (16) is an extension of constraint (13), applied to the rest of the days in the schedule.

$$
\begin{aligned}
max_{wait} = &\max(0, openingTimes_{i,dayNumbers_k} - daysStart_k + \\
&+ timeCost_{accommodation,i}) \\
\text{if } &\left\{ ord_i - 1 = ord_j \text{ and } day_i = k \text{ and } day_i \neq day_j \right. \\
\text{then } &\begin{cases} visitTimesSt_i = max_{wait} + timeCost_{accommodation,i} + \\ \quad + daysStart_k \\ \text{and} \\ surplus_k = visitTimesSt_i - daysStart_k \end{cases} \\
&\forall i,j \in [0, n-1],\ i \neq j,\ \forall k \in [0, m-1],\ accommodation \neq -1.
\end{aligned} \tag{17}
$$

Constraint (17) is an extension of constraint (14) for the rest of the days in the schedule.

$$
\begin{aligned}
\text{if } &\left\{ ord_i - 1 = ord_j \text{ and } day_i = day_j \right. \\
\text{then } &\left\{ visitTimesSt_i = visitTimesSt_j + succCost_j \right. \\
&\forall i,j \in [0, n-1].
\end{aligned} \tag{18}
$$

Constraint (18) ensures continuity of visiting hours from one POI to another on the same day.

$$
\begin{aligned}
\text{if } &\left\{ ord_i \neq n-1 \text{ and } ord_i + 1 = ord_j \right. \\
\text{then } &\left\{ day_i \leq day_j \right. \\
&\forall i,j \in [0, n-1].
\end{aligned} \tag{19}
$$

Constraint (19) is used to prevent alternating days in the schedule.

$$
\begin{aligned}
\text{if } &\left\{ ord_i + 1 = ord_j \text{ and } day_i \neq day_j \right. \\
\text{then } &\left\{ succCost_i = visitDurations_i \right. \\
&\forall i,j \in [0, n-1],\ accommodation = -1.
\end{aligned} \tag{20}
$$

Constraint (20) is applied in the absence of accommodation to limit the time of the last POI in a day to its visit duration.

$$visitTimesEn_i = visitTimesSt_i + visitDurations_i, \forall i \in [0, n-1]. \quad (21)$$

Constraint (21) is used to link the starting and ending hours of visiting a POI.

$$\begin{aligned} \text{if } &\left\{ ord_i = n-1 \right. \\ \text{then } &\left\{ succCost_i = visitDurations_i \right. \\ &\forall i \in [0, n-1],\ accommodation = -1. \end{aligned} \quad (22)$$

Constraint (22) is an edge-case of constraint (20) for the last POI in the last day, if no accommodation was provided.

$$ord_i \neq ord_j,\ \forall i, j \in [0, n-1],\ i \neq j. \quad (23)$$

Constraint (23) ensures that the positions of the POIs are distinct.

$$\sum_{i=0}^{n-1} succCost_i = totalTimeCost. \quad (24)$$

Constraint (24) is used to calculate the total time cost.

$$\text{Minimize } totalTimeCost. \quad (25)$$

The objective function (25) is used to minimize the total time cost over the entire schedule.

4 Application Showcase

The Front-end application is where the user provides the necessary data for the constraint solver. When first opening the application, the user will be provided with a page, where he can choose a city to create an itinerary for. After selecting a city, the user will be redirected to a new page, where he can create his itinerary.

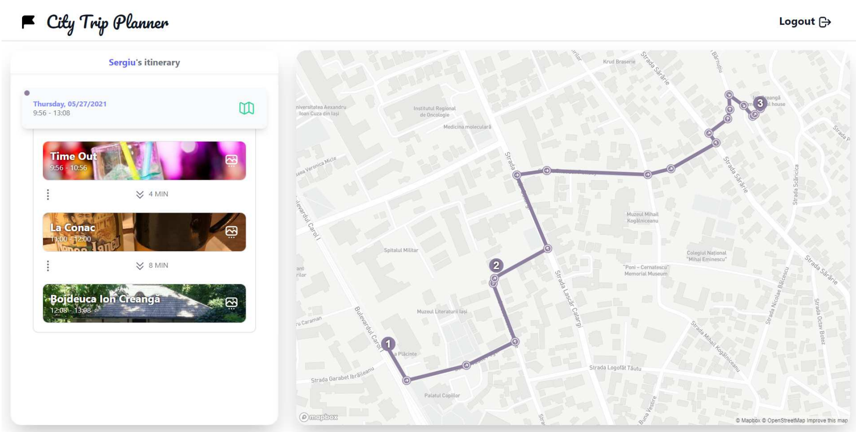


Figure 1. Itinerary Page

The itinerary creation page contains a map populated with POIs that load dynamically according to the user position. The POI data is cached from *Foursquare API* to allow for faster requests. On the left side of the page, there will be a panel, where the user can add days, together with their time windows, to the schedule, as well as remove them.

Upon clicking on one of the POIs on the map, a popup will open that will show extra information about the POI. After opening the popup, the user can add the POI to the schedule by clicking again on the map marker.

When the user finished creating the schedule, he can request the server to create an itinerary from the data. After the itinerary is created, the page will change accordingly (see Fig.1) and it will show the generated schedule in the left panel, as well as the routes between POIs on the map.

5 Conclusion

The application presented in this paper remarks itself as a notable contribution to a vast field with ample potential for development and innovation. Improvements to the branch of itinerary creation, such as user freedom in creating itineraries and route marking, stand out among similarly themed applications. However, there are still many areas that could be expanded upon, such as:

1. A notable extension would be the integration with an accommodation system such as *Booking.com*. This would eliminate the need for users to use multiple applications to plan their entire holiday, everything being done on the same platform.

2. Another improvement, complementing point 1, would be to extend the creation to multi-city itineraries, using different means of transportation. In that sense, the app could incorporate a way to buy tickets and book travel hours.

3. A review system independent of that provided by Foursquare could benefit the creation of itineraries. As such, the algorithm could be extended to create itineraries without human input, based on the reviews offered by the users.

References

[1] Aldy Gunawan, Hoong Chuin Lau, and Pieter Vansteenwegen, "Orienteering Problem: A Survey of Recent Variants, Solution Approaches and Applications," *European Journal of Operational Research*, vol. 255, no. 2, 2016, pp. 315–332.

[2] Damianos Gavalas, Vlasios Kasapakis, Charalampos Konstantopoulos, Grammati Pantziou, Nikolaos Vathis, and Christos Zaroliagis, "A Personalized Multimodal Tourist Tour Planner," in *The 13th International Conference on Mobile and Ubiquitous Multimedia*, 2014.

[3] Damianos Gavalas, Charalampos Konstantopoulos, Konstantinos Mastakas, and Grammati Pantziou, "A survey on algorithmic approaches for solving tourist trip design problems," *Journal of Heuristics*, vol. 20, no. 3, 2014, pp. 291–328.

[4] Wouter Souffriau, Pieter Vansteenwegen, and Greet Vanden Berghe, "A Personalized Tourist Trip Design Algorithm For Mobile Tourist Guides," *Applied Artificial Intelligence*, vol. 22, no. 10, 2008, pp. 964–985.

[5] Bruce L. Golden, Larry Levy, and Rakesh V. Vohra, "The orienteering problem," *Naval Research Logistics*, vol. 34, no. 3, 1987, pp. 307–318.

[6] Charles Prud'homme, Jean-Guillaume Fages, Xavier Lorca, *Choco solver documentation*, TASC, INRIA Rennes, LINA CNRS UMR, vol. 6241, 2016.

Sergiu Amihăesei[1], Cristian Frăsinaru[2]

Received June 14, 2022
Accepted July 30, 2022

[1]University "Alexandru Ioan Cuza" of Iasi, Faculty of Computer Science
E–mail: amihaeseisergiu@gmail.com

[2]University "Alexandru Ioan Cuza" of Iasi, Faculty of Computer Science
E–mail: acf@info.uaic.ro

Towards methodological foundations of propositional logic

Mykola Nikitchenko

Abstract

The paper is devoted to the foundations and main notions of propositional logic according to the three-level approach. This approach proposes logic explication in integrity of three levels of consideration: methodological (philosophical), conceptual (scientific), and formal (mathematical). We start with categories of the philosophical level and project them to scientific and then to mathematical levels. At the methodological level, logic notions are defined according to the pentadic scheme constructed of two triads: subject–object–activity and activity–thinking–logic. The main activity for propositional level is a choice operation based on the philosophical category of quality. This permits to give the explications of foundations of propositional logic.

Keywords: Propositional logic, foundations of logic, philosophical category of quality, three-level approach to logic.

MSC 2020: 03A05, 03B05, 03B30.

1 Introduction

The title of this paper may look strange because a general opinion is that all questions concerning propositional logic are answered and therefore it is not clear what new results can be obtained. But we will try to demonstrate that it is still possible to add to the topic something new. In particular, we aim to clarify in more details relationship between a system of philosophical categories and propositional logic. This work is a part of our research for explication of methodological foundations of mathematical logic with respect to philosophical theory of development [1, 2]. This research was initiated in [3, 4].

©2023 by Mykola Nikitchenko

In this paper, foundations are treated as the study of the basic notions of mathematical logic, their interrelations, and their development. Having specified notions as a main component of the foundations, we can classify notions according to the levels of their generality. The most general notions are considered as philosophical categories. Such notions are infinite notions [1]. Therefore, the next level constitutes finite (in philosophical sense) notions, and finally, the third level is presented by formal notions. Finite notions represent scientific notions (concepts) and formal notions represent mathematical notions.

These short considerations lead to the idea to present foundations of mathematical logic at three levels: methodological (philosophical), conceptual (scientific), and formal (mathematical). At each level of considerations, we specify notions related to mathematical logic and then describe their interrelations. Let us note that the importance of relating philosophy and mathematical logic is widely recognized [5]. Here we explicitly add one more level: the level of finite categories (scientific notions). The proposed treatment of three levels of foundations is just a general scheme. To make it practical, we should indicate to those notions which form the basis of logic.

In this paper, we will substantiate that notions of mathematical logic are based on the philosophical category of activity. Thus, the proposed approach to the foundations of mathematical logic may be called activity-oriented. Finally, we demonstrate that propositional logic is derived from a such activity as choosing objects on the base of their qualities.

The paper is structured in the following way. In Section 1, a short introduction is presented. Section 2 is devoted to the description of a three-level scheme of logic notions explication. In Section 3, we concentrate on the explication of the main notions of logic according to the proposed three-level scheme. Section 4 describes the foundations and the main notions of propositional logic developed according to the proposed scheme. At last, Section 5 presents conclusions.

2 Thee-level scheme to logic foundations

In this section, we give a short description of the methodological (philosophical), conceptual (scientific), and formal (mathematical) levels.

The methodological level aims to present the most general features of objects. Such questions belong to gnoseology; therefore, we use principles of gnoseology to describe the foundations of logic [3]. The central part of gnoseology is a system of philosophical categories. The first system of categories is attributed to Aristotle. Later, Kant extended his system. In this paper, we advocate the usage of an elaborated system developed by Hegel [1]. The main principles of Hegelian approach are:

- development from the abstract to the concrete;
- grouping categories by stages of the development;
- unity of opposite categories in the third category;
- unity of theory and practice.

The *principle of development (ascent) from the abstract to the concrete* states that the development is definitely oriented change from the abstract to the concrete (from the simple to the complex, from a lower level to a higher one, from the old to the new).

The *principle of grouping categories by stages of the development* means that we should identify stages and relate categories to them. Here we indicate the following stages of being [1, 2]:

- *determinate*
- *real*
- *actual*

Among categories related to these stages, we identify only those categories that evolve to the notions of mathematical logic:

- determinate being: *quality*, *quantity*, and *measure*;
- real being: *thing*, *property*, and *relation*; *form* and *content*; *whole* and *part*;

- actual being: *action* and *reaction*; *cause* and *effect*; *ends* and *means* (*goals* and *tools*).

The *principle of unity of opposite categories in the third category* is often presented by the development triad: *thesis – antithesis – synthesis*).

This principle can be explained in the following way. Any object can be considered as the totality of all its aspects. But all aspects cannot be immediately investigated; therefore, a certain aspect is chosen by *abstraction*, then another aspect is chosen by *negation*, and finally, the *union* (*synthesis*) of both aspects is presented. It is important that chosen aspects be essential ones and be chosen according to practical and theoretical criteria. In Hegel's philosophy, the movement to antithesis (and then to synthesis) is often considered as *sublation* (negation that preserves the essential characteristics).

The *principle of unity of theory and practice* can be formulated in the following way: theory and practice should be considered as influencing each other. This principle substantiates development of logic notions in *praxeological* perspective, i.e. this development should be based on analysis of human action in subject domains. The praxeological aspect is one of the main philosophical aspects relating categories of *subject* and *object* such as *ontological*, *gnoseological*, and *axiological* aspects.

At the second (scientific) level, additional principles are introduced that are used to develop the basic scientific notions and their interrelations. Such system of basic notions presents the structure and properties of a domain under investigation. Developing notions (concepts) of logic systems, we choose the categories related to logic and "project" ("finitize") them to the scientific concepts (notions). Such projections transfer categories and their relationships from the philosophical onto the scientific level. We start with the categories *universal*, *particular*, and *individual*. Their projections to the scientific level are called *intension*, *particular intension* (at the scientific level), and *extension* respectively. This terminology stems from the set-theoretic treatment of *extensionality* that defines two sets equal if they consist of the same elements. Therefore, at this level we adopt and follow the *principle of intension–extension integrity*: a notion should be presented by the following triad: *notion intension – notion extension – integrity of intension and extension; the intension in this integrity plays a leading role.*

At the mathematical (formal) level of consideration, the notions from the previous level are restricted in order to get their reasonable formalization. This formalization should take into account intensional and extensional aspects of notions. Importance of this level is explained not only by its formality but also by possibilities to construct a basis for automatization of various processes related to logic.

Traditionally, mathematical formalization is based on the notion of set. This notion is defined in extensional manner. But now we can see more examples that indicate on necessity of adding intensional aspects to the notion of set [6, 7]. Still, even being extended, the notion of a set is not expressive enough to represent numerous intensional aspects. Therefore, we propose to base mathematical formalization not only on the notion of set but also on the notion of function (mapping). Of course, this notion of function should be extended with intensional aspects. Such considerations give the *principle of intensionalized function-theoretic formalization of the notions of logic.*

3 Categories related to logic

Based on the formulated principles, we can now identify notions (categories) related to logic at the philosophical level.

We choose an *anthropological approach* as a starting point. Then the first question is: what is the most important characteristic of a man?

The analysis of various aspects of a human being permits us to say that the most important thing for appraising a man is his *activity.* This statement was formulated in different forms, we mention just a few. For example, activity plays an exceptional role in the philosophy of Fichte; for him "the world does exist in human activity" [8].

Similar ideas were formulated by Hegel; some additional considerations on the topic can be found in [9, 10].

Of course, importance of activity was clearly understood by famous thinkers [11], for example:

- Happiness is activity (Aristotle);
- To live well is to work well, to show a good activity (Thomas Aquinas);
- The highest activity a human being can attain is learning for understanding, because to understand is to be free (Baruch Benedict de Spinoza);
- The pursuit of truth and beauty is a sphere of activity (Albert Einstein).

Activity presupposes the following three moments [4]:

1) a man should be active;
2) his activity should be thoughtful (rational, reasonable, meaningful, comprehended, sensible);
3) man's activity should be fruitful (logical, productive, efficient) in the sense that it gives the desired results.

Analyzing these moments, we can say that the first moment represents the category of *activity*, the second leads to the category of *thinking*, and the last moment gives the category of *logic*.

Clearly, these categories are related in a certain way.

First, thinking can be considered as a sublated activity. Second, logic can be treated as a sublated thinking. But additionally, logic unites activity and thinking. The measure of this union is called the *truth* relation.

Thus, three categories constitute the *logic triad* in which activity is thesis, thinking is antithesis, and logic is synthesis (Fig.1).

The logic triad represents the initial explication of logic, but in this triad, the category of activity is not specified. We do this by Fichte's triad: *subject – object – activity* (Fig.2). In this triad, activity integrates subject and object.

Summarizing, we have specified the *logic pentad* (Fig.3):

subject – object – activity – thinking – logic.

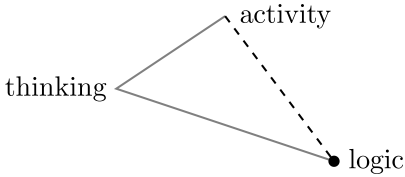


Figure 1. Logic triad

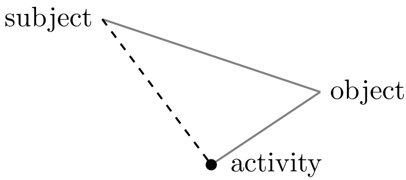


Figure 2. Activity triad

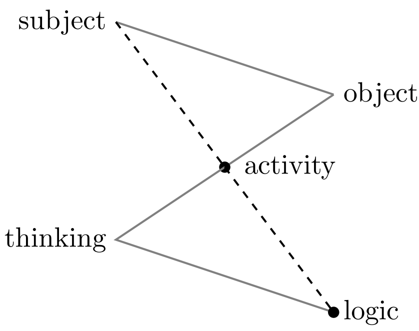


Figure 3. Logic pentad

The logic pentad may be regarded as a minimal diagram identifying the main notions of logic. This pentad can be generalized by adding new notions or deleting the existing ones and incorporating their meanings into the remaining notions. For example, we can delete the notion of subject obtaining objectivity of other notions; or we can delete the notions of subject, object, and activity obtaining Hegelian definition of logic as a sublated thinking traditionally formulated in the following form: logic is the science of thinking.

In the logic pentad, the category of activity plays a central role. Therefore, the proposed approach to logic explication may be called *activity-oriented*. This means that further explication of the logic notions should be based on the category of activity.

Activity can be unfolded by the following pentad (Fig.4):

goal – means – plan (program) – plan execution – result.

