Learning from Examples and Counterexamples using Nominal Logic

Daniele Nantes-Sobrinho

Imperial College & UnB

Abstract

Equational problems are fundamental in computer science, frequently arising as subproblems in areas such as program analysis and learning from examples and counterexamples. This talk examines equational problems in languages with binding operators, formulating them within the nominal framework as Nominal Equational Problems (NEP). Solutions to NEPs are obtained by applying simplification rules defined within nominal logic.

A key focus of the talk is the role of generalisation, a core operation for systems that learn from background knowledge and can itself be specified as an NEP. By leveraging anti-unification techniques, such systems can discover reusable program fragments by computing a least general generalisation (lgg) – sometimes called the most specific generalisation – for a collection of terms. This provides a principled basis for structured learning from examples.

Verifying Progress in Cyclic Reasoning (and in Research)

Liron Cohen

Ben-Gurion University

Abstract

Cyclic reasoning, in which induction is managed implicitly, is a promising approach to automatic verification. The soundness of cyclic proof graphs is ensured by checking them against a trace-based Infinite Descent property. While deciding Infinite Descent is PSPACE-complete in general, its cost varies sharply with two natural parameters: the number of graph vertices and the vertex-width that bounds the components tracked simultaneously. To gain speed in practice, one can trade completeness for efficiency. Cyclone follows this strategy: it layers several fast, but necessarily incomplete, semi-decision procedures for Infinite Descent atop a complete exponential fallback. These alternative algorithms are developed and benchmarked, and Cyclone is shown to deliver substantial runtime gains on real workloads. This talk outlines the theoretical landscape as well as the practical improvements and their empirical impact. We also briefly reflect on how we, as researchers, can use Infinite Descent for managing the inherently (often frustrating) cyclic nature of research itself.

Automata and Logics over Nested Data

Adriana Baldacchino[a]

[a] *University of Oxford*

Abstract

Automata and logics over infinite data are widely studied due to their effectiveness as tools for verification, database theory and programming language semantics. We focus on automata and logics over *nested data*, which additionally model parent-child relationships between data values. Existing models focus on bounded depth nested data. We extend the ideas to unbounded depth nested data, introducing two new models of nested data automata. Furthermore, we relate these models to well-structured transition systems to prove decidability of emptiness of these automata, and explore their closure properties.

1 Introduction

Automata over infinite alphabets are extensions of classical automata, whose input consists of *data words*. These words consist of symbols (a, d) where $a \in \Sigma$ is a letter from a finite set, and d is the data value, which comes from some infinite set $\mathcal{D}$. These automata and their corresponding logics are widely studied in computer science, as the unbounded set $\mathcal{D}$ allows us to model various sets that arise naturally. This includes modelling attribute values in database theory, unbounded agents in verification and fresh variables in programming language semantics.

Our focus is on *class memory automata* (CMA) [2], which for each data value $d \in \mathcal{D}$, keeps track of the state at which it last saw d. This extra information is used when deciding the next transition. Furthermore, a CMA has a set of locally and globally accepting states, and it accepts a word if and only if the current state is globally accepting, and the data was last seen in locally accepting states. We further study *nested data class memory automata* (NDCMA) [5], which are a refinement of the class memory automata described above. These operate over *nested data*, first introduced in [1], which is data with an additional hierarchical structure. We consider the *weak* variant of these automata, where the set of locally accepting states is the set of all states, as this is necessary for decidability of emptiness.

Nested data can be expressed in different ways. One way (as used in [1,6]) is to consider words over the alphabet $\Sigma \times \mathcal{D}^k$ for a fixed k. Another interpretation (used in [5]) is to endow $\mathcal{D}$ itself with a forest structure by considering a predecessor relation $\mathrm{pred} : \mathcal{D} \rightharpoonup \mathcal{D}$. These representations are equivalent as there are effective translations from one representation to the other, as shown in [5]. In our work we make use of the forest presentation, as our techniques follow those in [5].

Logics over nested data quickly become undecidable. Even with only two layers of nested data, two-variable first-order logic equipped with $x = y + 1$ and $x < y$ is undecidable. This was shown by a translation to multicounter automata [1]. In this same paper, it was shown that restricting to only the successor function results in a decidable logic, by a translation to shuffle expressions. A similar positive result was established in [6], where the authors established ND-LTL, a temporal logic on nested data words extending LTL by navigation along data values, which has a decidable satisfiability question. To do this, they introduce *prefix-closed nested data automata (pNDA)*, which turn out to be equivalent to weak NDCMA [5].

A key limitation to the existing models of automata over nested data is that the nesting is limited by some bound k. In the tuple representation, this is because each tuple is of size k, and for the forest representation this limitation arises as a bound for the depth of the forest structure of $\mathcal{D}$. Therefore, an interesting question to ask is whether this limitation can be lifted without introducing undecidability to the automata. We present new models on nested data extending NDCMA, which allow for parsing of data sets of unbounded depth and additionally have a decidable non-emptiness problem.