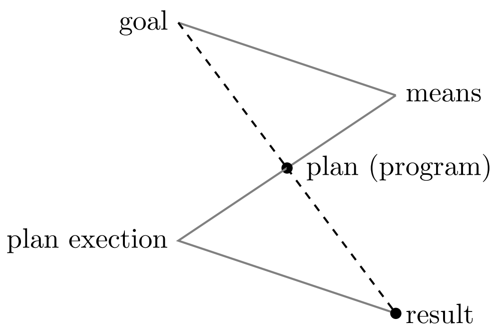


Figure 4. Pentad of unfolded activity

This pentad is constructed from two triads: *goal – means – plan* and *plan – execution – result.*

To make further progress in the explication of logic notions, we should formulate what kind of activity is substantial for logic. Analysis of such activities leads to the following conclusion: the main activity related to logic is *choosing objects from a certain class.* Indeed, formulas of a logic language specify a certain class of objects by their truth domains; in other words, interpreting formulas over a class of objects we obtain a characteristic function for a subclass of objects. Activity of choosing objects can be rather complicated; therefore, we associate it with the levels of the category of *being*: *determinate*, *real*, and *actual* levels [1]. These levels give rise to the different levels of mathematical logic. In particular, the level of determinate being leads to propositional logic, the level of real being leads to predicate logic, and the level of actual being leads to action (program) logic.

In this paper, we concentrate on the foundations of propositional logic.

4 Foundations and main notions of propositional logic

To specify the foundations of propositional logic, we should study in more detail the main activity related to logic (choosing objects from a certain class) at the level of determinate being. At this level the main categories are *quality*, *quantity*, and *measure*. According to the principle of the development from the abstract to the concrete, we start with the most abstract category of this level: with the category of quality. Hegel writes that "something has a quality, and in this quality it is not only determined but delimited; its quality is its limit and, affected by it, something remains affirmative, quiescent existence" [1]. So, we adopt that at the level of determinate being objects can be characterized by their qualities. But note that if a quality of an object is changed, then a new object appears. We stop here not going into further characteristics of the category of quality because the described characteristics are sufficient to move to the next level. Summarizing, it is possible to say that we accept the *ontological assumption* (*foundations*) that at the determinate level of being objects are specified by their qualities.

At the scientific level, it is necessary to make finitizations (to limit) of the category of quality. This is done in the following way:

1) we consider objects with a fixed set of qualities;

2) we assume that qualities can be named;

3) there is a certain procedure which given an object can list its qualities.

Here we assume that an object quality has some quality characteristics; for the sake of simplicity we also call them qualities of an object.

The formulated items constitute *gnoseological assumptions* (*foundations*) because they describe what knowledge can be extracted from a given object. Please note that gnoseological assumptions can be treated as parametrical ones, say, we can use *finite or infinite sets of qualities*, or for the third item we can identify *total or partial procedures.*

At last, let us consider formalization of the identified activity – choosing objects by their qualities – with respect to the ontological and gnoseological assumptions. Characteristics of this activity will constitute *praxeological assumptions* (*foundations*). To formalize this activity, a formal model of an object should be first constructed. As was said earlier, objects are specified by the sets of their qualities. In our case, it is reasonable to make formalization not on the base of the notion of a set, but on the notion of function that better reflects the activity-oriented approach to logic explication. Therefore, a set of qualities of an object is represented by a total (or partial) characteristic function (by a function from the set of names of qualities to Boolean values). This function is called a Boolean assignment and can be regarded as an extensional model of an object. (In our papers such assignments are also called nominative data.)

At the second step of formalization, a formal model of the chosen *class of objects* should be constructed. It can be done in two steps:

1) identifying object qualities obtaining Boolean assignments;

2) defining a characteristic function from the set of Boolean assignments to Boolean values.

This can be written in the following way. Let Obj be a class of objects, Q be a finite set of quality names, and $Bool = \{T, F\}$ be the set of logical values. Then the class $Q \to Bool$ of partial functions from V to $Bool$ is called the class of Boolean assignments (or quality Boolean models of objects) and is denoted ${}^{Q}Bool$. A mapping $qb : Obj \to {}^{Q}Bool$

is called a *quality evaluation function.* Partial functions from ${}^{Q}Bool$ to $Bool$ are called quality Boolean models of classes; their set is denoted as QBB^{Q}. Such functions are characteristic functions of classes of Boolean models of objects; they represent various activities of choosing objects by their Boolean models. So, for any activity $choose : Obj \to Bool$ there is its quality evaluation function qb and its characteristic function $qbb \in QBB^{Q}$ such that for any obj from Obj we have that $choose(obj) = qbb(qb(obj))$.

The next step of formalization is representing QBB^{Q} as a certain algebra. Considering QBB^{Q} as a carrier of the algebra, compositions of disjunction and negation are traditionally used as basic operations of this algebra.

So, algebra of partial Boolean functions

$$A^{Q} = (QBB^{Q}; \vee, \neg)$$

is defined. This algebra is a bases for propositional logic construction.

To construct a logic, we should specify [12]

- intensions of the logic;
- class of algebras which form semantic base (extension) for the logic;
- the logic language (based on logic signature);
- the class of interpretations of language constructs;
- the consequence relation;
- the inference relation based on some calculus.

Logic intensions describe aims of the logic, its relations with the levels of being, its orientation to some class of activities, its ontological, gnoseological, praxeological, and other philosophical assumptions (foundations) accepted for the logic construction, its main notions and their interrelations, etc. The intensional component of the logic is primarily intuitive and rather vague. Understanding of intensions is laid down in subsequent more formal definitions of logic notions. In our case of propositional logic, its main intension is the formalization of choosing procedures of objects by their qualities; thus, propositional logic can be related to the level of determinate being; ontological foundations describe "the world" of propositional logic as a class of objects of this level; gnoseological foundations specify qualities and methods of their identifications; praxeological foundations characterize procedures of choosing the objects.

Logic extensions are specified as classes of algebras which conform to the logic intensions and which can be considered as formal models of subject domains. In our case, this is an algebra of Boolean functions A^Q.

The language of the logic (formulas, terms, programs, etc.) is defined inductively upon the logic signature which includes symbols for basic mappings and types of the logic compositions. Actually, the language is derived from the algebra expressions. In our case of propositional logic, the signature consists of the set of names (propositional variables) Q, and names (symbols) of operations (compositions) over QBB^Q.

The logic interpretations define semantics of the logic language. In our case, it is a mapping that transforms formulas of propositional language to Boolean functions. The consequence relation specifies the understanding of valid reasoning in the logic. It is important to note that logic requires various types of consequence relations such as irrefutability relation, consequence on truth, consequence on falsity, consequence under conditions of undefinedness, etc. [13].

The inference relation aims to formalize the consequence relation. In our work, we use primarily sequent calculi to specify the inference relation [12–14].

Varying ontological, gnoseological, and praxeological assumptions, we can construct different kinds of propositional logic. For example, we can consider finite or infinite set of qualities, finite or infinite formulas, total or partial Boolean models of objects, total or partial Boolean models of classes, monotone or nonmonotone logics, etc. Numerous mathematical results were obtained for such logics and their components [12–15]. Here we will not go into details.

Thus, it is possible to construct various classes of logics which can be considered as restricted formalizations of certain philosophical categories. Such construction has been done for foundations of propositional logic.

5 Conclusion

In the paper, we have demonstrated the close relationship of philosophical categories with mathematical logic, and in particular, with propositional logic. Several ideas have been used for such demonstration. First, three-level approach has been adopted. The main principle of this approach is explication of the notions of mathematical logic in integrity of philosophical, scientific, and mathematical levels. Second, at the philosophical level, it has been substantiated that the main categories related to logic constitute the following pentad: subject – object – activity – thinking – logic. In this pentad, the category of activity is the main category upon which logic is constructed. So, the proposed explication of the foundations and notions of logic can be called activity-oriented. The main activity related to logic is choosing objects of a certain class. Third, several levels of the category of activity associated with such levels of being as determinate, real, and actual being have been identified. It has been shown that formalization of category of quality directly leads to propositional logic. Such approach to propositional logic formalization simultaneously describes ontological, gnoseological, and praxeological foundations of propositional logic.

The obtained results demonstrate that philosophical categories permit to formulate methodological foundations upon which propositional logic is constructed. Such approach can be further generalized for predicate and program logic [4], and also for the notions of Data–Information–Knowledge–Wisdom (DIKW) hierarchy [16].

References

[1] G.W.F. Hegel, *Hegel's Science of Logic* / translated by A.V. Miller; foreword by J.N. Findlay, London: G. Allen & Unwin, 1969.

[2] V. Gavryluk, *Theory of Development*, Kyiv: Publishing House of European university, 2009. (in Russian)

[3] M. Nikitchenko, "Gnoseology-based Approach to Foundations of Informatics," in *Proceedings of the ICTERI-2011*, CEUR Workshop Proceedings, vol. 716, 2011, pp. 27–40.

[4] M. Nikitchenko, "Towards Defining Program Logics via Three-level Scheme," in *Proc. MFOI-2020*, Kyiv: Interservice, 2021, pp. 324–329.

[5] *Philosophy of Logic*, Dov M. Gabbay, Paul Thagard, Eds. North Holland: John Woods, 2006.

[6] P. Apostoli, R. Hinnion, A. Kanda, and T. Libert, "Alternative Set Theories," in *Philosophy of Mathematics*, A.D. Irvine, Ed. Elsevier, 2009, pp. 461–491.

[7] M. Nikitchenko and A. Chentsov, "Basics of Intensionalized Data: Presets, Sets, and Nominats," *Comp. Sc. J. of Moldova*, vol. 20, no 3, pp. 334–365, 2012.

[8] A. Blunden, "Johann Fichte: The Subject as Activity," Abandoned Draft, 2005. [Online]. Available: https://www.ethicalpolitics.org/ablunden/works/the-subject.htm.

[9] M. Filoni, "Man is action, not being" Hegel contra Heidegger in an unpublished essay by Kojève," *Philosophical Inquiries*, vol. 8, no. 2, pp. 203–208, 2020.

[10] M. Vater, "Reason and Agency in Kant and Fichte," *Revista de Estud(i)os sobre Fichte*, vol. 16, 2018. [Online]. Available: http://journals.openedition.org/ref/845.

[11] *Activity proverbs.* [Online]. Available: https://proverbicals.com/activity.

[12] M. Nikitchenko and S. Shkilniak, *Applied Logic*, Kyiv: Publishing house of Taras Shevchenko National University of Kyiv, 2013. (in Ukrainian).

[13] M. Nikitchenko, O. Shkilniak, S. Shkilniak, and T. Mamedov, "Completeness of the first-order logic of partial quasiary predicates with the complement composition". *Comp. Sc. J. of Moldova*, vol. 80, no. 2, pp. 162–187, 2019.

[14] M. Nikitchenko, S. Shkilniak, *Mathematical logic and theory of algorithms*, Kyiv: Publishing house of National Taras Shevchenko Univ. of Kyiv, 2008. (in Ukrainian).

[15] M. Nikitchenko, I. Ivanov, A. Korniłowicz, and A. Kryvolap, "Extended Floyd–Hoare logic over relational nominative data," in *Information and Communication Technologies in Education, Research, and Industrial Applications. ICTERI 2017* (Communications in Computer and Information Science, vol. 826). Cham: Springer, 2018, pp. 41–64. https://doi.org/10.1007/978-3-319-76168-8_3.

[16] M. Nikitchenko, "Revisiting Data-Information-Knowledge-Wisdom Hierarchy from a Logic Perspective," in *2021 IEEE 23rd Int Conf on High Performance Computing & Communications; 7th Int Conf on Data Science & Systems; 19th Int Conf on Smart City; 7th Int Conf on Dependability in Sensor, Cloud & Big Data Systems & Application (HPCC/DSS/SmartCity/DependSys)*, 2021, pp. 2159–2165, doi: 10.1109/HPCC-DSS-SmartCity-DependSys53884.2021.00322.

Mykola Nikitchenko

Received June 15, 2022
Accepted July 15, 2022

Taras Shevchenko National University of Kyiv
64/13, Volodymyrska Street, City of Kyiv, Ukraine, 01601
E–mail: mykola.nikitchenko@gmail.com

Using Natural Language Processing for Centralising Research Project Calls

Ervin Maftei, Cristian-Mihai Rosu, Cristi-Constantin Rusu, Adrian Iftene

Abstract

There is a multitude of research project calls scattered across a variety of websites – some are easier to find than others or just very difficult to parse for the right information. The task for this project is to bring all relevant information of such projects together in one centralized website and present them in an easily digestible format, useful to prospecting researchers, that will also allow for searching and filtering operations. In order to make this possible, we have built a system comprised of three main modules: (1) the crawling component, (2) the semantic analysis component, and (3) the back-office component for database operations and UI. With the help of this third component, especially, future work can potentially enable us to create applications on various platforms such as mobile, desktop, or web.

Keywords: Natural Language Processing, Text Mining, Crawling, REST API.

1 Introduction

Currently, researchers can benefit from funding and material resources from several funding lines, both national and international. Their diversity and the different way of applying them greatly complicate the life of researchers, who have to actively monitor several funding lines, with different deadlines, with different requirements, with specific target groups, etc. The project we present in this paper aims to unify the information distributed in all these calls, and offer a unique solution, where the user can find all the information they want.

©2023 by Ervin Maftei, Cristian Rosu, Cristi Rusu, Adrian Iftene

Below, in the continuation of the paper, we will be able to see the following:

1. the sources of financing that we intend to consider. As we will see, some we have already considered, and others we will consider in the next period.

2. the way we took the data from the national and international sites, in order to be able to process them locally and to be able to use them later in the search process.

3. the techniques used in the area of natural language, so necessary in the context in which the data we obtain do not follow a precise format, and we need specific text mining techniques to structure our information as well as possible.

4. structured information can be queried by users to obtain the desired information, using various filters and sorting methods.

2 Project Calls

The funds obtained by the research teams from the national and international project calls allow them to work on the ideas they have and to develop them further. They are needed both to keep the teams together and to acquire the material resources needed for the research process, or for the activities of publishing and disseminating the results.

The **international resources** considered are:

- European Commission[1]:
  - Horizon Europe (HORIZON),
  - Defense Fund (EDF),
  - Environment and Climate Action (LIFE),
  - Creative Europe (CREA),
  - Euratom Research and Training Program (EURATOM),

[1] https://ec.europa.eu/info/index_en

  - Erasmus+ Programme (ERASMUS),
  - EU Anti-fraud Programme (EUAF),
  - Citizens, Equality, Rights and Values Programme (CERV),
  - Single Market Programme (SMP),
  - EU4Health Programme (EU4H),
  - European Solidarity Corps (ESC),
  - Justice Programme (JUST),
  - Protection of the Euro against Counterfeiting Programme (PERICLES),
  - European Social Fund + (ESF).

- European Research Council (ERC)[2];
- European Court of Auditors (ECA)[3];
- Marie Skłodowska-Curie Actions[4];
- COST (European Cooperation in Science and Technology)[5];
- Humboldt Foundation[6].

The **national resources** considered are:

- Executive Unit for the Financing of Higher Education, Research, Development and Innovation (UEFISCDI)[7]:
  - ”Spiru Haret” Research Fellowships,
  - Demonstration Experimental Project (PED),
  - Grant Scheme (SGS),
  - EEA/NO - Mobility Projects,

[2] https://erc.europa.eu/
[3] https://www.eca.europa.eu/en/Pages/ecadefault.aspx
[4] https://ec.europa.eu/research/mariecurieactions/
[5] https://www.cost.eu/
[6] https://www.humboldt-foundation.de/en/
[7] https://uefiscdi.gov.ro/despre-uefiscdi

  - Women in Science - L'Oréal-UNESCO scholarships,
  - Mobility projects for researchers,
  - Mobility projects for experienced diaspora researchers,

- German Academic Exchange Service (DAAD)[8];
- Scholarships from the Romanian Cultural Institute (ICR)[9];
- North-East Regional Development Agency (ADRNordEst)[10];
- New Europe College (NEC)[11];
- Ministry of European Investment and Projects (MFE)[12];
- The Fulbright Program[13];
- The Credit and Scholarship Agency[14].

As we can see, we considered a lot of funding sources from the beginning, but one of the purposes of the application we developed was to provide the opportunity to easily add other resources in the future.

3 State of the Art

3.1 Semantic Web

The specification and monitoring of conditional obligations and prohibitions with starting points and deadlines is a crucial aspect in the design of open interaction systems. In paper [1], the authors regard such obligations and prohibitions as cases of social commitment, and

[8] https://www.daad.ro/en/
[9] https://www.icr.ro/categorii/burse/en
[10] https://www.adrnordest.ro/en/homepage/
[11] https://nec.ro/fellowships/apply-for-a-fellowship
[12] https://mfe.gov.ro/pocu-apelul-de-proiecte-sprijin-pentru-doctoranzi-si-cercetatori-post-doctorat/
[13] https://fulbright.ro/
[14] https://roburse.ro/

propose to model them in OWL[15], the logical language recommended by the W3C for Semantic Web applications. They identified 2 main approaches:

- using an object oriented language as JAVA using OWL;
- using an object oriented language as JAVA using OWL with SWRL[16] to overcome certain expressiveness limitations of OWL.

They describe in this paper the second approach, and even though the result seems to be better than in other research programs in the same area, they ended up saying that this is still not a complete model until the complete OCeAN meta-model [2] will be formalized with SWL.

A creative solution to the problem of data collection is to reuse existing electronic records of social interaction that were not created for the purposes of network analysis in the first place. In paper [3], the authors use the basic method of Kautz in a slightly different way. Since their goal is the extraction of social networks, they are given a list of names, to begin with. They consult the search engine for investigating the possible tie between all pairs of names.

The number of queries required grows quadratically with the number of names, which is not only costly in terms of time but is limited by the number of queries that search engines allow. Ultimately, the system could have brought families and friends directly in contact after matching their records instead of waiting for them to search for each other.

3.2 Data Filtering

Although the study [4] concentrates on what web filtering is in general in regard to the different communication levels of the Internet (ex. Network layer, Transport layer, Application layer), it introduces a classification for filtering methods that may prove useful for our project, and that is:

[15]https://www.w3.org/OWL/

[16]https://www.w3.org/Submission/SWRL/

- Rating based filtering;
- Blacklisting;
- Keyword matching;
- Dynamic filtering.

Of these, dynamic filtering may be the most useful as it uses various statistical machine learning methods like Bayesian or KNN to understand the semantic content of the information to be filtered. The paper goes on and compares the enumerated filtering methods, discussing effectiveness and limitations, reaching the conclusion that the filtering system should be adapted to the system using it.

The study [5] shifts the focus of data importance from data origin to data correctness and discusses the topic of inconsistent data, like duplicate data, contradictory data, error codes, data outliers or missing data, altogether. The paper introduces a variety of complex solutions based largely on statistical analysis ranging from linear and polynomial interpolation to statistical model curve fitting.

3.3 Information Retrieval

The book [6] is an essential reference to issues and future directions in Information Retrieval. Information Retrieval (IR) can be defined as the process of representing, managing, searching, retrieving, and presenting the information. A more technical definition states that IR represents the process of matching a query against the information objects that are indexed. This book presents the theoretical underpinnings of the field of Information Retrieval as well as other subjects of interest such as:

- Web Information Retrieval: a chapter outlining distinctive characteristics of Web IR in order to show what makes it different, and also references relevant literature in the area.
- Context and Information Retrieval: this chapter talks about the importance of context information in handling challenges of information search and retrieval, and how it can be captured and

represented in order to build context-aware information systems and applications.

- The Role of Natural Language Processing in Information Retrieval: a chapter that presents various relevant NLP techniques and how they can be applied to Information Retrieval.
- Cross-Language Information Retrieval: as the name implies, this chapter addresses the problem of finding information in one language in response to queries expressed in another in IR.

The book [7] goes more into detail regarding Information Retrieval processes on the Web, as well as aspects of Web search. It touches subjects regarding Natural Language Processing in Information Retrieval, Link Analysis and Page Ranking, Semantic Search, Meta-search, and Multi-domain Search.

4 Proposed Solution

We aimed to create a Service Oriented application [8], [9], thus our architecture has three components: (1) An internal process that collects data from websites, (2) one that processes the project calls details, and (3) one that stores them in a database. The end-user will use a graphic interface (website/mobile app) in order to access our database stored data (different project calls from different websites). The main reason for our chosen architecture is to be able to give the opportunity to have a "hidden brain" that will collect data from the internet and to be able to provide this data in as many ways as possible, thus we expose some APIs in order to access them. That being said, we can show them in a website, in different mobile applications (Android/iOS), and we are basically able to have a cross platform graphic interface with the same data provider backend. Below, we will have a more in-depth look over each component.

4.1 Crawling Component

The crawling module stands at the forefront of the application and serves as a source of all the raw project data [10]. The main logic behind it is that we define a customizable list of project calls source websites to be explored periodically using, where necessary, a designated crawler (see Figure 1).

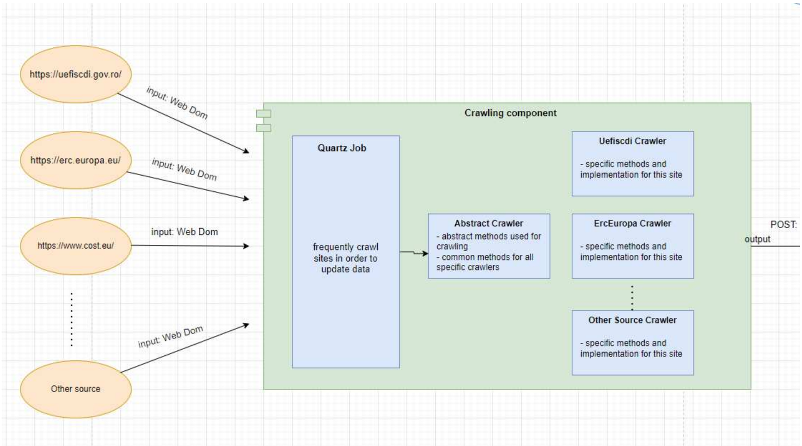


Figure 1. The Crawling Module

This recurrent search is done using an implementation of a cron-job (like the Quartz framework for Java) that will first try to call a generic crawler unless there exists a specific crawler defined, meant for that website. This implies that for any project call source website for which we need special crawling methods, a new class needs to be implemented. Thus, we first define an Abstract Crawler class meant to serve as a template for all other concrete crawler implementations, as well as future ones.

This raw project data is stored and/or updated with each recurrent search of the cron job, and a REST API makes this data accessible through a POST endpoint to the semantic analysis module for later processing.

4.2 Semantic Analysis Component

The semantic analysis module is an independent Java Spring Boot application meant to extract only the relevant information from the raw project input data [11], [12] (see Figure 2). This was done in order to make the development process easier, as well as streamline exposing a REST API of its own, to be accessed by the back-office module.

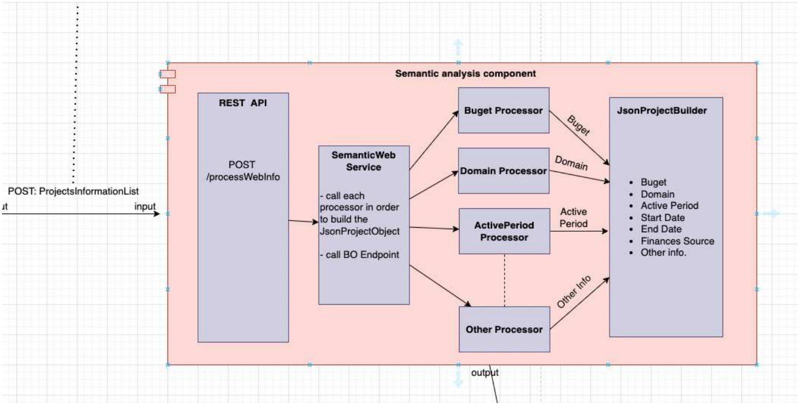


Figure 2. The Semantic Analysis Module

When the back-office module calls this API, several processors are put into motion to extract the project information. To accommodate for this, we define an abstract processor class, from which all other concrete processors derive, that contains two functions:

- searchKeywordsInProjectString;
- matchRegexInProjectString.

The two functions are similar in functionality, the difference being that one ensures support for custom concrete keywords to look for when searching the relevant information in the text, while the other ensures regular expression (regex) support to account for a variety of formatting differences.

As such, we have defined the following: *ActivePeriodProcessor*, *BudgetProcessor*, *DomainProcessor*, *EligibleActivitiesProcessor*, *EligibleApplicantsProcessor*, *FinancesProcessor*, *MoreInfoProcessor*, and *PurposeProcessor*. The results of all these processors are then used by a builder class to construct a *ProjectDetails* object that is the final result of this module.

4.3 Back-Office Component

This module comes in the form of an easily expandable OPEN API. The API exposes a series of endpoints with some being intended for internal use to facilitate communication between project modules, i.e., the POST endpoint that is called by the Semantic Analysis module to persist the processed data; and some intended for public use, i.e., the GET endpoints to be used by 3rd-party applications or UI modules [13] (see Figure 3 for details).

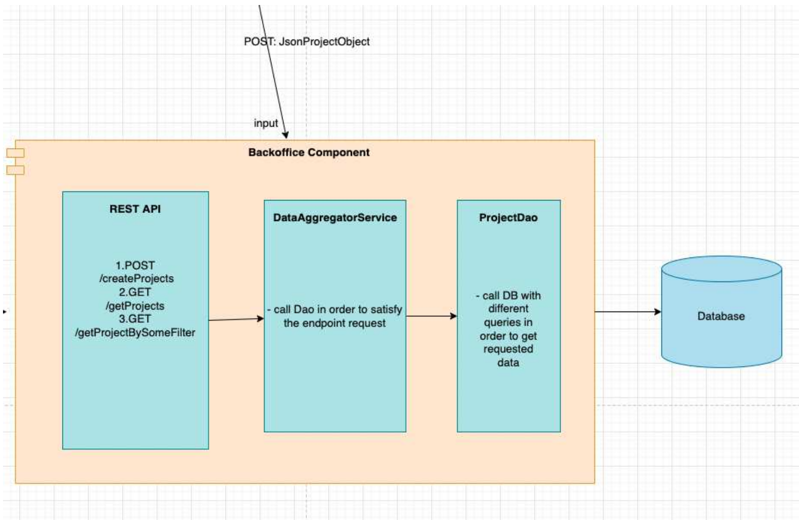


Figure 3. The Back-Office Component

The main purpose of the module is to handle the persistence and delivery of processed project call information. In order to do that, it implements a repository pattern and uses a cloud-based MongoDB database as a storage unit. For more technical information regarding the API, refer to the OPEN API specification [14] present in the GitHub repository [15].

5 Conclusion

In the end, we consider that the proposed solution presents an elegant and easy to build way to solve the problem. The fact that the application offered is higly modular makes it so that any of the components can evolve mostly independently or even be replaced as long as the cross component communication is mantained or updated accordingly.

In terms of future work we plan to continue to improve each module, as well as the way they communicate with each other. For the Crawling component, we plan to work on its adaptability by improving the ability of the general purpose crawler to collect relevant curated data and by designing more specialised crawlers. The Semantic Analysis module will benefit from an OWL integration for an increase in the accuracy of the information extracted for a project call. The improvements to the Backoffice and UI will mainly come in the form of security layers and integration of query language in the API, to facilitate and speed up the data acquisition process for the end user.

Acknowledgments. This work was supported by project REVERT (taRgeted thErapy for adVanced colorEctal canceR paTients), Grant Agreement number: 848098, H2020-SC1-BHC-2018-2020/H2020-SC1-2019-Two-Stage-RTD. Data processing and analysis in this paper were supported by the Competitiveness Operational Programme Romania, under project SMIS 124759 - RaaS-IS (Research as a Service Iasi).

References

[1] N. Fornara and M. Colombetti, "Ontology and Time Evolution of Obligations and Prohibitions Using Semantic Web Technology," in *International Workshop on Declarative Agent Languages and Technologies DALT 2009: Declarative Agent Languages and Technologies VII*, LNCS, vol. 5948, 2009, pp. 101–118.

[2] I. Lahoud, D. Monticolo, and S. Gomes, "OCEAN: A Semantic Web Service to Extract Knowledge in E-Groupwares," *2010 Sixth*

International Conference on Signal-Image Technology and Internet Based Systems, 2010, pp. 354–362.

[3] P. Mika, *Social Networks and the Semantic Web. Semantic Web and Beyond: Computing for Human Experience,* Springer, 2007.

[4] M.T. Banday and N.A. Shah, "A Concise Study of Web Filtering," *Sprouts: Working Papers on Information Systems*, vol. 10, no. 31, 2010, Article 352, 11 p. [Online]. Available: https://aisel.aisnet.org/sprouts_all/352.

[5] O. Wedin, J. Bogren, and I. Grabec, "Data Filtering Methods," EU Project Deliverable, Roadidea 215455, 2008.

[6] A. Goker and J. Davies, *Information Retrieval: Searching in the 21st Century,* Wiley Telecom, 2009.

[7] S. Ceri, A. Bozzon, M. Brambilla, E. Della Valle, P. Fraternali, and S. Quarteroni, *Web Information Retrieval. Data-Centric Systems and Applications,* Springer, 2013.

[8] M. Baboi, A. Iftene, and D. Gîfu, "Dynamic Microservices to Create Scalable and Fault Tolerance Architecture," in *23rd International Conference on Knowledge-Based and Intelligent Information* & *Engineering Systems, Procedia Computer Science, vol. 159*, 4-6 September, Budapest, Hungary, 2019, pp. 1035–1044.

[9] D. Alexandru, A. Iftene, and D. Gîfu, "Using New Technologies to Learn Programming Languages," in *28th International Conference on Information Systems Development (ISD2019)*, Toulon, France, August 28-30, 2019.

[10] C.G. Cușmuliuc, L. G. Coca, and A. Iftene, "Identifying Fake News on Twitter using Naive Bayes, SVM and Random Forest Distributed Algorithms," in *13th Edition of the International Conference on Linguistic Resources and Tools for Processing Romanian Language (ConsILR-2018)*, 2018, pp. 177–188.

[11] A.L. Gînscă, A. Popescu, M. Lupu, A. Iftene, and I. Kanellos, "Evaluating User Image Tagging Credibility," in *Experimental IR meets Multilinguality, Multimodality, and Interaction* (Lecture Notes in Computer Science, vol. 9283), Springer International Publishing, In Proceedings of 6th International Conference of the CLEF Association, CLEF'15 Toulouse, France, September 8-11, 2015, pp. 41–52.

[12] A. Iftene and A.M. Baboi, "Using semantic resources in image retrieval," in *20th International Conference on Knowledge Based and Intelligent Information and Engineering Systems, KES2016, vol. 96*, York, United Kingdom: Elsevier, 5-7 September 2016, pp. 436–445.

[13] C.S. Atodiresei, A. Tănăselea, and A. Iftene, "Identifying Fake News and Fake Users on Twitter," in *International Conference on Knowledge Based and Intelligent Information and Engineering Systems, KES2018* (Procedia Computer Science, vol. 126), 3-5 September 2018, Belgrade, Serbia, 2018, pp. 451–461.

[14] API: `https://github.com/mafteiervin1/Research-Projects-Applications/blob/main/docs/open\_api\_spec/backoffice\_spec.json`

[15] Project github: `https://github.com/mafteiervin1/Research-Projects-Applications`

Ervin Maftei[1,2], Cristian-Mihai Rosu[1,3],
Cristi-Constantin Rusu[1,4], Adrian Iftene[1]

Received September 01, 2022
Accepted October 07, 2022

[1] "Alexandru Ioan Cuza" University of Iasi, Faculty of Computer Science
[2] Endava Romania, Iasi, Romania
[3] Chargetrip SRL, Romania
[4] Continental Automotive Romania, Iasi, Romania
E–mail: {emaftei5, cristian.rosu453, cristirusu.99, adiftene}@gmail.com

E-course: developing a model for content generation

Alexandr Parahonco, Mircea Petic

Abstract

The intensive development of the computer and information industries made a huge leap forward, which influenced the emergence of new technologies and enormous sources of information. Nowadays most of e-courses represent text information that can be taken from the Internet. The article proposes the model of the application for creating dynamic content of training courses and discusses practical implementations for content parsing. Our solution is based on web-technologies and web approaches.

Keywords: E-learning, content generation, web scraping.

1 Introduction

The twenty-first century has attended the emergence of groundbreaking information technologies that brought changes in our life. Since the mid-1990s, the Internet gave a start to methods, tools, and gadgets that covered all academic disciplines and business sectors. Soon afterwards we witnessed a chain of web 2.0 technologies like E-commerce, which started social media platforms, E-Business, E-Learning, E-government, Cloud Computing, and more other in 2021 [1].

According to the analysis of Statista Company, 74 zettabytes (trillion gigabytes) of data were created in 2021. Compared to 2020, it has grown from 59 zettabytes, and from 41 zettabytes in 2019 [2]. This is accompanied by 1,903,642,574 websites of "controversial quality" of content; taking into account that information there is similar, somewhere outdated, or fake [3]. In these circumstances, work on content

©2023 by Alexandr Parahonco, Mircea Petic

creation and actualization becomes a problem of spending lots of resources. Especially it matters in the educational area, where qualitative and up-to-date content plays an important role in the specialist formation. Meanwhile, there are growing appeals to automate the process of creation educational content using both web technologies and web-approaches.

In our previous papers, we presented our solution [1,2] – a program model of an application for content generation that is built on top of the focused web crawler. In our first article on this topic, we discovered a way of getting information from the Internet by using a focused web crawler. Furthermore, the development of web crawlers led us to web scraping techniques, and we designed our own model for grabbing and "processing" data. That paper included, beyond the above research products, the realization of phase 1 (from the model) with the formation of synonymous networks for each word from the search query (see Fig. 1). Our next article covered a description of the phases 2–4. There, as obtained earlier, synonymous networks or search queries alone (by choice) are used to find web sites with the Google search engine. Our application succeeded in prepossessing searched data and exposing it in HTML format for user arrangement.

However, the central thesis of this paper is to provide an extended look at our technical solutions in content processing.

The article begins with an analysis of web-scraping techniques focused on fetching information from the web network. It discusses their role in modern life and ways of application. Then the scheme of the model for the dynamic creation of training courses is presented. Finally, the paper discusses content processing situations and their solutions.

[1] *The model of Web crawler for expansion the scope of initial search*, Proceedings of the Workshop on Intelligent Information Systems WIIS2020, December 04-05, 2020, Chisinau, Republic of Moldova.

[2] *Steps in content generation for e-courses*, International Conference "Mathematics & IT: Research and Education" (MITRE-2021), July 01 – 03, Chisinau, Republic of Moldova.

2 State of the art

E-learning, which is a direct result of the integration of technology and education, has become a powerful learning tool, especially through the use of Internet technologies. The undeniable importance of e-learning in education has led to a huge increase in the number of courses and e-learning systems offering various types of services. However, the majority of the current online-based learning systems have two serious drawbacks: 1) non-availability of ready content, which leads to a dead end for an instructor who begins to make up a course without the material to start up, and 2) the rapid changes in the educational content, the vast amount of published papers, and the ever-increasing training tutorials necessitate the existence of the dynamic update for the existing courses in E-learning system [4]. To solve all these problems, it is necessary to build some web systems or technologies that will take information from the internet and process it in some way.

The first articles about fetching information from the web network appeared in 2008. One of these papers was Udit Sajjanhar's. He throws light on educational content mined from university websites in the form of course pages. His system tries to learn the navigation path by observing the user's clicks on as few example searches as possible, and then uses the learnt model to automatically find the desired pages using as few redundant page fetches as possible. Following that, H.W. Hijazi and J.A. Itmazi use key words divided into two categories as a basis for launching web crawlers: included and excluded from the search query. The authors crawl websites of open educational resources (OER), mainly the Massachusetts Institute of Technology (MIT), which first announced plans to make all of its course materials freely available [4].