Given that NDCMA have been applied to obtain decidability results in the context of programming languages [3,4], our hope is that our extension could unlock further decidability results in the study of programming languages, as it would enable us to model unbounded creation of names, such as in loops. To aid in this pursuit, and to understand these models further, we study further closure properties of these automata, such as closure under complement, intersection and union. The talk will summarise the results of the author's master's project, which is supervised by Andrzej Murawski.

2 Defining Automata Models over Unbounded Nested Data

We now present the ideas behind two new models of unbounded-depth NDCMA. Note that to allow new data values to be available at any level, we require our forest to be infinitely full, that is there are infinite roots, and each node has an infinite number of children. Our automata need a finite description, so given some data value $d \in \mathcal{D}$ with an arbitrary number of predecessors $\mathrm{pred}(d), \mathrm{pred}^2(d), \ldots, \mathrm{pred}^n(d)$ we need to find a way to only consider some subset of a fixed size k of these predecessors.

One way we can do this is by matching with any subsequence of predecessors of d. Essentially, our transition function is of the form $\Delta = \bigcup_{i=0}^{k} \Delta_i$ where

$$\Delta_i : Q \times \Sigma \times Q \times (Q \cup \{\perp\})^i \rightharpoonup \mathcal{P}(Q).$$

Then, given a data value d and the current labelling function $f : \mathcal{D} \to Q \cup \{\perp\}$, a transition $q \xrightarrow{a,p,p_1,\ldots,p_n} q'$ is possible if d is labelled with p, and the path $d = d_0, d_1, \ldots, d_k = r$ from d to the root contains a subsequence $d_{m_1}, d_{m_2}, \ldots, d_{m_n}$ which has the labels $p_1, \ldots, p_n$. We call such NDCMA *memoryless NDCMA*, as a node does not discriminate between any of its predecessors.

While memoryless NDCMA are well-defined models for handling unbounded nested data, they are a bit unwieldy to use. This is mainly because even if the automata is deterministic in the 'usual' sense, meaning the image of Δ_i is made up of singletons, a word can still have multiple possible runs due to the nondeterminism in picking the subsequence of predecessors.

To allow for more control, we define a different model, referred to as k-memory NDCMA. As the name suggests, in this model each node has a bounded size memory. Whenever a new data value d is encountered, its memory is initialised based on the memory of its parent and the current state. Then, if we see d later on in the word, we can decide how to transition based on the label of d, and the label of the nodes in its memory. For this definition to work, we require that the memory of a new node is a suffix of the memory of the parent, and bound the maximum size of the memory by k.

3 Closure Properties of Unbounded NDCMA

These automata both admit a decidable non-emptiness problem. Our technique to prove this extends that of [5], which reduces non-emptiness of NDCMA to the coverability problem for *well-structured transition systems* (WSTS) [8]. These are sets, equipped additionally with a transition relation $\to$ and well-quasi order $\leq$ satisfying an *upward-compatibility* property.

If the order $\leq$ is decidable, and our WSTS additionally has an *effective pred-basis*, then the coverability problem is decidable. The original proof makes use of the tree homomorphism ordering, which is a well-quasi ordering over bounded depth trees. This is not a well-quasi ordering for unbounded trees, so it cannot be used analogously for our unbounded NDCMA. Our two automata models require different approaches. For memoryless NDCMA, we utilise the homeomorphic embedding ordering, which is a well-quasi order on all trees by Kruskal's tree theorem [7]. To handle k-memory NDCMA, we essentially extract a k-depth tree from the memory of the nodes – allowing us to use the tree homomorphism ordering.

Theorem 3.1 *The problems of emptiness of memoryless NDCMA and emptiness of k-memory NDCMA are reducible to the covering problem for a well-structured transition system with an effective pred-basis and decidable $\leq$, hence are decidable.*

Furthermore, we look at closure properties of these automata. The positive results about memoryless NDCMA can be obtained using the typical power construction. It follows that they are not closed under complement from the fact that CMA are not closed under complement.

Lemma 3.2 *Memoryless NDCMA are closed under union and intersection, but not closed under intersection.*

For k-memory NDCMA it is not as straightforward. These NDCMA are closed under union, using a non-deterministic construction, however they are not closed under intersection as we can in fact obtain the following undecidability result using a reduction to Turing machines.