However, the idea of processing web content automatically, technically, came in 1993 with World Wide Web Wanderer, the first web robot, which sole purpose was to gauge the size of the web. Though, it gave the birth of the first concept of web-crawling. Soon afterwards, the first crawler-based web search engine, JumpStation, was developed. It built a new milestone in web-technologies – the prototype of Google, Bing, Yahoo, and other search engines on the web today. Since 2004, it acquired a new concept – visual web scraping, provided by BeautifulSoup HTML parser and Web Integration Platform version 6.0 [5]. This brought popularity to that technique. Many people, including researchers, started the use of web scraping in different domains.

During the ages, the concept of fetching information from the Internet has evolved into new technology – web-scraping (web data extraction). It includes two categories of techniques such as manual equipment (copy-paste) and automatic data scraping. Manual scraping involves copying and pasting web content, which takes a lot of effort and is highly repetitive in the way it is carried out. Automated scraping techniques shifts from HTML Parsing [6 – 7], DOM Parsing, XPath to Google Sheets, and Text Pattern Matching. Moreover, some semi-structured data query languages, such as XQuery and the HTQL, can be used to parse HTML pages and to retrieve and transform page content [8].

At present, there are 8 domains of web-scraping usage. A series of recent studies has indicated that most of them are employed in favor of commercial benefits and machine learning (general aspect). Thus, less attention is paid to content generation in the educational scope.

3 Program model for the dynamic creation of training courses

Our numerous studies brought us to the idea of dynamic content generation application. We started from the model of manual scraping and came finally to the program model represented in Fig. 1.

According to our approach, we have 6 phases. In phase 1, some web-crawlers create networks of synonymous words. In phase 2, our application uses the original request and/or their selected synonyms for advanced search using Google search. Next, in phase 3, we gain from Google links for the requests and process them (crawl, select necessary fragments), storing all the information in the database. According to phases 4 and 5, well-merged content should be generated and further exported in HTML or PDF formats in Phase 6.

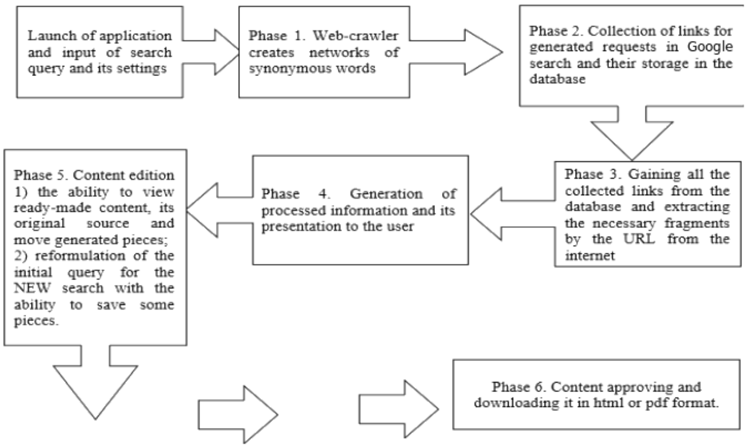


Figure 1. Scheme of the program model for the dynamic creation of training courses.

4 Content processing solutions

During the model realization, we encountered situations of content processing. Below, we present these situations and their solutions.

The most common task is related to the web-page structure handling. To create networks of synonyms, first, the structure of a predefined online dictionary was analyzed and a web scraping approach was selected (an example below is given for Wordsmyth The Premier Educational Dictionary-Thesaurus). Second, code implementation of the DOM parsing approach followed to get data from the structure in Fig. 2.

```
<tr class="definition">
    <td class ="title"> … </td>
    <td class ="data">
        <dl>
            <dt></dt>
            <dd></dd>
            ...
        </dl>
    </td>
</tr>
```

Figure 2. Dom structure from the dictionary.

In order to gain synonyms from the web page, a certain function from the application processes it, as shown in Fig. 3.

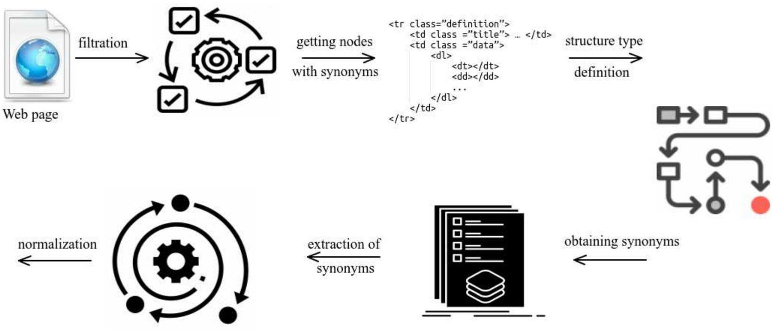


Figure 3. Extraction of synonyms.

In the beginning, it filters the whole DOM structure of the page to access meaningful nodes. Then it checks the existence of synonyms and similar words for the current word and, if so, the algorithm defines the case of the inner structure (e.g., the string may start with the word "Synonyms:" followed by synonyms with comma separation or even "Antonyms" with the same style) and, stores *these* words in the variable **$words**. Finally, it normalizes this list of synonyms (removes extra spaces and last comma from the string). In this way, synonyms are fetched from the structure (Fig. 2) and stored in the array $words.

Then another structure is created with obtaining only unique synonyms from the string:

```
if (!empty($words)) {
        $extended_search_query[] = [
        'index' => $index,
              'synonyms' => implode(',', array_unique(explode(','
                  ↪ , $words))),
              'original' => $word
       ];
}
```

This structured array binds every word ("original" parameter and its index) from the search request with its synonyms. In the next step, the obtained synonyms are turned into full-search query strings that help to find the demanded content on the internet (see Fig. 4).

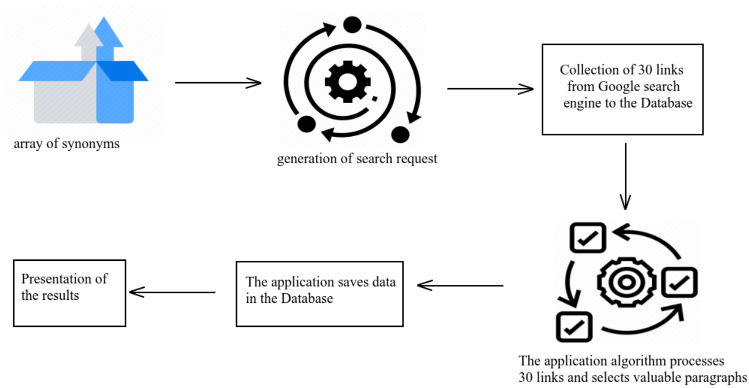


Figure 4. Fetching requested information from the Internet.

The *searchOriginal* method uses Google API to collect links from websites containing search queries and store them in a database.

Thereafter, the application fetches data from these links and selects only important sentences. In this case, the HTML Parsing strategy is applied. Thus, the application picks the sentences out from the web script based on the request query bound with the website in the database. Hence, some valuable information is received and delivered to the end user.

5 Conclusion

The findings of this study have demonstrated that E-learning environments suffer from the "cold start" problem – lack of prepared content for courses. Researchers propose to build a focused web crawler for intelligent data extraction from web sources. We have adopted this idea in our application model and enhanced it by widening keywords area by composing semantically-related network of word and search zone by

Google engine.

At the given moment, phases 3 – 4 – 5 endure development works. We need to process not only HTML pages, but Word and PDF documents. Moreover, user interface needs to be well thought and updated to let the teacher get generated content with ease. Another topic lays under Data mining and Text mining approaches implementation to give upon the generated content more logic.

Acknowledgments. This article was written as part of the research project 20.80009.5007.22 "Intelligent information systems for solving ill-structured problems, processing knowledge and big data".

References

[1] M. Yamin, "Information technologies of 21st century and their impact on the society," *International Journal of Information Technology*, vol. 11, no. 16-8, pp. 759–766, 2019. DOI: 10.1007/s41870-019-00355-1.

[2] D. Pavils, "How much data will the world produce in 2021?," 2021. [Online]. Available: https://www.cloverdx.com/blog/how-much-data-will-the-world-produce-in-2021.

[3] "Internet Live Stats. Total number of Websites," 2021. [Online]. Available: https://www.internetlivestats.com/total-number-of-websites/.

[4] A. Parahonco, M. Petic, and C. Negara, "The model of Web crawler for expansion the scope of initial search," in *Proceedings WIIS2020*, Chisinau, Moldova, 4-5 December, 2020, pp. 139–151. ISBN 978-9975-68-415-6.

[5] D. Kremer, "Brief History of Web Scraping," 2021. [Online]. Available: https://webscraper.io/blog/brief-history-of-web-scraping.

[6] Bo. Zhao, "Web Scraping," in *Encyclopedia of Big Data*, L. Schintler, C. L. McNeely, Eds. Springer International Publishing, 2017, pp. 1–3. DOI: 10.1007/978-3-319-32001-4_483-1.

[7] S. Ruihua, "Joint Optimization of Wrapper Generation and Template Detection," in *KDD'07: Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, August, 2007, pp. 894–902. DOI: 10.1007/978-1-4419-7735-9_4.

[8] "Introduction to Web Scraping," 2019. [Online]. Available: https://www.geeksforgeeks.org/introduction-to-web-scraping/.

Alexandr Parahonco, Mircea Petic

Received September 12, 2022
Accepted October 17, 2022

Alexandr Parahonco
Vladimir Andrunachievici Institute of Mathematics and Computer Science, MD-3100, Moldova
Department of Information Technologies, Alecu Russo State University of Balti, MD-3100, Moldova
E–mail: `alexandr.parahonco@usarb.md`

Mircea Petic
Vladimir Andrunachievici Institute of Mathematics and Computer Science, MD-3100, Moldova
Alecu Russo State University of Balti, MD-3100, Moldova
E–mail: `mircea.petic@math.md`

Knowledge Base Logics

Alexander Sakharov

Abstract

This paper describes a logical characterization of knowledge bases with non-Horn rules and facts. The proof theory is a sequent calculus with logical rules limited to the negation rules. Instances of this calculus are given by non-logical axioms expressing knowledge base rules and facts. One sub-calculus of this sequent calculus corresponds to knowledge bases with Horn rules and facts, and another one corresponds to inference without Reductio Ad Absurdum. The models are specified by constraints on truth values. Models for inference without Reductio Ad Absurdum are 3-valued whereas the other models are 2-valued.

Keywords: non-Horn knowledge base, sequent calculus, constrained model

1 Introduction

A great deal of research has been devoted to logical characterizations of emerging AI inference methods. Logical characterizations bring clarity to the methods. Semantics establishes the legitimacy of method results. Calculi make the results of these methods explainable. Besides, the apparatus of mathematical logic can be used to analyze and compare AI methods. Simpler characterizations are mostly sought after.

Inference from logic programs and knowledge bases (KB) is usually interpreted as inference in first-order logic (FOL). Nonetheless, inference from KBs and logic programs differs significantly from inference in FOL. The outcome of this inference is literals as opposed to arbitrary FOL formulas. In contrast to FOL axiom sets, the number of rules and especially the number of facts involved in KB inference may be huge. The interpretation of negation may be different from FOL [1].

©2023 by Alexander Sakharov

KBs and logic programs may include computable (aka evaluable) functions and predicates [2]. Reasoning by contradiction (aka Reductio Ad Absurdum) is not quite compatible with computable functions or predicates [3].

The languages of logic programs and KBs are simpler than that of FOL. Usually, KB facts are atoms or literals. Atoms are expressions $P(t_1, ..., t_k)$ where P is a predicate and $t_1, ..., t_k$ are terms. Literals are atoms or their negations. Usually, KB rules are expressions of the form $A \Leftarrow A_1 \wedge ... \wedge A_k$. In non-Horn rules, $A, A_1, ..., A_k$ are literals. In Horn rules, A and all A_i are atoms. In normal logic programs, A is an atom and A_i are literals.

The semantics of KB with Horn or non-Horn rules/facts is given by classical 2-valued FOL models for implications whose premises are the conjunction of KB facts and rules and whose conclusions are inference goals. FOL calculi are used as proof theories for Horn and non-Horn KBs. This paper adopts the terminology of [4]. FOL models and proof theories are not adequate for normal logic programs, their logical characterizations are nonmonotonic [5]. We consider KBs with standard negation because the concept of negation as failure leads to controversies [6]. Nonmonotonic systems are prone to erroneous derivations of negative literals for KBs or logic programs with evolving rule sets.

Much simpler model and proof theories for non-Horn KBs are presented here. The proof theory is a sequent calculus without the weakening rule. Its instances are given by non-logical axioms representing KB rules and facts. The negation rules are the only logical rules of this calculus. Its sub-calculi can be obtained by excluding the negation rules or the contraction rules. The first sub-culculus corresponds to Horn KBs, and the second one corresponds to direct inference from non-Horn KBs, i.e., inference without using Reductio Ad Absurdum. The models for the three calculi are determined by constraints induced by KB rules and facts. The models for non-Horn and Horn KBs are 2-valued and the models for direct inference from non-Horn KBs are 3-valued.

2 Sequent Calculi

A KB is called consistent if no atom is a fact or is derivable from this KB along with its negation being derivable or a fact. A literal is called ground if it does not contain variables. Let $-A$ denote the complement of literal A. The complement of an atom is the negation of this atom, and the complement of a negative literal is the atom of this literal.

A substitution is a finite mapping of variables to terms. These variables constitute the domain of the substitution, and the terms constitute its range. The result of applying a substitution θ to a symbolic expression A is the expression $A\theta$ obtained from A by replacing every occurrence of every variable in the domain of θ by the term with which it is associated. The result of applying a substitution to an expression is called its instance [7].

We rely on sequent calculi with non-logical axioms as an instrument for proof-theoretical characterization of KB inference. A sequent is $\Gamma \vdash \Pi$ where Γ is an antecedent and Π is a succedent. Consider a variant of Gentzen's *LK* [8] in which antecedents and succedents are multisets of formulas instead of sequences. *LK* will refer to this variant which does not require the exchange rule. *LK* has one logical axiom $A \vdash A$. The structural rules are cut, contraction, and weakening:

$$\frac{\Gamma \vdash A, \Delta \quad A, \Pi \vdash \Sigma}{\Gamma, \Pi \vdash \Delta, \Sigma}\ cut$$

$$\frac{A, A, \Gamma \vdash \Pi}{A, \Gamma \vdash \Pi}\ LC \qquad \frac{\Gamma \vdash A, A, \Pi}{\Gamma \vdash A, \Pi}\ RC$$

$$\frac{\Gamma \vdash \Pi}{A, \Gamma \vdash \Pi}\ LW \qquad \frac{\Gamma \vdash \Pi}{\Gamma \vdash A, \Pi}\ RW$$

KB inference and logic programming are concerned about the derivation of literals, i.e., sequents of the form $\vdash A$ in the terminology of sequent calculi where A is a literal. KB facts and rules can be treated as non-logical axioms [4]. Sequents of the form $\vdash A$ represent facts, and rules are represented by sequents of the form $A_1, ..., A_n \vdash A$ where $A, A_1, ..., A_n$ are literals. Variables can be replaced by ground terms in instances of these axioms.

Standard cut elimination techniques can be applied even in the presence of non-logical axioms [4]. Cut instances are moved upward so that one premise of every cut is a non-logical axiom. Therefore, the subformula property [4] holds for the formulas that are not literals. They have one of the forms $A \wedge B$, $A \vee B$, $A \Rightarrow B$, $\forall x A(x)$, $\exists x A(x)$, and $\neg C$ where C is not an atom. Hence, the logical rules for connectives $\wedge, \vee, \Rightarrow$ and for quantifiers are admissible in derivations of literals and so are all formulas except for literals. The only logical rules that are necessary in these derivations are the two negation rules:

$$\frac{\Gamma \vdash A, \Pi}{\neg A, \Gamma \vdash \Pi}\ L\neg \qquad\qquad \frac{A, \Gamma \vdash \Pi}{\Gamma \vdash \neg A, \Pi}\ R\neg$$

Lemma 1. *Weakening is admissible in derivations of literals in LK extended with non-logical axioms of the form $A_1, ..., A_n \vdash A$ ($n \geq 0$, $A, A_1, ..., A_n$ are literals) expressing rules and facts of a consistent KB.*

Proof. Consider a LK derivation with the endsequent $\vdash G$. The applications of the weakening rules can be moved below all other rules. We present relevant permutations for the cases that the weakening formula is principal in the following cut, contraction, or negation rule. The other cases are even simpler and left to the reader.

$$\dfrac{\Gamma \vdash A, \Pi \quad \dfrac{\Delta \vdash \Sigma}{A, \Delta \vdash \Sigma}}{\Gamma, \Delta \vdash \Pi, \Sigma} \quad \to \quad \dfrac{\dfrac{\Delta \vdash \Sigma}{\cdots}}{\Gamma, \Delta \vdash \Pi, \Sigma}$$

$$\dfrac{\dfrac{\Gamma \vdash \Pi}{\Gamma \vdash A, \Pi} \quad A, \Delta \vdash \Sigma}{\Gamma, \Delta \vdash \Pi, \Sigma} \quad \to \quad \dfrac{\dfrac{\Gamma \vdash \Pi}{\cdots}}{\Gamma, \Delta \vdash \Pi, \Sigma}$$

$$\dfrac{\dfrac{\Gamma \vdash \Pi, A}{\Gamma \vdash \Pi, A, A}}{\Gamma \vdash \Pi, A} \quad \to \qquad\qquad \dfrac{\dfrac{\Gamma, A \vdash \Pi}{\Gamma, A, A \vdash \Pi}}{\Gamma, A \vdash \Pi} \quad \to$$

$$\dfrac{\dfrac{\Gamma \vdash \Pi}{\Gamma \vdash \Pi, A}}{\Gamma, \neg A \vdash \Pi} \to \dfrac{\Gamma \vdash \Pi}{\Gamma, \neg A \vdash \Pi} \qquad\qquad \dfrac{\dfrac{\Gamma \vdash \Pi}{A, \Gamma \vdash \Pi}}{\Gamma \vdash \neg A, \Pi} \to \dfrac{\Gamma \vdash \Pi}{\Gamma \vdash \neg A, \Pi}$$

Weakening cannot be the last rule in a derivation of $\vdash G$ because sequent $\vdash$ is not derivable from consistent KBs. Hence, weakening can be eliminated from derivations of sequents like this one. □

Definition 1. *L_c is the set of sequent calculus instances in which formulas are literals, the structural rules are cut, LC, RC, the logical rules are $L\neg$ and $R\neg$ restricted to atoms as the principal formulas, the logical axiom is $A \vdash A$, and non-logical axioms represent KB rules and facts.*

Definition 2. *L_d is the set of sequent calculus instances in which formulas are literals, the structural rule is cut, the logical rules are $L\neg$ and $R\neg$ restricted to atoms as the principal formulas, the logical axiom is $A \vdash A$, and non-logical axioms represent KB rules and facts.*

Definition 3. *L_h is the set of sequent calculus instances in which formulas are atoms, the structural rule is cut, and non-logical axioms represent KB rules and facts.*

Since weakening and logical rules except for the negation rules are admissible in derivations of literals, L_c are complete for derivations of literals in FOL. L_h and L_d are sub-calculi of L_c. All these calculi are more lucid than calculi for FOL in its entirety.

L_h are de facto single-succedent calculi because the conclusion of every application of cut has one atom in the succedent. The negation rules are admissible in derivations of atoms, and negative literals are redundant in the absence of the negation rules. Clearly, axiom $A \vdash A$ would have been admissible in L_h. The contraction rule would have been admissible in L_h, too. The proof of the latter proposition is given after Theorem 1 below because it is basically a corollary of that theorem. Hence, L_h are complete for derivations of atoms from Horn KBs in FOL.

The implications $-A_i \Leftarrow -A \wedge A_1 \wedge ...A_{i-1} \wedge A_{i+1} ... \wedge A_n$ are called contrapositives to KB rule $A \Leftarrow A_1 \wedge ... \wedge A_k$. Backward chaining extended with KB rule contrapositives is a complete method for derivation of literals without using Reductio Ad Absurdum [3]. Any backward chaining step can be interpreted as an application of cut. Any contrapositive can be derived in L_d from its rule by applying the negation rules or the cut rule. For instance, contrapositive $B \Leftarrow \neg A \wedge C$ of rule $A \Leftarrow \neg B \wedge C$ is obtained by applying $L\neg$ to the sequent corresponding to this rule and applying cut to $\vdash B, \neg B$ and the conclusion of $L\neg$. Hence, L_d are complete for derivations of literals from non-Horn KBs without Reductio Ad Absurdum.

L_d are not intuitionistic calculi, they do not correspond to an intermediate logic either [4]. For instance, A is derivable in L_d from $\neg B$ and $\neg A \Rightarrow B$ but it is not intuitionistically derivable from the same. At the same time, negative literals can be proved by contradiction in intuitionistic logic. For example, $\neg P$ is intuitionistically derivable from $Q \Leftarrow P$ and $\neg Q \Leftarrow P$, but it is not derivable in L_d.

Single-succedent variants of L_c correspond to intuitionistic inference of literals. The respective calculi are not investigated here because considering literals is not sufficient for intuitionistic logic. Double-negated atoms should be considered too. They may be intuitionistically derivable whereas their respective atoms are not. Double-negated literals would be confusing for software developers who are not logicians.

Given that single-succedent calculi are problematic for languages limited to literals, there are two options for reducing the power of L_c: exclude the contraction rules or exclude the negation rules. The cut rule cannot be excluded from calculi with non-logical axioms, it plays the role of Modus Ponens in Hilbert-style calculi. It is fair to say that all interesting cases of inference of literals in FOL or its sub-logics are covered by L_c, L_h, L_d.

Usually, if a formal theory is inconsistent, then any formula is derivable in this theory. This is why inconsistent theories are discarded. KBs may have bugs and may be inconsistent. Any literal can be directly derived from sequents $\vdash A$ and $\vdash \neg A$ in a calculus with $L\neg$, cut, and weakening. Proliferation of inconsistencies is limited in calculi without the weakening rule. Nothing else could be derived from sequent $\vdash$ alone in L_h, L_d, L_c. Nonproliferation of inconsistencies is important in KB development because bugs do not lead to vast amounts of gibberish results in this case.

3 Constrained Models

Models are usually defined by truth tables (or functions) for logical connectives so that the truth values of ground formulas can be calculated. No other formulas than literals are produced during KB derivations. Because of this, legitimate models for KB inference can be defined by a negation truth function and by constraints on truth values in ground instances of facts and rules as opposed to truth tables for other connectives. Classical 2-valued models specify L_h and L_c. Let 1 represent true, -1 represent false, and $|A|$ denote the truth value of ground literal A. The following equation defines the truth values for negation: $|\neg A| = -|A|$.

Definition 4. *An assignment of truth values* $-1, 1$ *to ground atoms is a* $\mathcal{M}_h$ *model if the following constraints are satisfied:*
1. A *is a ground fact instance:* $|A| = 1$
2. $A_0 \Leftarrow A_1 \wedge ... \wedge A_k$ *is a ground rule instance:*
If $|A_i| = 1$ *for* $i = 1...k$, *then* $|A_0| = 1$.

Definition 5. *An assignment of truth values* $-1, 1$ *to ground literals is a* $\mathcal{M}_c$ *model if* $|\neg A| = -|A|$ *for any ground atom* A *and the following constraints are satisfied:*
1. A *is a ground fact instance:* $|A| = 1$
2. $A_0 \Leftarrow A_1 \wedge ... \wedge A_k$ *is a ground rule instance:*
$\max\{|A_0|, -|A_1|, ..., -|A_k|\} = 1$

Undefined truth values are represented by 0 in L_d models. Equation $|\neg A| = -|A|$ applies to these values as well. This is how negation truth values are defined in natural 3-valued logics [9]. Inference for multi-valued logics is often specified via nonstandard proof theories [10].

Definition 6. *An assignment of truth values* $-1, 0, 1$ *to ground literals is a* $\mathcal{M}_d$ *model if* $|\neg A| = -|A|$ *for any ground atom* A *and the following constraints are satisfied:*
1. A *is a ground fact instance:* $|A| = 1$
2. $A_0 \Leftarrow A_1 \wedge ... \wedge A_k$ *is a ground rule instance: If* $|A_i| = 1$ *for* $i = 1...k$, *then* $|A_0| = 1$.
3. $A_0 \Leftarrow A_1 \wedge ... \wedge A_k$ *is a ground rule instance: If* $|A_0| = -1$ *and* $|A_i| = 1$ *for* $i = 1...j-1$ *and* $i = j+1...k$, *then* $|A_j| = -1$.

Literal A is valid regarding $\mathcal{M}_c/\mathcal{M}_h/\mathcal{M}_d$ models if $|A'| = 1$ for all groundings A' of literal A in all respective models. The constraints of $\mathcal{M}_c/\mathcal{M}_h/\mathcal{M}_d$ can be reformulated for non-logical axioms of the respective calculi instead of KB rules and facts.

Theorem 1. *L_c inference is sound and complete with respect to $\mathcal{M}_c$ models for consistent KBs.*

Proof. Soundness. Consider a $\mathcal{M}_c$ model and an arbitrary grounding of a L_c derivation. We prove that the following property holds for any sequent $A_1, ..., A_k \vdash B_1, ..., B_n$, in this derivation:
1c. $\max\{|B_1|, ..., |B_n|, -|A_1|, ..., -|A_k|\} = 1$

Clearly, property 1c holds for the conclusion of the contraction rule and both negation rules if it holds for the premise. Suppose ground sequent $A_1, ..., A_k, C_1, ..., C_l \vdash B_1, ..., B_n, D_1, ..., D_m$ is the conclusion of the cut rule with premises $A_1, ..., A_k \vdash B_1, ..., B_n, E$ and $E, C_1, ..., C_l \vdash D_1, ..., D_m$, and property 1c holds for both premises. If $|E| = 1$, then $\max\{|D_1|, ..., |D_m|, -|C_1|, ..., -|C_l|\} = 1$. If $|E| = -1$, then $\max\{|B_1|, ..., |B_n|, -|A_1|, ..., -|A_k|\} = 1$. Property 1c holds for sequent $A \vdash A$ and for non-logical axioms. By induction on the depth of derivations, 1c holds for the endsequent of any ground L_c derivation. Hence, L_c derivations of literals are sound with respect to $\mathcal{M}_c$ models.

Completeness. Suppose G is a ground literal, $|G| = 1$ in all $\mathcal{M}_c$ models, and G is not derivable in L_c from KB facts and rules. Let $R_1, ..., R_r$ be the set of all KB rules and facts. Note that rules can be treated as clauses. Due to the deduction theorem, $R_1 \wedge ... \wedge R_r \Rightarrow G$ is valid in classical FOL. Therefore, there is a resolution refutation [7] of the set of clauses $R_1, ..., R_r, \neg G$. Since the KB is consistent, $R_1, ..., R_r$ is not refutable itself. Hence, G is used in the refutation in question.

Let us transform this refutation by discarding all resolution steps with $\neg G$ as one premise. If G is added to the descendant resolvents, then the entire resolution tree remains legitimate. The endclause becomes $G \vee ... \vee G$ instead of the empty clause. Let us propagate all substitutions upward in this resolution tree. Clauses $L_1 \vee ... \vee L_k$ can be treated as sequents $\vdash L_1, ..., L_k$. Clearly, sequents representing rules can be converted to this form by applying $R\neg$ and possibly cut with $\vdash A, \neg A$ as the other premise. Now, every resolution step with premises $A_1, ..., A_k, \neg C$ and $B_1, .., B_n, C$ is mapped to the following:

$$\frac{\vdash A_1, ..., A_k, \neg C \quad \dfrac{\vdash B_1, .., B_n, C}{\neg C \vdash B_1, .., B_n}}{\vdash A_1, ..., A_k, B_1, .., B_n}$$

Every factoring step corresponds to a contraction step in L_c. If the endclause contains more than one occurence of G, then the contraction rule is applied to it. The result is a L_c derivation of G. □

Suppose all $R_1, ..., R_r$ are Horn clauses. Since input resolution without factoring is complete for Horn KBs [7], there is such resolution refutation of $R_1, ..., R_r, \neg G$ that $\neg G$ occurs in the first resolution step only, and the factoring rule is not used in it. Clearly, this refutation corresponds to a L_h derivation, and hence, the contraction rule is admissible in derivations of atoms from Horn KBs.

Theorem 2. *L_d inference is sound and complete with respect to $\mathcal{M}_d$ models for consistent KBs.*

Proof. Soundness. Consider a $\mathcal{M}_d$ model and an arbitrary grounding of a L_d derivation. We prove that the following two properties hold for any sequent $A_1, ..., A_k \vdash B_1, ..., B_n$ in this derivation:

1d. If $|A_i| = 1$ for $i = 1...k$ and $|B_i| = -1$ for $i = 1...j-1$ and $i = j+1...k$, then $|B_j| = 1$.

2d. If $|B_i| = -1$ for $i = 1...k$ and $|A_i| = 1$ for $i = 1...j-1$ and $i = j+1...k$, then $|A_j| = -1$.

Suppose the premise of any of the two negation rules satisfies these properties. Clearly, these two properties hold for the conclusion in the cases that B_j from 1d is not the principal formula of $R\neg$ and A_j from 2d is not the principal formula of $L\neg$. If B_j from 1d is the principal formula of $R\neg$, then 1d holds for the conclusion because 2d holds for the premise. If A_j from 2d is the principal formula of $L\neg$, then 2d holds for the conclusion because 1d holds for the premise.

Suppose ground sequent $A_1, ..., A_k, C_1, ..., C_l \vdash B_1, ..., B_n, D_1, ..., D_m$ is the conclusion of the cut rule with premises $A_1, ..., A_k \vdash B_1, ..., B_n, E$ and $E, C_1, ..., C_l \vdash D_1, ..., D_m$, and properties 1d and 2d hold for both premises. Let $|A_i| = 1$ for $i = 1...k$, $|C_i| = 1$ for $i = 1...l$, $|B_i| = -1$ for $i = 1...j-1$ and $i = j+1...n$, $|D_i| = -1$ for $i = 1...m$. $|E| = -1$ due to 2d for the second premise. Consequently, $|B_j| = 1$ due to 1d for the first premise. If $|B_i| = -1$ for $i = 1...n$, $|D_i| = -1$ for $i = 1...j-1$ and $i = j+1...m$ instead, then $|E| = 1$ due to 1d for the first premise, and $|D_j| = 1$ due to 1d for the second premise.

Now let $|A_i| = 1$ for $i = 1...k$, $|C_i| = 1$ for $i = 1...j-1$ and $i = j+1...l$, $|B_i| = -1$ for $i = 1...n$, $|D_i| = -1$ for $i = 1...m$. $|E| = 1$ due to 1d for the first premise. Consequently, $|C_j| = -1$ due to 2d for the second premise. If $|A_i| = 1$ for $i = 1...j-1$ and $i = j+1...k$, $|C_i| = 1$ for $i = 1...l$ instead, then $|E| = -1$ due to 2d for the second premise, and $|A_j| = -1$ due to 2d for the second premise.

In all cases, both 1d and 2d hold for the conclusion. Properties 1d and 2d hold for sequent $A \vdash A$ and for non-logical axioms. By induction on the depth of derivations, the two properties hold for the endsequent of any L_d derivation. Hence, L_d derivations of literals are sound with respect to $\mathcal{M}_d$ models.

Completeness. Suppose A is a ground literal, $|A| = 1$ in all $\mathcal{M}_d$ models, and A is not derivable in L_d from KB facts and rules. Let us look at model M in which $|B| = 1$ for every ground literal B that is derivable from KB facts and rules, $|C| = -1$ for every such ground literal C that $\neg C$ is derivable, and $|D| = 0$ for every other ground literal D. Such model M exists for any consistent KB, and $|A| \neq 1$ in M.

Constraint 1 holds for M because ground instances of facts are derivable. If constraint 2 is violated for ground rule instance $A_0 \Leftarrow A_1 \wedge ... \wedge A_k$, then all sequents $\vdash A_i$ for $i = 1...k$ are derivable in L_d. Hence, A_0 is derivable from the latter by k applications of cut to $A_1, ..., A_k \vdash A_0$ and to every $\vdash A_i$ for $i = 1...k$. Hence, constraint 2 holds for this rule instance.

If constraint 3 is violated for ground rule instance $A_0 \Leftarrow A_1 \wedge ... \wedge A_k$, then all A_i for $i = 1...j-1$ and $i = j+1...k$ are derivable in L_d. $\neg A_0$ is also derivable. If both A_0 and A_j are atoms, then $A_1, ...A_{j-1}, A_{j+1}, ..., A_k, \neg A_0 \vdash \neg A_j$ is derivable by applying the negation rules to $A_1, ..., A_k \vdash A_0$ with A_0 and then A_j as the principal formula. If A_j is a negative literal, then cut is applied to $\vdash A_j, \neg A_j$ and $A_1, ..., A_k \vdash A_0$. Sequent $A_0, \neg A_0 \vdash$ is used as one premise of cut if A_0 is a negative literal. After that, $\vdash \neg A_j$ is derived by k applications of cut as it was done for constraint 2, and thus, constraint 3 holds for this rule instance. Therefore M is a $\mathcal{M}_d$ model and the assumption about A cannot be true. $\square$

Theorem 3. *L_h inference is sound and complete with respect to $\mathcal{M}_h$ models for consistent KBs.*

The proof of this theorem is similar but much simpler than the proof of Theorem 2, and thus it is left to the reader.

4 Related Work

Logics with more limited capabilities than FOL, e.g., description logics, are relevant to KB inference. Such logics also play an important role in argumentation. Simple logic has one inference rule (Modus Ponens) and no logical axioms [11]. Simple logic characterizes Horn KBs, it is similar to L_h but it is a Hilbert-style calculus. Direct derivations are defined in [12] as natural deduction done without using Reductio Ad Absurdum. These derivations are not limited to literals.

Instead of the FOL interpretation of KBs, the sets of rules with the same predicate in their heads can also be treated as inductive definitions, which implies that some fixed-point semantics is employed. The notation of such rules from FO(FD) [13] is based on FOL. The paper [13] considers inference in FO(FD) for finite domains only. Proof theories of inductive definitions as extensions of FOL are researched in [14] and [15]. Inference in these calculi is significantly more complicated than inference in FOL, it includes cyclic proofs. The practical value of these calculi with respect to KBs has not been investigated yet.

The semantics of normal logic programs are based on the concept of negation-as-failure [6]. Stable and well-founded semantics [1] are the most researched among these semantics. Inference procedures for stable and well-founded semantics [16], [17] are quite complicated. They have not been investigated as much as inference procedures for FOL and its fragments. Also, 3-valued models have been used for describing the semantics of logic programs [18], [19].

Sequent calculi without contraction have been investigated for decades [20]–[22]. Sometimes these calculi are referred to as affine logic. Direct predicate logic [23] is LK without the contraction rule. This calculus is equivalent to its single-succedent version LJ without contraction and with the axiom of double negation [22]. The latter is also known as stable logic [4]. Models for LK without contraction are studied in [20], [22].

The difference of L_h, L_c, L_d is that their instances contain non-logical axioms, and the cut rule is essential for these calculi. Usually, cut elimination is a central part of any investigation of sequent calculi. Non-logical axioms are an obstacle to the admissability of cut. Nonetheless, the cut rule can be pushed up to them. Sequent calculi with non-logical axioms are investigated in [4]. This investigation covers non-logical axioms in the form of so-called mathematical basic sequents. Non-logical axioms corresponding to KB rules/facts can be transformed to these sequents.

Sequent calculi for Horn and hereditary Harrop formulas are researched in [24]. The focusing calculi [25] characterize forward and backward chaining. These calculi are used for specifying nonstandard logics such as linear intuitionistic logic. A sequent calculus from [26] combines forward and backward chaining in linear logic.

Substructural logics may employ nonstandard logical connectives [27]. Non-Horn KBs rely on standard logical connectives. They are intuitive and well understood by software developers. In author's opinion, using nonstandard connectives in KBs that are developed and debugged by non-logicians is a recipe for problems, and those connectives should be avoided in KB rules and facts.