Theorem 3.3 *Given two (deterministic) k-memory NDCMA $\mathcal{A}$, $\mathcal{B}$, the problem of deciding if $\mathcal{L}(\mathcal{A}) \cap \mathcal{L}(\mathcal{B}) = \varnothing$ is undecidable.*

The theorem has the following consequences:

Corollary 3.4 (i) *k-memory NDCMA are not closed under intersection.*

(ii) *Given two deterministic k-memory NDCMA $\mathcal{A}$, $\mathcal{B}$, it is undecidable whether $\mathcal{L}(\mathcal{A}) \subseteq \mathcal{L}(\mathcal{B})$.*

(iii) *Given two k-memory NDCMA $\mathcal{A}$, $\mathcal{B}$, it is undecidable whether $\mathcal{L}(\mathcal{A}) = \mathcal{L}(\mathcal{B})$.*

It is still open whether it is decidable if $\mathcal{L}(\mathcal{A}) = \mathcal{L}(\mathcal{B})$ given two *deterministic* k-memory NDCMA $\mathcal{A}$, $\mathcal{B}$. We summarise the closure results in the following table:

	memoryless NDCMA	k-memory NDCMA	det. k-memory NDCMA	det. NDCMA [5]
L^c	X	X	✓	✓
$L_1 \cap L_2$	✓	X	X	✓
$L_1 \cup L_2$	✓	✓	X	✓
$L_1 \subseteq L_2$	X	X	X	✓
$L_1 = L_2$	X	X	?	✓

References

[1] Björklund, H., Bojańczyk, M.: Shuffle Expressions and Words with Nested Data. Mathematical Foundations of Computer Science **4708**, 750–761 (2007), https://doi.org/10.1007/978-3-540-74456-6_66

[2] Björklund, H., Schwentick, T.: On Notions of Regularity for Data Languages. Theoretical Computer Science **411**(4), 702–715 (2007), https://doi.org/10.1016/j.tcs.2009.10.009

[3] Bunting, B. and Murawski, A. S.: Contextual Equivalence for State and Control via Nested Data. Proceedings of the 39th Annual ACM/IEEE Symposium on Logic in Computer Science, LICS '24. (19) (2024), https://doi.org/10.1145/3661814.3662109

[4] Cotton-Barratt, C. and Hopkins, D. and Murawski, A. S. and Ong, C.-H. L.: Fragments of ML decidable by nested data class memory automata. International Conference on Foundations of Software Science and Computation Structures. 249–263 (2015), https://doi.org/10.1007/978-3-662-46678-0_16

[5] Cotton-Barratt, C., Murawski, A. S., Ong, C.-H. L.: Weak and Nested Class Memory Automata. Language and Automata Theory and Applications **8977**(14), 188–199 (2015), https://doi.org/10.1007/978-3-319-15579-1_14

[6] Decker, N., Habermehl, P., Leucker, M., Thoma, D.: Ordered Navigation on Multi-attributed Data Words. CONCUR 2014. Lecture Notes in Computer Science, **8704**. 497–511 (2014), https://doi.org/10.1007/978-3-662-44584-6_34

[7] Kruskal, J. B.: Well-Quasi-Ordering, The Tree Theorem, and Vazsonyi's Conjecture. Transactions of the American Mathematical Society, **95**(2), 210–225 (1960), https://doi.org/10.2307/1993287

[8] Schmitz, S. and Schnoebelen, P.: Algorithmic Aspects of WQO Theory. Lecture Notes (2012), https://cel.hal.science/cel-00727025v2/file/lecturenotes.pdf

Growing a Modular Framework for Modal Systems: HOLMS

Antonella Bilotta[a] Marco Maggesi[b] Cosimo Perini Brogi[c]

[a] *Scuola Normale Superiore di Pisa*

[b] *Università di Firenze*

[c] *Scuola IMT Alti Studi Lucca*

Abstract

We present **HOLMS** (**HOL** Light Library for **M**odal **S**ystems), an evolving modular framework for mechanising modal reasoning within the HOL Light proof assistant. Building on earlier work on Gödel-Löb logic (GL), HOLMS introduces a compositional architecture to formalise modal adequacy proofs and implement automated decision procedures for various normal modal systems, currently including K, T, K4, and GL. To clarify the compositional nature of our framework and illustrate how it bridges general-purpose proof assistants, enriched sequent calculi, and formalised mathematics, we highlight some design choices and structural features of HOLMS, such as its use of the metalanguage, embedding strategies, and modularity metrics.

1 Introduction

Modal Logic. Modal logic extends classical logic by going beyond the true/false dichotomy and introducing operators that reflect non-truth-functional linguistic constructs, such as necessity, possibility, knowledge, belief, or obligation. Its versatility spans domains as diverse as computer science, linguistics, and normative reasoning. Various modal operators have proven capable of applications ranging from knowledge representation and verification of consistency of normative corpora to multi-agent systems, as well as modelling of decision-making. Notable examples include temporal logics, widely used in model checking and system verification [7], and provability logic ($\mathbb{GL}$), which formalises the notion of formal provability within mathematical theories [5].

Proof Assistants and HOL Light. An interactive theorem prover, or a proof assistant, is a software system designed to assist users in developing formal mathematical proofs.