5 Conclusion

Allowing for equisatisfiability, non-Horn KBs are as expressive as FOL [7]. Nonetheless, they are representable in a language that is much simpler than the language of FOL. Non-Horn KBs are monotonic. Horn KBs are widely used in AI systems. They are the basis of the Prolog language.

Inference methods for both Horn and non-Horn KBs have been extensively investigated [5]. Efficient resolution methods [28] are fit for non-Horn KBs. Direct inference methods are more adequate for non-Horn KBs containing computable functions and predicates. These functions and predicates are a vital part of real AI systems. There exist efficient direct inference methods [3]. Horn KBs facilitate faster inference.

Defining models via constraints as opposed to truth functions for FOL formulas seems simpler and more suitable for KBs. Sequent calculi are considered the best tool for proof-theoretical investigations. Substructural sequent calculi are well-studied formal systems. Calculi L_h, L_d, L_c are comprehensible due to few logical rules and to the natural mapping of KB rules/facts to non-logical axioms. It is fair to say that various substructural sequent calculi whose logical rules are limited to the negation rules correspond to meaningful types of KBs.

References

[1] J. S. Schlipf, "The expressive powers of the logic programming semantics," *J. Comput. Syst. Sci.*, vol. 51, no. 1, pp. 64–86, 1995.

[2] W. McCune, "Otter 3.3 reference manual and guide," Argonne National Lab., Tech. Rep., 2003.

[3] A. Sakharov, "Inference methods for evaluable knowledge bases," in *Software Engineering Application in Informatics*, ser. Lecture Notes in Networks and Systems. Springer, 2021, pp. 499–510.

[4] S. Negri and J. Von Plato, *Structural proof theory.* Cambridge University Press, 2001.

[5] S. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach*, 3rd ed. Prentice Hall Press, 2009.

[6] M. Denecker, M. Truszczynski, and J. Vennekens, "About negation-as-failure and the informal semantics of logic programming," *Association for Logic Programming*, 2017.

[7] C.-L. Chang and R. C.-T. Lee, *Symbolic logic and mechanical theorem proving.* Academic press, 1973.

[8] M. E. Szabo, Ed., *The collected papers of Gerhard Gentzen.* North-Holland, 1969.

[9] A. Avron, "Natural 3-valued logics–characterization and proof theory," *The Journal of Symbolic Logic*, vol. 56, no. 1, pp. 276–294, 1991.

[10] S. Gottwald, *A treatise on many-valued logics.* Research Studies Press, 2001.

[11] P. Besnard and A. Hunter, "A review of argumentation based on deductive arguments," *Handbook of Formal Argumentation*, pp. 437–484, 2018.

[12] A. C. Kakas, P. Mancarella, and F. Toni, "On argumentation logic and propositional logic," *Studia Logica*, vol. 106, no. 2, pp. 237–279, 2018.

[13] P. Hou, B. De Cat, and M. Denecker, "FO (FD): Extending classical logic with rule-based fixpoint definitions," *Theory and Practice of Logic Programming*, vol. 10, no. 4-6, pp. 581–596, 2010.

[14] J. Brotherston and A. Simpson, "Sequent calculi for induction and infinite descent," *Journal of Logic and Computation*, vol. 21, no. 6, pp. 1177–1216, 2010.

[15] L. Cohen and R. N. S. Rowe, "Uniform inductive reasoning in transitive closure logic via infinite descent," in *27th EACSL Annual Conference on Computer Science Logic*, vol. 119. LIPICS, 2018, pp. 17:1–17:16.

[16] Y.-D. Shen, L.-Y. Yuan, and J.-H. You, "SLT-resolution for the well-founded semantics," *Journal of Automated Reasoning*, vol. 28, no. 1, pp. 53–97, 2002.

[17] S. Yamasaki and Y. Kurose, "A sound and complete procedure for a general logic program in non-floundering derivations with respect to the 3-valued stable model semantics," *Theoretical Computer Science*, vol. 266, no. 1-2, pp. 489–512, 2001.

[18] S. Hölldobler and C. D. P. K. Ramli, "Logic programs under three-valued Łukasiewicz semantics," in *International Conference on Logic Programming.* Springer, 2009, pp. 464–478.

[19] L. Naish, "A three-valued semantics for logic programmers," *Theory Pract. Log. Program.*, vol. 6, no. 5, pp. 509–538, 2006.

[20] V. N. Grishin, "Predicate and set-theoretic calculi based on logic without contractions," *Izvestiya Rossiiskoi Akademii Nauk. Seriya Matematicheskaya*, vol. 45, no. 1, pp. 47–68, 1981.

[21] R. Dyckhoff, "Contraction-free sequent calculi for intuitionistic logic," *The Journal of Symbolic Logic*, vol. 57, no. 3, pp. 795–807, 1992.

[22] H. Ono, "Logics without the contraction rule and residuated lattices," *Australasian Journal of Logic*, 2010.

[23] J. Ketonen and R. Weyhrauch, "A decidable fragment of predicate calculus," *Theoretical Computer Science*, vol. 32, no. 3, pp. 297–307, 1984.

[24] D. Miller, G. Nadathur, F. Pfenning, and A. Scedrov, "Uniform proofs as a foundation for logic programming," *Annals of Pure and Applied logic*, vol. 51, no. 1-2, pp. 125–157, 1991.

[25] K. Chaudhuri, F. Pfenning, and G. Price, "A logical characterization of forward and backward chaining in the inverse method," *Journal of Automated Reasoning*, vol. 40, no. 2-3, pp. 133–177, 2008.

[26] J. A. Harland, D. J. Pym, and M. Winikoff, "Forward and backward chaining in linear logic," *Electronic Notes in Theoretical Computer Science*, vol. 37, pp. 1–16, 2000.

[27] J.-Y. Girard, "Linear logic," *Theoretical computer science*, vol. 50, no. 1, pp. 1–101, 1987.

[28] L. Bachmair and H. Ganzinger, "Resolution theorem proving," in *Handbook of automated reasoning*. Elsevier, 2001, vol. 1, pp. 19–99.

Alexander Sakharov

Received September 29, 2022
Accepted October 19, 2022

Synstretch, Framingham, MA, USA
E–mail: `mail@sakharov.net`

A Formulation of Classical Higher-Order Logic in Isabelle/Pure

Jørgen Villadsen

Abstract

We describe a formulation of classical higher-order logic in Isabelle/Pure. It has successfully been used three times in a computer science course on automated reasoning and it allows students to study proofs of the Grandfather formula and Cantor's theorem as well as dozens of standard illustrative examples.

1 Introduction

Classical higher-order logic (namely second-order logic) goes back to Frege's *Begriffsschrift*. The modern formulation based on type theory goes back to Bertrand Russell (1908) and Alonzo Church (1940), with applications in mathematics, linguistics (e.g., Richard Montague 1973), and computer science (e.g., Mike Gordon 1988) [1].

A difficult example of formal reasoning, although in first-order logic, is described as follows on page 128 of the Handbook of Tableau Methods (Kluwer Academic Publishers 1999):

> *If every person that is not rich has a rich father, then some rich person must have a rich grandfather.*

The authors discuss a formalization with r (*rich*) and f (*father*):

$$\forall x(\neg r(x) \to r(f(x))) \to \exists x(r(x) \wedge r(f(f(x)))).$$

Proofs of the formula are discussed on pages 175–6, 181–3, and 193.

We consider first-order logic and its natural extension to higher-order logic as used at the Technical University of Denmark (DTU)

©2023 by Jørgen Villadsen

in the advanced BSc course Logical Systems and Logic Programming (`https://kurser.dtu.dk/course/02156`) and the MSc course Automated Reasoning (`https://kurser.dtu.dk/course/02256`). In the two courses we teach several other proof systems as well [2].

The Isabelle proof assistant by default starts in Isabelle/HOL [3]. We have recently described our approach to teaching intuitionistic and classical propositional logic using Isabelle/Pure [4]. In Isabelle/Pure we define the entire formulation of the logic ourselves, and the main contribution of the present paper is the extension of our formulation of propositional logic to higher-order logic.

In higher-order logic we can formally prove Cantor's theorem: For any set, there is a one-to-one correspondence of this set to its power set, and not vice versa (where ⊥ is a shorthand for *False* and *contr* is a convenient theorem):

```
theorem contr: ‹ ¬ p ⟷ p ⟹ q ›
proof −
  assume ‹ ¬ p ⟷ p ›
  have ‹ ¬ p ›
  proof
    assume p
    with ‹ ¬ p ⟷ p › have ‹ ¬ p › ..
    from this and ‹ p › show ⊥ ..
  qed
  with ‹ ¬ p ⟷ p › have p ..
  with ‹ ¬ p › have ⊥ ..
  then show q ..
qed

theorem Cantor: ‹ ¬ (∃f. ∀s :: 'a ⇒ bool. ∃x :: 'a. s = f x) ›
proof
  assume ‹ ∃f. ∀s :: 'a ⇒ bool. ∃x :: 'a. s = f x ›
  then obtain f where ‹ ∀s :: 'a ⇒ bool. ∃x :: 'a. s = f x › ..
  let ?D = ‹ λx. ¬ f x x ›
  from ‹ ∀s. ∃x. s = f x › have ‹ ∃x. ?D = f x › ..
  then obtain c where ‹ ?D = f c › ..
  from subst [of ?D] and this and refl have ‹ ¬ f c c ⟷ f c c › .
  with contr show ⊥ .
qed
```

This is a detailed proof in the so-called Isar language [5]–[8], which is the recommended style for large proofs in Isabelle/HOL. Isar stands for Intelligible Semi-Automated Reasoning and draws from both traditional mathematical texts and high-level programming languages. The proof can be substantially shortened using the automatic provers in Isabelle/HOL, but for students it is a good exercise to study the detailed proof. In fact, it uses just default proofs (the double period command ..) and, in a few places, immediate proofs (the single period command .). Default proofs involve a single elimination/introduction rule unambiguously given by the context. However, these features are hard to isolate and explain in the powerful Isabelle/HOL.

The Isabelle proof assistant is generic, which means that we do not need to work in Isabelle/HOL. For advanced mathematics we can work in Isabelle/ZF, axiomatic set theory, but we can also work in Isabelle/Pure which has no logic at all to start with [9]. The aim of the present paper is to describe our formulation of classical higher-order logic in Isabelle/Pure. In about a hundred lines we can provide the necessary proof infrastructure such that the formal proof of Cantor's theorem above can be mechanically checked — verbatim as shown.

The paper strives to be self-contained so consulting the online formalization is optional:

`https://hol.compute.dtu.dk/True/`

As mentioned, we use Isabelle in our logic courses [2]. We have successfully used the formulation of classical higher-order logic with online exam problems like the following well-known formula:

```
proposition ‹ (∀x. p x ⟶ q x) ∧ p c ⟶ q c ›
proof
  assume ‹ (∀x. p x ⟶ q x) ∧ p c ›
  then have ‹ ∀x. p x ⟶ q x › ..
  then have ‹ p c ⟶ q c › ..
  from ‹ (∀x. p x ⟶ q x) ∧ p c › have ‹ p c › ..
  with ‹ p c ⟶ q c › show ‹ q c › ..
qed
```

The paper is organized as follows. We continue with a discussion of the closest related work (§2). We describe first a formulation of intuitionistic higher-order logic with equality (§3) and then classical logic (§4). Finally, we conclude and discuss future work (§5).

2 Related Work

Makarius Wenzel's notable examples [10] for first-order logic and in particular higher-order logic have been the main starting point for our formulation. In the present paper, we have completely reworked the examples and made a lot of simplifications and clarifications. Makarius Wenzel has not published about the examples and they combine bits and pieces from very many logicians. However, only a few relevant papers are available on Isabelle/Pure and Isar [5]–[8]. Other uses of proof assistants in education have been considered, for example, in Coq [11] and Isabelle/HOL [4], [12].

3 Intuitionistic Logic

The following lines set up Isabelle/Pure such that *undefined* is the empty axiomatization for all types. Our only basic type is *bool* but other basic types can be introduced in a similar manner, e.g., a type *ind* for individuals, as is done in Isabelle/HOL.

axiomatization *undefined* :: *'a*

typedecl *bool*

judgment *Trueprop* :: ‹ *bool* ⇒ *prop* › (‹ - › *5*)

We start with reflexivity and substitutivity rules for equality.

axiomatization *Equality* (**infix** ‹ = › *50*)
  where *refl* [*intro*]: ‹ $x = x$ ›
    and *subst* [*elim*]: ‹ $x = y \Longrightarrow p\ x \Longrightarrow p\ y$ › **for** y :: *'a*

From equality we obtain bi-implication (elimination rules only; the introduction rule will be available in classical logic).

abbreviation *Iff* (**infixr** ‹ $\longleftrightarrow$ › *25*)
  where ‹ $p \longleftrightarrow q \equiv p = q$ › **for** p :: *bool*

lemma *Iff-E1* [*elim*]: ‹ $p \longleftrightarrow q \Longrightarrow p \Longrightarrow q$ ›
proof –
 assume ‹ $p \longleftrightarrow q$ › **and** p
 then show q **..**
qed

lemma *ssubst* [*elim*]: ‹ $x = y \Longrightarrow p\ y \Longrightarrow p\ x$ ›
proof –
 assume ‹ $x = y$ ›
 have ‹ $x = x$ › **..**
 with ‹ $x = y$ › **have** ‹ $y = x$ › **..**
 assume ‹ $p\ y$ ›
 with ‹ $y = x$ › **show** ‹ $p\ x$ › **..**
qed

lemma *Iff-E2* [*elim*]: ‹ $p \longleftrightarrow q \Longrightarrow q \Longrightarrow p$ ›
proof –
 assume ‹ $p \longleftrightarrow q$ › **and** q
 then show p **..**
qed

We can define truth using *undefined* (other definitions are possible).

definition *Truth* (‹ $\top$ ›) **where** ‹ $\top \equiv$ *undefined* $\longleftrightarrow$ *undefined* ›

lemma *Truth-I* [*intro*]: $\top$
 unfolding *Truth-def* **..**

Implication and universal quantification are next.

axiomatization *Imp* (**infixr** ‹ $\longrightarrow$ › *25*)
 where *Imp-I* [*intro*]: ‹ $(p \Longrightarrow q) \Longrightarrow p \longrightarrow q$ ›
  and *Imp-E* [*elim*]: ‹ $p \longrightarrow q \Longrightarrow p \Longrightarrow q$ ›

axiomatization *Uni* (**binder** ‹ $\forall$ › *10*)
 where *Uni-I* [*intro*]: ‹ $(\bigwedge x.\ p\ x) \Longrightarrow \forall x.\ p\ x$ ›
  and *Uni-E* [*elim*]: ‹ $\forall x.\ p\ x \Longrightarrow p\ c$ › **for** c :: *'a*

And existential quantification follows.

```
definition Exi (binder ‹ ∃ › 10)
  where ‹ ∃ x. p x ≡ ∀ r. (∀ x. p x ⟶ r) ⟶ r ›

lemma Exi-I [intro]: ‹ p c ⟹ ∃ x. p x ›
proof (unfold Exi-def)
  assume ‹ p c ›
  show ‹ ∀ r. (∀ x. p x ⟶ r) ⟶ r ›
  proof
    fix r
    show ‹ (∀ x. p x ⟶ r) ⟶ r ›
    proof
      assume ‹ ∀ x. p x ⟶ r ›
      then have ‹ p c ⟶ r › ..
      from this and ‹ p c › show r ..
    qed
  qed
qed

lemma Truth: ‹ ∃ p. p ›
proof
  show ⊤ ..
qed

lemma Exi-E [elim]: ‹ ∃ x. p x ⟹ (⋀x. p x ⟹ r) ⟹ r ›
proof (unfold Exi-def)
  assume ‹ ∀ r. (∀ x. p x ⟶ r) ⟶ r ›
  then have ‹ (∀ x. p x ⟶ r) ⟶ r › ..
  assume ‹ ⋀x. p x ⟹ r ›
  have ‹ ∀ x. p x ⟶ r ›
  proof
    fix x
    from ‹ ⋀x. p x ⟹ r › show ‹ p x ⟶ r › ..
  qed
  with ‹ (∀ x. p x ⟶ r) ⟶ r › show r ..
qed
```

Falsity and negation are straightforward.

```
definition Falsity (‹ ⊥ ›) where ‹ ⊥ ≡ ∀ p. p ›

lemma Falsity-E [elim]: ‹ ⊥ ⟹ p ›
```

```
proof (unfold Falsity-def)
  assume ‹∀p. p›
  then show p ..
qed

definition Neg (‹¬ -› [40] 40) where ‹Neg p ≡ p ⟶ ⊥›

lemma Neg-I [intro]: ‹(p ⟹ ⊥) ⟹ ¬ p›
  unfolding Neg-def ..

lemma Neg-E [elim]: ‹¬ p ⟹ p ⟹ q›
proof (unfold Neg-def)
  assume ‹p ⟶ ⊥› and p
  then have ⊥ ..
  then show q ..
qed
```

This is sufficient for Cantor's theorem, but of course we also define conjunction and disjunction.

First conjunction.

```
definition Con (infixr ‹∧› 35)
    where ‹p ∧ q ≡ ∀r. (p ⟶ q ⟶ r) ⟶ r›

lemma Con-I [intro]: ‹p ⟹ q ⟹ p ∧ q›
proof (unfold Con-def)
  assume p and q
  show ‹∀r. (p ⟶ q ⟶ r) ⟶ r›
  proof
    fix r
    show ‹(p ⟶ q ⟶ r) ⟶ r›
    proof
      assume ‹p ⟶ q ⟶ r›
      from this and ‹p› have ‹q ⟶ r› ..
      from this and ‹q› show r ..
    qed
  qed
qed

lemma Con-E1 [elim]: ‹p ∧ q ⟹ p›
proof (unfold Con-def)
```

```
  assume ‹ ∀ r. (p ⟶ q ⟶ r) ⟶ r ›
  then have ‹ (p ⟶ q ⟶ p) ⟶ p › ..
  have ‹ p ⟶ q ⟶ p ›
  proof
    assume p
    then show ‹ q ⟶ p › ..
  qed
  with ‹ (p ⟶ q ⟶ p) ⟶ p › show p ..
qed

lemma Con-E2 [elim]: ‹ p ∧ q ⟹ q ›
proof (unfold Con-def)
  assume ‹ ∀ r. (p ⟶ q ⟶ r) ⟶ r ›
  then have ‹ (p ⟶ q ⟶ q) ⟶ q › ..
  have ‹ p ⟶ q ⟶ q ›
  proof
    show ‹ q ⟶ q › ..
  qed
  with ‹ (p ⟶ q ⟶ q) ⟶ q › show q ..
qed
```

Then disjunction.

```
definition Dis (infixr ‹ ∨ › 30)
    where ‹ p ∨ q ≡ ∀ r. (p ⟶ r) ⟶ (q ⟶ r) ⟶ r ›

lemma Dis-I1 [intro]: ‹ p ⟹ p ∨ q ›
proof (unfold Dis-def)
  assume p
  show ‹ ∀ r. (p ⟶ r) ⟶ (q ⟶ r) ⟶ r ›
  proof
    fix r
    show ‹ (p ⟶ r) ⟶ (q ⟶ r) ⟶ r ›
    proof
      assume ‹ p ⟶ r ›
      from this and ‹ p › have r ..
      then show ‹ (q ⟶ r) ⟶ r › ..
    qed
  qed
qed
```

lemma *Dis-I2* [*intro*]: ‹ $q \Longrightarrow p \vee q$ ›
proof (*unfold Dis-def*)
  assume q
  show ‹ $\forall r.\ (p \longrightarrow r) \longrightarrow (q \longrightarrow r) \longrightarrow r$ ›
  proof
    fix r
    show ‹ $(p \longrightarrow r) \longrightarrow (q \longrightarrow r) \longrightarrow r$ ›
    proof
      show ‹ $(q \longrightarrow r) \longrightarrow r$ ›
      proof
        assume ‹ $q \longrightarrow r$ ›
        from *this* **and** ‹ q › **show** r **..**
      qed
    qed
  qed
qed

lemma *Dis-E* [*elim*]: ‹ $p \vee q \Longrightarrow (p \Longrightarrow r) \Longrightarrow (q \Longrightarrow r) \Longrightarrow r$ ›
proof (*unfold Dis-def*)
  assume ‹ $\forall r.\ (p \longrightarrow r) \longrightarrow (q \longrightarrow r) \longrightarrow r$ ›
  then have ‹ $(p \longrightarrow r) \longrightarrow (q \longrightarrow r) \longrightarrow r$ › **..**
  assume ‹ $p \Longrightarrow r$ ›
  then have ‹ $p \longrightarrow r$ › **..**
  with ‹ $(p \longrightarrow r) \longrightarrow (q \longrightarrow r) \longrightarrow r$ › **have** ‹ $(q \longrightarrow r) \longrightarrow r$ › **..**
  assume ‹ $q \Longrightarrow r$ ›
  then have ‹ $q \longrightarrow r$ › **..**
  with ‹ $(q \longrightarrow r) \longrightarrow r$ › **show** r **..**
qed

4 Classical Logic

We provide the axiom schemas for classical logic.

axiomatization
    where *Iff-I* [*intro*]: ‹ $(p \Longrightarrow q) \Longrightarrow (q \Longrightarrow p) \Longrightarrow p \longleftrightarrow q$ ›

axiomatization
    where *extensionality*: ‹ $(\bigwedge x.\ f\ x = g\ x) \Longrightarrow f = g$ › **for** f :: ‹ $'a \Rightarrow 'b$ ›

axiomatization *Eps* **where** *someI*: ‹ $p\ c \Longrightarrow p\ (Eps\ p)$ › **for** $c :: 'a$

Arguably, the first axiom schema or even the first and the second axiom schemas could be considered part of intuitionistic logic.

We have the following elegant but long proof of excluded middle.

```
theorem Diaconescu: ‹ p ∨ ¬ p ›
proof -
  let ?P1 = ‹ λq. p ∧ q ∨ ¬ q ›
  let ?P2 = ‹ λq. p ∧ ¬ q ∨ q ›
  have ‹ ¬ ⊥ › ..
  then have ‹ ?P1 ⊥ › ..
  with someI have ‹ ?P1 (Eps ?P1) › .
  have ⊤ ..
  then have ‹ ?P2 ⊤ › ..
  with someI have ‹ ?P2 (Eps ?P2) › .
  from ‹ ?P1 (Eps ?P1) › show ‹ p ∨ ¬ p ›
  proof
    assume ‹ p ∧ Eps ?P1 ›
    then have p ..
    then show ‹ p ∨ ¬ p › ..
  next
    assume ‹ ¬ Eps ?P1 ›
    from ‹ ?P2 (Eps ?P2) › show ‹ p ∨ ¬ p ›
    proof
      assume ‹ p ∧ ¬ Eps ?P2 ›
      then have p ..
      then show ‹ p ∨ ¬ p › ..
    next
      assume ‹ Eps ?P2 ›
      have ‹ ¬ ?P1 = ?P2 ›
      proof
        assume ‹ ?P1 = ?P2 ›
        from subst [of ?P1 ?P2] and this and refl
             have ‹ Eps ?P1 ⟷ Eps ?P2 › .
        from this and ‹ Eps ?P2 › have ‹ Eps ?P1 › ..
        with ‹ ¬ Eps ?P1 › show ⊥ ..
      qed
      have ‹ ¬ p ›
      proof
        assume p
        have ‹ ?P1 = ?P2 ›
        proof (rule extensionality)
```

```
          show ‹ ?P1 q ⟷ ?P2 q › for q
          proof
            assume ‹ ?P1 q ›
            then show ‹ ?P2 q ›
            proof
              assume ‹ ¬ q ›
              with ‹ p › have ‹ p ∧ ¬ q › ..
              then show ‹ p ∧ ¬ q ∨ q › ..
            next
              assume ‹ p ∧ q ›
              then have q ..
              then show ‹ p ∧ ¬ q ∨ q › ..
            qed
          next
            assume ‹ ?P2 q ›
            then show ‹ ?P1 q ›
            proof
              assume ‹ p ∧ ¬ q ›
              then have ‹ ¬ q › ..
              then show ‹ p ∧ q ∨ ¬ q › ..
            next
              assume q
              with ‹ p › have ‹ p ∧ q › ..
              then show ‹ p ∧ q ∨ ¬ q › ..
            qed
          qed
        qed
        with ‹ ¬ ?P1 = ?P2 › show ⊥ ..
      qed
      then show ‹ p ∨ ¬ p › ..
    qed
  qed
qed
```

The following classical corollary is often useful. Note that the classification as either theorem, corollary, lemma or proposition is just a convenience for humans but is ignored by Isabelle.

```
corollary classical: ‹ (¬ p ⟹ p) ⟹ p ›
proof -
  have ‹ p ⟹ p › .
```

```
  assume ‹ ¬ p ⟹ p ›
  with Diaconescu and ‹ p ⟹ p › show p ..
qed
```

Both Clavius's law and Peirce's law are excellent student exercises.

```
theorem Clavius: ‹ (¬ p ⟶ p) ⟶ p ›
proof
  assume ‹ ¬ p ⟶ p ›
  show p
  proof (rule classical)
    assume ‹ ¬ p ›
    with ‹ ¬ p ⟶ p › show p ..
  qed
qed
```

```
theorem Peirce: ‹ ((p ⟶ q) ⟶ p) ⟶ p ›
proof
  assume ‹ (p ⟶ q) ⟶ p ›
  show p
  proof (rule classical)
    assume ‹ ¬ p ›
    have ‹ p ⟶ q ›
    proof
      assume p
      with ‹ ¬ p › show q ..
    qed
    with ‹ (p ⟶ q) ⟶ p › show p ..
  qed
qed
```

The following *Boole* rule is useful for proofs in classical logic.

```
theorem Boole: ‹ (p ⟶ ⊥ ⟹ ⊥) ⟹ p ›
proof (rule classical [unfolded Neg-def])
  assume ‹ p ⟶ ⊥ ⟹ ⊥ › and ‹ p ⟶ ⊥ ›
  then have ⊥ .
  then show p ..
qed
```

The following rule for classical contradiction is a useful variant.

corollary *ccontr*: ‹ $(\neg\ p \Longrightarrow \bot) \Longrightarrow p$ ›
  using *Boole* [*folded Neg-def*] **.**

We continue with the well-known rules about double negation.

theorem *notnotD*: ‹ $\neg\ \neg\ p \Longrightarrow p$ ›
proof –
  assume ‹ $\neg\ \neg\ p$ ›
  show p
  proof (*rule ccontr*)
    assume ‹ $\neg\ p$ ›
    with ‹ $\neg\ \neg\ p$ › **show** $\bot$ **..**
  qed
qed

proposition ‹ $p \Longrightarrow \neg\ \neg\ p$ ›
proof
  assume ‹ $\neg\ p$ › **and** p
  then show $\bot$ **..**
qed

Finally, an alternative proof of Peirce's theorem (negation not used).

proposition ‹ $((p \longrightarrow q) \longrightarrow p) \longrightarrow p$ ›
proof
  assume ‹ $(p \longrightarrow q) \longrightarrow p$ ›
  show p
  proof (*rule Boole*)
    assume ‹ $p \longrightarrow \bot$ ›
    have ‹ $p \longrightarrow q$ ›
    proof
      assume p
      with ‹ $p \longrightarrow \bot$ › **have** $\bot$ **..**
      then show q **..**
    qed
    with ‹ $(p \longrightarrow q) \longrightarrow p$ › **have** p **..**
    with ‹ $p \longrightarrow \bot$ › **show** $\bot$ **..**
  qed
qed

5 Conclusions and Future Work

We have described our concise formulation of classical higher-order logic in Isabelle/Pure. It has successfully been used in our computer science course on automated reasoning and supplements our use of Isabelle/HOL to formalize various proof systems [4], [12].

For future work, we consider an online logic textbook using the approach presented here, including the details of our proof of the quite challenging Grandfather formula:

$$\forall x(\neg r(x) \to r(f(x))) \to \exists x(r(x) \land r(f(f(x)))).$$

An online textbook about programming language semantics [13] serves as inspiration, although it uses Isabelle/HOL and not Isabelle/Pure.

Acknowledgements Thanks to Asta Halkjær From, Frederik Krogsdal Jacobsen, Alexander Birch Jensen and Simon Tobias Lund for help. The anonymous reviewers provided very useful comments.

References

[1] W. M. Farmer, "The seven virtues of simple type theory," *Journal of Applied Logic*, vol. 6, no. 3, pp. 267–286, 2008.

[2] J. Villadsen and F. K. Jacobsen, "Using Isabelle in two courses on logic and automated reasoning," in *Formal Methods Teaching*, J. F. Ferreira, A. Mendes, and C. Menghi, Eds. Springer, 2021, pp. 117–132.

[3] T. Nipkow, L. C. Paulson, and M. Wenzel, *Isabelle/HOL - A Proof Assistant for Higher-Order Logic*, ser. Lecture Notes in Computer Science. Springer, 2002, vol. 2283.

[4] J. Villadsen, A. H. From, and P. Blackburn, "Teaching intuitionistic and classical propositional logic using Isabelle," in Proceedings 10th International Workshop on *Theorem Proving Components for*

Educational Software, ThEdu'21, (Remote) Carnegie Mellon University, Pittsburgh, PA, United States, 11 July 2021, ser. Electronic Proceedings in Theoretical Computer Science, J. Marcos, W. Neuper, and P. Quaresma, Eds., vol. 354. Open Publishing Association, 2022, pp. 71–85.

[5] M. Wenzel, "The Isabelle/Isar Reference Manual," 2021. [Online]. Available: https://isabelle.in.tum.de/documentation.html

[6] S. Berghofer and M. Wenzel, "Logic-free reasoning in Isabelle/Isar," in *Intelligent Computer Mathematics, 9th International Conference, AISC 2008, 15th Symposium, Calculemus 2008, 7th International Conference, MKM 2008, Birmingham, UK, July 28 - August 1, 2008. Proceedings*, ser. Lecture Notes in Computer Science, S. Autexier, J. A. Campbell, J. Rubio, V. Sorge, M. Suzuki, and F. Wiedijk, Eds., vol. 5144. Springer, 2008, pp. 355–369.

[7] L. C. Paulson, "The foundation of a generic theorem prover," *Journal of Automated Reasoning*, vol. 5, no. 3, pp. 363–397, 1989.

[8] M. Wenzel, "Isar - A generic interpretative approach to readable formal proof documents," in *Theorem Proving in Higher Order Logics, 12th International Conference, TPHOLs'99, Nice, France, September, 1999, Proceedings*, ser. Lecture Notes in Computer Science, Y. Bertot, G. Dowek, A. Hirschowitz, C. Paulin-Mohring, and L. Théry, Eds., vol. 1690. Springer, 1999, pp. 167–184.

[9] M. Wenzel, L. C. Paulson, and T. Nipkow, "The Isabelle framework," in *Theorem Proving in Higher Order Logics, 21st International Conference, TPHOLs 2008, Montreal, Canada, August 18-21, 2008. Proceedings*, ser. Lecture Notes in Computer Science, O. A. Mohamed, C. A. Muñoz, and S. Tahar, Eds., vol. 5170. Springer, 2008, pp. 33–38.

[10] M. Wenzel, "Notable examples in Isabelle/Pure," 2021. [Online]. Available: https://isabelle.in.tum.de/library/Pure/Pure-Examples/

[11] M. Henz and A. Hobor, "Teaching experience: Logic and formal methods with Coq," in *Certified Programs and Proofs - First International Conference, CPP 2011, Kenting, Taiwan, December 7-9, 2011. Proceedings*, ser. Lecture Notes in Computer Science, J. Jouannaud and Z. Shao, Eds., vol. 7086. Springer, 2011, pp. 199–215.

[12] A. H. From, J. Villadsen, and P. Blackburn, "Isabelle/HOL as a meta-language for teaching logic," in *Proceedings 9th International Workshop on Theorem Proving Components for Educational Software, ThEdu@IJCAR 2020, Paris, France, 29th June 2020*, ser. Electronic Proceedings in Theoretical Computer Science, P. Quaresma, W. Neuper, and J. Marcos, Eds., vol. 328. Open Publishing Association, 2020, pp. 18–34.

[13] T. Nipkow and G. Klein, *Concrete Semantics - With Isabelle/HOL*. Springer, 2014. [Online]. Available: http://concrete-semantics.org/

Jørgen Villadsen

Received September 29, 2022
Accepted October 21, 2022

Technical University of Denmark
E–mail: `jovi@dtu.dk`

Triplum: a Semantic Triples Symbolism for Logic of Natural Languages (Essay)

Ioachim Drugus

Abstract

The symbolism *Triplum* described in this paper can serve as a first draft for a W3C standard linking natural languages with Semantic Web (SW). The term *semantic triple* used in SW refers to any three formal expressions playing roles of *subject*, *predicate* and *object* in natural language sentences. Triplum is a symbolism of semantic triples obtained from taking over from SW's Notation3 the "triple format" and adding means for coding arbitrary clauses rather than only the Notation3 statements.

Keywords: logic of natural languages, computational linguistics, clause, semantic web.

1 Introduction

The Semantic Web (SW), sometimes known as Web 3.0, is an extension of the World Wide Web compliant with special standards set by the World Wide Web Consortium (W3C). The goal of the SW is to make all the Internet data machine-readable ([1]). Separate items of Internet data are referenced as *resources* and various resources are identified by IRIs (Internationalized Resource Identifiers) and described by the use of various symbolisms, like Notation3 (N3, [2]) which have been designed in conformity with RDF (Resource Description Framework, [3]).

The formal description in SW of a piece of reality is a dataset named *ontology*. The N3 and other normative standards associated with it that are widely used in SW, prescribe using *triples* of syntactic objects as atomary elements of such datasets to be processed by software. In

©2023 by Ioachim Drugus

this paper, the three syntactic objects used in a triple are referenced, for short, as “names”. An N3 ontology is essentially a set, i.e., an unordered structure of such triples, and this complies with the term “dataset”.

Notice that in SW, an ontology can also be represented as a set of quadruples, named also “quads” (see [4], and section 4). Notice that datasets of quads are significantly easier to use than datasets of triples. However, the 4^{th} member of a quad can be treated as a label of a triple, and, thus, one can work only with triples, labeled or unlabeled, as in the current paper.

In literature about SW (not in SW’s standards), one can find the term “*semantic triple*”, where the modifier “semantic” suggests treating such formal objects as *interpretable* expressions, in contrast with the RDF triples, which are compliant with SW standards, and are destined to be processed by software. Here is the widely accepted definition of *semantic triple* concept ([5]):

A *semantic triple*, or *RDF triple* or simply *triple*, is the atomic data entity in the Resource Description Framework data model. As its name indicates, a triple is a set of three entities that codifies a statement about semantic data in the form of subject-predicate-object expressions (e.g., "Bob is 35", or "Bob knows John").

The above definition confines the use of “semantic triple” term to SW’s normative standard RDF’s triples. However, the use of modifier “semantic” presupposes *interpretability* and not only processabilty by software. Thus, the term *semantic triple* describes more complex objects than the term *triple*. One can say that semantic triples, about which this paper is, “live” in a reality that, alongside the software dimension of SW, also has a human dimension (since one usually attributes to humans the ability to interpret). With such treatment adopted in the current paper, the term *semantic triple* could find its use also in domains like linguistics, logic of natural languages, and philosophy.

Notice, that the above definition actually admits the treatment of the two terms, *triple* and *semantic triple* as semantically having nothing in common since it says that a *triple codifies* (it does not say “is”) a “statement about semantic data in the form of subject-predicate-object expressions”. In other words, it says that RDF’s triples are to be treated as *codes* of the *semantic triples*.

The code of an object may have nothing in common with the object, and thus, one cannot consider the RDF triples as a sub-class of semantic triples. RDF triples are nothing else but codes, as said in the above definition.