HOL Light [19,20,21] is a proof assistant based on **h**igher-**o**rder **l**ogic, distinguished by its minimalist and elegant deductive system. It employs a simply typed lambda calculus and features a small logical core consisting of three classical axioms, ten inference rules, and a powerful principle for definitional extensions. This foundation ensures relative-reliability,[1] support for a broad range of formalisable mathematical theories, and extensibility, allowing users to construct proofs using built-in functions and to program and exploit new ones.

Mechanisation of Modal Logics. To mechanise a modal system means to develop formal and computational tools to represent, analyse, and manipulate it. This problem has given rise to a rich body of

[1] That is, reliability is guaranteed relative to the soundness of the logical core.

literature, including theoretical contributions [25,29,1] and implementations within proof assistants. Notable examples include Prolog formalisation of $\mathbb{S}5$ and $\mathbb{IS}5$ cubes [17,16], Lewis's conditionals [15,14], and non-normal logics [9,8]. Coq, too, supports intuitionistic epistemic logic [10] and classical modal systems in constructive type theory [18]. HOL-based proof assistants have been used for formalisation in higher-order logic of modal and temporal systems [20], and completeness proofs for some epistemic logics [11,12]. A decision procedure for $\mathbb{GL}$, developed by two co-authors of this paper [23,24], has been integrated into the official HOL Light distribution, covering axiomatisation, relational semantics, and an adequacy theorem.

Despite a high degree of modularity, existing approaches lack tools for assessing portability across modal logics and proof assistants, as well as metrics for evaluating compositional design. Extensible mechanisations have proven valuable in several contexts, such as formal analysis of complex normative texts, e.g. European AI Act [22], hybrid AI [28], and formalisation of classical philosophical argument, e.g. Gödel's ontological proof [2]. By leveraging our additional formalisation of adequacy results for the logics B, S4 and S5, we expect that HOLMS will also contribute to these applicative aspects of research in mechanised modal reasoning in the future.

2 HOLMS Framework

HOLMS, which stands for **HOL** Light Library for **M**odal **S**ystems, is the ongoing modular framework we are introducing to enhance the capabilities of the HOL Light proof assistant in (automated) modal reasoning. More concretely, we have extended the HOL Light deductive system with a new inference rule (`HOLMS_RULE`) which automatically decides whether a given modal formula is a theorem of a particular modal logic or, alternatively, constructs a countermodel. In its current state, HOLMS supports decision procedures for $\mathbb{K}$, $\mathbb{T}$, $\mathbb{K}4$ [2] and $\mathbb{GL}$, by implementing root-first proof search in the associated labelled calculi. The modular architecture of HOLMS — described in the following sections — allows the user to extend the library to cover the entire modal cube and, potentially, all normal modal logics. With additional effort, the framework may also be adapted to support non-normal logics.

In this talk, we present the library from a general perspective, highlighting key design choices and structural features of HOLMS. A more detailed and technically focused account is currently being refined. An earlier, embryonic version of the framework was also presented at the international workshop *OVERLAY 2024* [4], shortly before the defence of the first author's MA thesis [3]. An additional extension covering modal adequacy of further normal systems within the so-called S_5-cube is available from our repository ↗ and discussed in a different communication paper, currently under review.

2.1 Implementation Choices

Metalanguage and Object Languages. In our formalisation of modal systems, HOL Light serves as the *metatheory*, while modal logics are treated as *object logics*. This distinction requires the embedding of an object language within HOL Light—one capable of explicitly differentiating between formal statements of the modal language (e.g. $\Box A \to \Box\Box A$) and statements about modal systems in the metatheory (e.g. `|- !`$A.\vdash_{\mathbb{K}4} \Box A \to \Box\Box A$).

HOL Light is implemented in OCaml, a functional programming language that also serves as its *metalanguage*. Through this meta-level, we can define and manipulate the deductive apparatus of HOL Light. For instance, new inference rules can be encoded and executed via OCaml code such as `HOLMS_RULE `!`$A. \vdash_{\mathbb{K}4} \Box A \to \Box\Box A$`'`.

Embeddings. When representing formal systems within HOL-based proof assistants, two primary techniques are commonly used: *deep* and *shallow embedding* [6]. These two approaches differ in how they encode syntax and semantics of the object theory within the host HOL-based metatheory; the former requires the user to define a new type and an interpretation function to represent the embedded theory, while the second inherits them from the metatheory.

HOLMS adopts a *deep embedding* of a standard syntax for modal logic and Kripke semantics, from which

[2] The general mechanism provides a semi-decision procedure for $\mathbb{K}4$. However, the literature on labelled sequent calculi offers uniform solutions to this issue [13], which may be incorporated in future versions of HOLMS.

it defines a general provability predicate and modularly proves adequacy results. To develop decision procedures, it additionally implements a *shallow embedding* of Sara Negri's labelled sequent calculi [26,27] via HOL Light's goal stack mechanism and embedded logics. Henceforth, in HOLMS we have three interconnected presentations of (normal) modal logics: (i) axiomatic calculi; (ii) relational semantics–both deeply embedded; and (iii) labelled sequent calculi–which we shallow embed in the goal-stack mechanism of HOL Light as a certificate of correctness of the decision procedure behind `HOLMS_RULE`. The formalised adequacy theorem lets us safely reduce the provability task for a given formula into the task of proving its validity in the corresponding class of frames; the latter goal is solved (or refuted) by performing a root-first proof search in the corresponding labelled sequent calculus shallowly embedded in HOL Light.

Modularity. HOLMS generalises the previously developed $\mathbb{GL}$ library [23,24], with the ultimate goal of equipping HOL Light with a uniform mechanism for automated theorem proving and countermodel generation for modal logics. To do so, we developed the previously mentioned compositional implementation methodology— which Figure 1 summarises—centred on a scalable and uniform proof of modal adequacy, inspired by George Boolos' proofs in [5, §5].

To precisely measure and enhance code modularity, we adopted a precise coding discipline, inspired by [30], and distinguished between: *parametric polymorphic* code, fully independent of specific parameter instantiations; and *ad-hoc polymorphic* code, whose components are tailored to the modal logic under consideration. Figure 2 reports the measured modularity of the different components of the HOLMS implementation. Although HOL Light lacks explicit mechanisms to support parametric and/or ad hoc polymorphism (unlike, e.g., Isabelle/HOL), this distinction remains helpful in presenting the abstract structure of our formalisation and discussing the potential portability of our results to proof assistants that implement this distinction through specific mechanisms. We remark *en passant* that the approach we have used in HOL Light can be applied in other theorem provers. This possibility is mainly due to the simplicity of the logical framework we relied on: HOL Light features a comparatively "weaker" logic, extended only through basic extra/meta-theoretical mechanisms. The simpler the language, the broader the scope of possible generalisation(s).

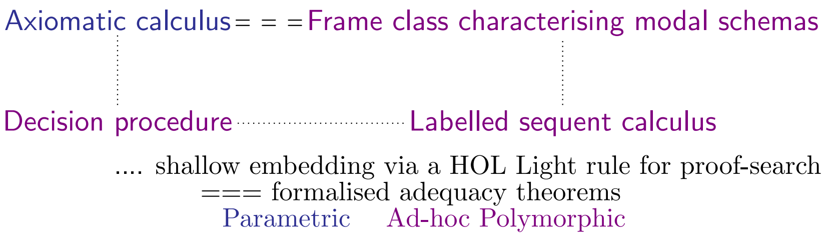
Axiomatic calculus
Frame class characterising modal schemas
Decision procedure
Labelled sequent calculus
.... shallow embedding via a HOL Light rule for proof-search
=== formalised adequacy theorems
Parametric
Ad-hoc Polymorphic

Fig. 1. The implementation methodology behind HOLMS.

Syntax		Parametric Polimorphic
Semantics		Parametric Polimorphic
Correspondence Theory	Concepts	Parametric Polimorphic
	Lemmata	Ad-hoc Polimorphic
Soundness		Parametric Polimorphic
Completeness	Standard Model	Parametric Polimorphic
	Truth Lemma	Parametric Polimorphic
	Countermodel Lemma	Parametric Polimorphic
	Standard Relation	Ad-hoc Polimorphic
	Identification of the Standard Model	Ad-hoc Polimorphic
	Type Generalisation	Ad-hoc Polimorphic
Shallow Embedding		Ad-hoc Polimorphic
Decision Procedure		Ad-hoc Polimorphic

Fig. 2. Measure of the modularity of the implementation.

References

[1] Benzmüller, C.: Faithful logic embeddings in HOL – A recipe to have it all: deep and shallow, automated and interactive, heavy and light, proofs and counterexamples, meta and object level (2025)

[2] Benzmüller, C., Scott, D.S.: Notes on Gödel's and Scott's variants of the ontological argument (Isabelle/HOL dataset). Archive of Formal Proofs (2025)

[3] Bilotta, A.: Growing a Modular Framework for Modal Systems- HOLMS: a HOL Light Library. Master's thesis, University of Florence (2025), https://arxiv.org/abs/2506.10048

[4] Bilotta, A., Maggesi, M., Perini Brogi, C., Quartini, L.: Growing HOLMS, a HOL Light Library for Modal Systems. In: Porello, D., Vinci, C., Zavatteri, M. (eds.) Short Paper Proceedings of the 6th International Workshop on Artificial Intelligence and Formal Verification, Logic, Automata, and Synthesis, OVERLAY 2024, Bolzano, Italy, November 28-29, 2024. CEUR Workshop Proceedings, vol. 3904, pp. 41–48. CEUR-WS.org (2024), https://ceur-ws.org/Vol-3904/paper5.pdf