There does not look like any framework has been proposed in the literature for semantic triples, and the current paper is an attempt in this direction. Triplum is not precisely specified as a symbolism in this paper – only the features of a framework that permits the development of such symbolisms outlined described. This is why the current paper is said “Essay”.

The word *triplum* is taken over from music where this is used for a poliphonic three voices composition or only for the third voice in such a composition.

A short description of Triplum to be further developed as a precize framework can be formulated like this:

Triplum is a knowledge presentation symbolism which offers symbolic devices to encrypt knowledge as datasets of unlabeled or labeled triples of names, where a “name” is treated as an interpretable syntactic object, and where knowledge is treated in the widest sense: statements, questions, exclamations, instructions and possibly sentences of other types, are treated as vehicles of knowledge.

This description can be treated as a blueprint of a symbolism to be developed, but for convenience of presentation, going forward the term “Triplum” is used as the name of an already developed symbolism.

From the above description, one can see that whereas N3 is a symbolism for presentation of *ontologies*, Triplum is (i.e., is intended to be) as a symbolism for presentation of a body of *knowledge*, where the term “knowledge” is treated as more general than the term *ontology*. More details on this follow next.

An ontology is a body of knowledge encrypted in a natural language text where only declarative sentences are used. One can say that N3 is specialized on presentation of knowledge formulated via statements, “declarative knowledge”. Probably, all sciences provide knowledge of this type, and all scientific knowledge can be presented as ontologies. However, formulations of open mathematical problems also provide a kind of knowledge to the community of mathematicians. In more general terms, the questions are also “vehicles of knowledge” (same as statements). A good source of various knowledge types conveyed by natural languages is the book [6].

Triplum is intended to serve for representing knowledge of all possible types and is treated here as a symbolism for *logic of natural languages* domain (see [7], classification code 03B65) to link the natural languages with SW.

It sounds appropriate to profit from N3’s symbolic practices and take from it over those essential features which are convenient for wide use, but give up the features which restrict the use to SW domain in order to make sure that what is enriched is the essential part of it. Thus, one can say that Triplum is obtained from N3 by “extraction and enrichment” method. Basically, what is “extracted” from N3 is the “triple format”, whereas the manner in how the “extracted matter” is “enriched” consists in adding new symbols for setting a type to a triple – declarative, interrogative, imperative, exclamative, etc. This method of obtaining Triplum from N3 reminds inheritance of the object-oriented programming and is so referenced next.

2 The Main Features of Triplum inherited from Notation3

Recall that in N3, the term “triple” is used to reference any sequence of three formal expressions which play the same roles as *subject*, *predicate*, and *object* in a natural language sentence. The subject and object must be regarded as a noun, and the predicate as a verb. Such a triple, written in the order the roles are mentioned above, with a period in the end is treated as a symbolic notation of a *declarative* sentence referenced as *statement*.

The denotation mechanism of N3 using triples proved to be sufficiently intuitive in the practice of ontology authoring, but the requirement to use only IRIs or abbreviations of IRI rather than customary names, confines the use of this mechanism to SW domain and is cumbersome or completely impractical in other domains like logic, philosophy, or informal discourse. Any intuitive symbolism which takes over the triple format of N3 must allow using reasonably arbitrary convenient names and this has been the first motivation for designing the Triplum symbolism. In order to understand why using the IRIs outside SW is a problem, consider the example of a triple which one would write in one line of text like this

$$A \textit{ implies } B,$$

but which in N3 would be encrypted as a triple with a predicate that looks like this (see [5]):

`<http://www.w3.org/2000/10/swap/log#implies>`.

The subject A and object B above must be also IRIs, and the N3 triple would take several lines of text – at least three lines (if each of the 3 IRIs is short enough to fit on one line).

The example above shows that N3 is unusable outside SW. In an informal discourse, it must be allowed to use resonably arbitrary names – natural language words or expressions, symbols or long formulas. Complete freedom in choosing a name cannot be allowed (in any symbolism) – certain limitations must be imposed by the rules of forming the names. This is why the expression "*reasonably* arbitrary" have been used above.

The larger freedom in using names with Triplum symbolism does not exclude using IRIs. If the simplification of N3 consisting only in increasing the freedom of introducing names would have been the single intention motivating development of the new symbolism, then Triplum could have been treated as the next version of N3. But there is a more important motivation for introducing Triplum: this symbolism is intended for encrypting sentences of arbitrary type, not only those of the declarative type.

There was no need to use the modifier *resonably arbitrary* of the word "type" in the end of previous paragraph, because the grammar of English names only four types of sentences:

- declarative sentence (statement)
- interrogative sentence (question)
- imperative sentence (command)
- exclamative sentence (exclamation)

N3 was designed specifically for encrypting declarative sentences and this symbolism does not offer any device for dealing with other sentence types. However, the type of a sentence reflects the manner how the reader must treat the sentence (see also below the discussion about "modality"), and in practice many manners (possibly, infinitely many) may be needed.

Various kinds of "queries" in SW encrypt natural language interrogative type of sentences – questions. SW offers a special symbolism named "SPARQL" for queries/questions. No types of sentences different from statements and queries can be encrypted by using SW's symbolisms.

However, the *atomic* interrogative, imperative, and exclamatory sentences can be also easily formulated in triple format. Therefore, one can expect that various symbolisms will be obviously designed in the oncoming Web 3.0 for each type, not only of declarative and interrogative ones. Triplum is intended here as a "root" for all such symbolisms.

One can treat the use of a period in the end of a triple as the N3's feature indicating the *declarative* type of the sentence encrypted by triple. Triplum inherits this feature and enriches the N3 symbolism by permission to use also other punctuation marks symbolizing various types of triples. Such additional marks can be "?" for questions and "!" for exclamations. For other types of sentences, the appropriate signs could be found in other languages or these can be selected from the wide set of Unicode characters.

The triple format enriched with reasonably arbitrary names permitted in triples and the linguistic devices for indicating a type from an enriched typeset which, alongside the declarative type, contains also other types, are two main features that Triplum inherits from N3.

3 Enouncements

A sentence of any type can be treated as result of application of syntactic *typifying* operator, which adds a punctuation mark (".", "?", "!", ...) in the end of sentence as described in previous section. The syntactic object to which such typifying operator applies is referenced here as "enouncement" (probably, the term "enounciation" can be also used here). A sentence is obtained by applying a typifying operator to an enouncement.

Whereas a triple in N3 *must* be followed by a period symbolyzing the declarative type, a triple of Triplum may not be necessarily followed by a punctuation mark. Thus, Triplum is both about sentences of various types and about enouncements out of which the sentences are made up.

Having introduced the concept *enouncement*, a terminology appropriate for it is required and is given next. For a sentence S of a certain type t there is an enouncement E to which the typifying operator T is applied – it adds a punctuation mark symbolizing the type t after E (one can say that the operator "ascribes a type"), and the result of this manipulation is S. The enouncement E is referenced here as "*content* of the sentence S".

In grammar, a sentence is said to express an idea, but this sounds correct only when one discusses about declarative sentences (one usually does not say that a question or exclamation expresses an idea). According to the approach of this paper, *the content* of a sentence expresses an idea, but the sentence expresses something larger: an idea and the manner how idea should be treated by the reader.

The typifying operator applied to an enouncement E adds a punctuation mark prescribing to readers a certain manner of treating enouncement E – as statement, question, or something else. Notice, that the typifying operator applies to *inscriptions* of enouncements, not to enouncements as abstract objects.

To examplify the concepts described above, consider the text:

$$Run?Run! \tag{1}$$

This text consists of two sentences, the content of which is the same but the modalities ascribed to them are different. One can state that two things are “the same”, and these “two things” are the two inscriptions of “Run”.

In semantics, the manner how the reader should treat the content expressed by a sentence is said “modality”. Therefore, the typifying operator can be treated as a syntactic operator which composes the inscription of an application of the “*modality* operator”.

The punctuation marks used to represent the modality operator (i.e., “used to express modality”) are said here “modality punctuation marks”. These are “.”, “?”, “!” and spanish-style punctuation marks “¿”, “?”, “¡”, “¿” used in pairs.

Notice, that this terminology is appropriate and convenient in discussions about *any* sentences, not only simple ones, with one predicate, usually named “clauses”. It is clauses which are represented as a triples (see section 7).

4 On Triple Format

One may wonder if the term “triple format” for N3 is correctly chosen given that the period in N3 or other punctuation signs symbolizing type in Triplum looks like “4^{th} member of a triple”. The Triplum’s semantics described in section 5 rules out such mis-interpretations. Consider this term referring only to the content of a sentence, and accordingly treat the expression “triple format of sentence S” as “triple format of S’s content”.

The term “triple format” is customarily treated as referencing an arrangement of three names in a sequence written from left to right. One can use a term more specific for this kind of arrangement: “left-right triple format”. This term could be used when discussing about N3 since it’s in this order that the terms in N3’s statements are arranged. For various purposes, different arrangements may be convenient: ”left-right” or “right-left”, and (for vertical arrangements) “up-down” or “down-up”, all of which can be referenced as “*positional* presentations” of a triple.

To avoid invoking the “arrangements”, the term “triple format” can be treated in a more abstract manner. Namely, *subject*, *predicate*, and *object* can be treated as *roles*, whereas triple format as a "cast of roles" (see [8]), which mathematically can be treated as a function that maps these roles to names. A presentation of a triple by indicating an entity on a role like this "*role: entity*" is referenced here as "*role-cast* presentation".

As an example of the two manners of representing a triple, consider the sentence "John runs fast." The *positional* representation specific for the Notation3 looks like this:

John runs fast .

Role-cast presentation of this sentence' content looks like this:

$$\texttt{subject: } John; \texttt{predicate: } runs; \texttt{object: } fast \tag{2}$$

or, for more "readibility", like this:

$$\texttt{subject: } John \tag{3}$$

$$\texttt{predicate: } runs \tag{4}$$

$$\texttt{object: } fast \tag{5}$$

Each of the two presentations has its merits and disadvantages. To name one of the difficulties coming up with role-cast presentation, one has to use a device for indicating that three syntactic objects (3), (4), (5) form one triple. Such a device can be as follows: enclose them with brackets of any kind, like the braces "{", "}" . In representing a (full) sentence (not only its content, an "enouncement"), for economizing symbols, one can use Spanish-style pairs of punctuation marks like "¿", "?", "¡", "¿", rather than braces.

5 On Descriptions

Since *description logic* (see [9]) is positioned in the logical foundations of SW, and since Triplum is destined to link SW with natural languages, the concept of *description* sounds as most appropriate to be positioned in foundation of Triplum's semantics.

The truth of the statement that a meta-language of a language L *describes* the language L, or of the statement that grammatical terms *describe* the objects of language, whereas words or expressions of a language *describe* objects in the World, can hardly raise any doubts. Therefore, the relation "X is described by Y" is a fundamental relation of linguistics, not only of SW.

Thus, the representation of a triple as (2) or as (3-5) are *descriptions* pertaining to description logic. A triple in any presentation is also such a *description*.

The modality punctuation marks used in Triplum will be treated here as Triplum's meta-symbols which describe the manner how the readers must interpret the content of sentences they read.

The conceptuality of descriptions can be adapted to phenomena within natural languages. Consider, for example, an enouncement encrypted as a triple like this:

$$\texttt{subject: } S; \texttt{predicate: } P; \texttt{object: } O \tag{6}$$

This looks as a set of notations, but is actually a role-cast presentation of an enouncement triple (!), and this shows the strength of this type of presentation.

A positional presentation of the above triple looks like this,

$$S \;\; P \;\; O \tag{7}$$

but the members of the triple can be associated in two manners, like this:

$$(S \;\; P) \;\; O \tag{8}$$

or like this:

$$S \;\; (P \;\; O) \tag{9}$$

Treating $(X\ Y)$ as denotation of phrase "X (is) described by Y", one can find two manners to interprete a triple:

1. The subject *described* by the predicate is a *description*, which is (further) *described* (better "complemented") by the object.

2. The subject is *described* by the predicate which (in turn) is *described* (better "modified") by the object.

These two manners of interpretation of a triple reflect the two manners of how the concept *predicate* is defined in natural languages grammars.

The relation *meta* between a language L and its meta-language M can be put in the phrase: "L is *described* by M". The *description* relation is a more general relation than the relation *meta*. However, since in a triple "a description is described" (see (8), where "$(S\ P)$ is described by O"), one can say that the triple "models" the relation meta. This hints to the logical reason why the triple (and not, say, the ordered pair) was chosen as the "*atomic* data entity in RDF data model" (see definition [5]). This reason is that the semantic triple models the relation *meta*. Some kind of a "semantic *ordered pair*" chosen for the role of "semantic atom" would be "too small" to model the relation "meta", which is a fundamental relation in semantics, and more generally, in linguistics.

6 Triplum's Vocabulary

A convenient term to refer to a word or stable expression of a natural language is *vocabula*; obviously, *vocabulary* is the most convenient term for the set of all vocabulas. In practice, "vocabulary" is a widely used term, whereas "vocabula" is rarely used in English. In this paper, however, vocabulas are treated as fundamental entities used to build a discourse in any language, natural or artificial, and vocabulary is treated as dependent on the used locale, slang or language. This is why both terms are frequently used in Triplum.

When used in domains different from SW, Triplum can limit itself with vocabulas, which are strings in an alphabet of a natural language, but when used in SW, Triplum's vocabulary must be convenient for encoding any natural language. A vocabula of Triplum used in SW must contain any sequence of Unicode characters. However, such a sequence with white space inside it would create ambiguity when used in a triple. To avoid such ambiguities, the Triplum vocabulas will be Unicode sequences enclosed between angular brackets "⟨", "⟩", but not containing these characters in other positions. This would permit to use natural language multiword expressions within Triplum vocabulas, and this is convenient for some practices.

Triplum permits using various vocabularies, which means that Triplum is a framework permitting specifying various symbolisms based on different vocabularies, one vocabulary defining one symbolism. The term "*Triplum symbolism* based on vocabulary V" is used here to reference a symbolism using vocabulas from V in its triples.

In Notation3, there are vocabulas named "blank nodes", which are used as "temporary names" of un-named entities. Such vocabulas correspond to expressions like "something", "something else", "another thing". Whereas, one can invent a rather limited number of such expressions in English, in Notation3 one can use an unlimited number of blank nodes in Notation3. These look like this "$\_ : S$", where S is string of characters without white space.

To allow for Notation3's blank nodes, Triplum must have a device for extending its vocabulary. Such a device could be provided by permission to use also "qualified names" in a Triple symbolism S based on a vocabulary V, i.e., names written in the format "$Q : N$", where Q is the name of a vocabulary and N is a vocabula in V. Notice, that in blank nodes used in Notation3, the underscore "_" can be treated as the name of the vocabulary of blanks nodes, and this would completely comply with Triplum's device of extending the vocabulary of a symbolism.

7 Triplum as a Symbolism of Clauses

Some analysis is presented below to support the thesis that in linguistics of natural languages, Triplum can be treated as a symbolism of clauses.

Clause is a unit of grammatical organization situated in rank next below *sentence*. If one treats a spoken (or written) discourse as vocalization (respectively, an inscription) of a train of thoughts, then the clauses are to be treated as separate cars of this train.

In natural languages grammars, a clause is treated as consisting of a *subject*, which could be missing but then it's implied, and a mandatory *predicate*. According to widely used terminology, a predicate could be missing only in "*incomplete* sentences"; what is called "sentence" must have a predicate.

The subject is the thing *about* which is a clause, and it must be treated as an existing entity since a clause "about nothing" does not have any use in communication. Even though the grammar of some natural languages permits the subject to be missing in a sentence (and still the latter is not referenced as an "incomplete sentence"), this is always treated as implied.

The grammar of English accepts the term "clause" to be used both for a proper part of a sentence and a standalone sentence with exactly one predicate. This means that for reference to a clause which is a proper part of a sentence, a special term is needed and here the term "*embedded* clause" is used.

Of the other parts which may be used in a natural language sentence (in English, eight ones are customarily distinguished as "main parts"), in this paper only the "direct complement" referenced here as *object* is considered.

Despite the fact that in grammar other formulations and wordings are used, one can see from the above that the linguists, consciously or unconsciouly, are using the conceptuality of Triplum framework.

8 Conclusion

The conceptuality named Triplum presented in this paper can serve as

- a unifying framework for various symbolisms used in SW, like Notation3, Turtle, N-Triple as well as those formal languages which are based on these;

- a formal framework to be used in the "logic of natural languages", a subject of the mathematical logic;

- a link between the SW symbolisms and the grammars of natural languages.

References

[1] *Semantic Web* (Definition). https://en.wikipedia.org/wiki/Semantic_Web. Retrieved on: November 15, 2022.

[2] *Notation3 (N3): A readable RDF syntax.* (W3C standard) https://www.w3.org/TeamSubmission/n3/. Retrieved on: November 15, 2022.

[3] *RDF 1.1 Concepts and Abstract Syntax* (W3C standard). https://www.w3.org/TR/rdf11-concepts/ Retrieved on: November 15, 2022.

[4] *RDF 1.1 N-Quads (W3C standard).* https://www.w3.org/TR/n-quads/ Retrieved on: November 15, 2022.

[5] *Semantic Triple* (Definition). https://en.wikipedia.org/wiki/Semantic_Triple Retrieved on: November 15, 2022.

[6] I. Ciardelli, J. Groenendijk, F. Roelofsen, *Inquisitive Semantics*, Oxford Univ. Press. 2018.

[7] *Mathematical Subjects Classification* (MSC2020). https://mathscinet.ams.org/mathscinet/msc/msc2020.html Retrieved on: November 15, 2022.

[8] T. E. Payne, *Describing morphosyntax: A guide for field linguists.* Cambridge; New York: Cambridge University Press, 1997.

[9] F. Baader et al., *The Description Logic Handbook: Theory, Implementation, Applications.* Cambridge University Press, Cambridge, UK, 2003.

Ioachim Drugus

Received December 7, 2022
Accepted December 12, 2022

Vladimir Andrunachievici Institute of Mathematics and Computer Science
E–mail: ioachim.drugus@math.md