[5] Boolos, G.: The logic of provability. Cambridge University Press (1995)

[6] Boulton, R., Gordon, A., Gordon, M., Harrison, J., Herbert, J., Tassel, J.V.: Experience with embedding hardware description languages in HOL. pp. 129–156

[7] Clarke, E., Grumberg, O., Peled, D., Peled, D.: Model Checking. The Cyber-Physical Systems Series, MIT Press (1999)

[8] Dalmonte, T., Negri, S., Olivetti, N., Pozzato, G.L.: Theorem Proving for Non-normal Modal Logics. In: OVERLAY 2020. Udine, Italy (Sep 2021), https://hal.science/hal-03159954

[9] Dalmonte, T., Negri, S., Olivetti, N., Pozzato, G., Terrioux, C.: PRONOM: A theorem prover for nonnormal modal logics. Available at http://193.51.60.97:8000/pronom/

[10] Doczkal, C., Smolka, G.: Constructive formalization of hybrid logic with eventualities. In: Jouannaud, J.P., Shao, Z. (eds.) Certified Programs and Proofs. pp. 5–20. Springer Berlin Heidelberg, Berlin, Heidelberg (2011)

[11] From, A.H.: Formalized soundness and completeness of epistemic logic. In: Silva, A., Wassermann, R., de Queiroz, R.J.G.B. (eds.) Logic, Language, Information, and Computation - 27th International Workshop, WoLLIC 2021, Virtual Event, October 5-8, 2021, Proceedings. Lecture Notes in Computer Science, vol. 13038, pp. 1–15. Springer (2021). https://doi.org/10.1007/978-3-030-88853-4_1, https://doi.org/10.1007/978-3-030-88853-4_1

[12] From, A.H.: An Isabelle/HOL Framework for Synthetic Completeness Proofs. In: Proceedings of the 14th ACM SIGPLAN International Conference on Certified Programs and Proofs. p. 171–186. CPP '25, Association for Computing Machinery, New York, NY, USA (2025). https://doi.org/10.1145/3703595.3705882, https://doi.org/10.1145/3703595.3705882

[13] Garg, D., Genovese, V., Negri, S.: Countermodels from sequent calculi in multi-modal logics. In: Proceedings of the 27th Annual IEEE Symposium on Logic in Computer Science, LICS 2012, Dubrovnik, Croatia, June 25-28, 2012. pp. 315–324. IEEE Computer Society (2012). https://doi.org/10.1109/LICS.2012.42, https://doi.org/10.1109/LICS.2012.42

[14] Girlando, M., Lellmann, B., Olivetti, N., Pesce, S., Pozzato, G.L.: Theorem proving for Lewis Logics of Counterfactual Reasoning. In: CILC 2020 - 35th Edition of the Italian Conference on Computational Logic. Rende / Virtual, Italy (Oct 2020), https://hal.science/hal-03080670

[15] Girlando, M., Lellmann, B., Olivetti, N., Pozzato, G.L., Vitalis, Q.: Vinte: An implementation of internal calculi for lewis' logics of counterfactual reasoning. In: Schmidt, R.A., Nalon, C. (eds.) Automated Reasoning with Analytic Tableaux and Related Methods. pp. 149–159. Springer International Publishing, Cham (2017)

[16] Girlando, M., Morales, M.: MOILab: towards a labelled theorem prover for intuitionistic modal logics (Dec 2020), https://hal.science/hal-03048966, working paper or preprint

[17] Girlando, M., Straßburger, L.: Moin: A nested sequent theorem prover for intuitionistic modal logics (system description). In: Peltier, N., Sofronie-Stokkermans, V. (eds.) Automated Reasoning. pp. 398–407. Springer International Publishing, Cham (2020)

[18] Hagemeier, C.: Formalizing intuitionistic epistemic logic in Coq. Ph.D. thesis, BSc thesis (2021)

[19] Harrison, J.: The HOL Light manual (1.1). University of Cambridge. https://www.cl.cam.ac.uk/~jrh13/hol-light/manual-1.1.pdf (2000)

[20] Harrison, J.: HOL Light tutorial. Intel Corporation. http://www.cl.cam.ac.uk/~jrh13/hol-light/tutorial.pdf (2017)

[21] Harrison, J.: The HOL Light Theorem Prover. https://hol-light.github.io/ (2025)

[22] Lawniczak, L., Benzmüller, C.: Logical modalities within the european ai act: An analysis (2025)

[23] Maggesi, M., Perini Brogi, C.: A formal proof of modal completeness for provability logic. In: Cohen, L., Kaliszyk, C. (eds.) 12th International Conference on Interactive Theorem Proving (ITP 2021). Leibniz International Proceedings in Informatics (LIPIcs), vol. 193, pp. 26:1–26:18. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl, Germany (2021). https://doi.org/10.4230/LIPIcs.ITP.2021.26, https://drops.dagstuhl.de/opus/volltexte/2021/13921

[24] Maggesi, M., Perini Brogi, C.: Mechanising Gödel-Löb Provability Logic in HOL Light. Journal of Automated Reasoning **67**(3), 29 (2023). https://doi.org/10.1007/S10817-023-09677-Z, https://doi.org/10.1007/s10817-023-09677-z

[25] Nalon, C., Hustadt, U., Papacchini, F., Dixon, C.: Buy one get 14 free: Evaluating local reductions for modal logic. In: Pientka, B., Tinelli, C. (eds.) Automated Deduction – CADE 29. pp. 382–400. Springer Nature Switzerland, Cham (2023)

[26] Negri, S.: Proof analysis in modal logic. Journal of Philosophical Logic **34**(5), 507–544 (2005)

[27] Negri, S., von Plato, J.: Structural proof theory. Cambridge university press (2008)

[28] Steen, A., Benzmüller, C.: Challenges for non-classical reasoning in contemporary AI applications. Künstliche Intelligenz **38**(1-2), 7–16 (2024)

[29] Steen, A., Sutcliffe, G., Benzmüller, C.: Solving quantified modal logic problems by translation to classical logics. Journal of Logic and Computation p. exaf006 (02 2025). https://doi.org/10.1093/logcom/exaf006, https://doi.org/10.1093/logcom/exaf006

[30] Strachey, C.: Fundamental concepts in programming languages. Higher-order and symbolic computation **13**, 11–49 (2000)

Base-extension Semantics for Classical Logic

Ekaterina Piotrovskaya

University College London, London, U.K.

Model-theoretic semantics defines logical consequence through models – abstract mathematical structures that interpret propositions and determine their truth. Following Tarski [14], a sentence ϕ follows from a set of premises Γ (written $\Gamma \Vdash \phi$) if and only if every model that satisfies all sentences in Γ also satisfies ϕ. In this view, consequence, meaning and validity are all characterised in terms of *truth*.

Proof-theoretic semantics (P-tS) [13,15] provides an alternative approach to the meaning of logical operators. It is based on inferentialism – a view according to which the meaning of expressions arises from their use in the system of inference and is defined by the rules of said system. The meaning is thus expressed in terms of proofs and provability: the concept of truth in model-theoretic semantics is now substituted with that of *proofs*.

One major branch of P-tS is *base-extension semantics* (B-eS) [9,10,11,12]. The central notion of B-eS is a fixed atomic system – a set of inference rules over atomic propositions – called a *base* ($\mathcal{B}$); said rules can be seen as fixing the inferential roles or meanings of those atoms. The characterisation of consequence in B-eS is given via a judgement called *support* ($\Vdash_{\mathcal{B}}$), which is inductively defined on the structure of formulas, with the base case (i.e. support of atoms) determined by *provability in a base*. The support relation mimics the compositional nature of meaning, as in model-theoretic semantics, but it is defined in terms of provability rather than truth in a model. In B-eS, the meaning of complex formulas is determined by their structure and the support of their subformulas, beginning with the proofs of atoms in a base. The validity is defined through the quantification over all bases:

$$\Gamma \Vdash \phi \;\text{ iff }\; \Gamma \Vdash_{\mathcal{B}} \phi \;\text{ for all } \mathcal{B}$$

B-eS has been given to a number of intuitionistic and substructural logics (see, e.g. [12] for intuitionistic propositional logic (IPL), [2,6] for intuitionistic linear logic (ILL) and its fragments, [7] for logic of bunched implications). Sandqvist [11] has given a B-eS for classical propositional logic (CPL); his semantics gives clauses to implication ($\to$) and $\bot$, while not admitting disjunction ($\vee$) as a primitive. This has been viewed as a limitation by Makinson [8] and inspired Gheorghiu and Buzoku [4] to give an alternative B-eS for CPL. They view CPL as IPL together with a duality and, while keeping the clauses of [12], work with literals – atoms together with their duals – as opposed to just atoms. According to Gheorghiu et al. [5,6], B-eS can be smoothly given to logics whose natural deduction systems exhibit a certain harmony; while such property holds for intuitionistic logics, it is lost in the classical case with rules such as *reductio as absurdum*. As a result, the methods used to develop B-eS for intuitionistic logic cannot be directly applied to classical logic, which gives rise to different approaches to B-eS for CPL.

In the case of the multiplicative-additive fragment of (classical) linear logic (MALL) [1], B-eS has been given by uniformly restricting the clauses for the connectives in ILL [2]: instead of demanding support of an arbitrary atom, one demands the support of $\bot$ (which is considered atomic in said approach). Such shape of the clauses allows for the semantic *reductio ad absurdum*, hence allowing for the semantics to mimic classical consequence. This approach is the starting point of this talk.

Shifting the perspective back to classical propositional logic, we will try to both simplify said approach and make it closer in spirit to the established treatment of B-eS. First, recalling the inductive definition of the support relation, we apply the restriction solely to the base case. Secondly, we return to treating $\perp$ as a connective, as done in [12], following Dummett's [3] view of falsity as logical explosion. The base case thus becomes defined not in terms of support of atomic bottom, but in terms of provability of all atoms. Finally, the restricted definition of the base case (both in [1] and in the currently discussed approach), makes the extension of derivability in a base to the full language no longer conservative (as opposed to the usual treatment of B-eS); more atoms are supported than provable in a base. We will discuss whether it is possible to tackle this by keeping the base clause as in [12] and defining a separate relation for formulas defined in terms of provability of all atoms. Finally, we will reflect on the similarities and differences of our approach with those in [11] and [4] and try to make a connection with both of them.

Acknowledgements. The author is supported by the Engineering and Physical Sciences Research Council grants EP/T517793/1 and EP/W524335/1.

References

[1] Barroso-Nascimento, V., Piotrovskaya, E., Pimentel, E.: A proof-theoretic approach to the semantics of classical linear logic. arXiv preprint arXiv:2504.08349 (2025)

[2] Buzoku, Y.: A proof-theoretic semantics for intuitionistic linear logic. arXiv preprint arXiv:2402.01982 (2024)

[3] Dummett, M.: The logical basis of metaphysics. Harvard university press (1991)

[4] Gheorghiu, A.V., Buzoku, Y.: Proof-theoretic semantics for classical propositional logic with assertion and denial. arXiv preprint arXiv:2503.05364 (2025)

[5] Gheorghiu, A.V., Gu, T., Pym, D.J.: A note on the practice of logical inferentialism. arXiv preprint arXiv:2403.10546 (2024)

[6] Gheorghiu, A.V., Gu, T., Pym, D.J.: Proof-theoretic semantics for intuitionistic multiplicative linear logic. Studia Logica pp. 1–61 (2024)

[7] Gu, T., Gheorghiu, A.V., Pym, D.J.: Proof-theoretic semantics for the logic of bunched implications. arXiv preprint arXiv:2311.16719 (2023)

[8] Makinson, D.: On an inferential semantics for classical logic. Logic Journal of IGPL **22**(1), 147–154 (2014)

[9] Piecha, T.: Completeness in proof-theoretic semantics. Advances in proof-theoretic semantics pp. 231–251 (2016)

[10] Piecha, T., de Campos Sanz, W., Schroeder-Heister, P.: Failure of completeness in proof-theoretic semantics. Journal of Philosophical Logic **44**, 321–335 (2015)

[11] Sandqvist, T.: Classical logic without bivalence. Analysis **69**(2), 211–218 (2009)

[12] Sandqvist, T.: Base-extension semantics for intuitionistic sentential logic. Logic Journal of the IGPL **23**(5), 719–731 (2015)

[13] Schroeder-Heister, P.: Proof-Theoretic Semantics. In: Zalta, E.N., Nodelman, U. (eds.) The Stanford Encyclopedia of Philosophy. Metaphysics Research Lab, Stanford University, Winter 2022 edn. (2022), https://plato.stanford.edu/archives/win2022/entries/proof-theoretic-semantics/

[14] Tarski, A.: On the concept of following logically. History and Philosophy of Logic **23**(3), 155–196 (2002)

[15] Wansing, H.: The idea of a proof-theoretic semantics and the meaning of the logical operations. Studia Logica **64**, 3–20 (2000)

Teaching logic as teaching to read and write

Milena Dassie Wilke[a]

[a] *Universidad Nacional de Córdoba*

Abstract

Starting from Goodman's concept of a notation system, this essay presents the analogy between the process of learning logic and the process of alphabetization in children as a metaphor to guide our logic teaching. This perspective draws our attention towards the formal languages and deductive systems in use in an introductory logical course. In particular, it focuses on the process of learning to construct a proof in a deductive system. It proposes a notational analysis of these systems in light of teaching logic in an introductory course.

1 Introduction

Taking Nelson Goodman's conception of a notation system and applying it to deductive systems, we obtain an innovative way of understanding these systems that has interesting repercussions for teaching logic. Under the notational perspective, focused on visual and mereological considerations, proofs do not appear only as a purely syntactical product, but rather as the result of a process of using strict combination rules. The notational analyses of deductive systems promote an analogy between the process of teaching and learning to construct proofs and the process of alphabetization. This analogy brings us to a socially oriented conceptualization of logic education, that addresses human relationships and takes into account the history and culture of logic. Such an approach focuses on dialogue and cooperation, respect for differences, and a careful accompaniment of students in addition to fostering their autonomy. This paper puts forward a socio-constructivist perspective on logical knowledge, logic students, and logic teachers, along with a visual and mereological conception on deductive systems.

Under Vygotsky's socio-constructivist theory [1], logical knowledge is understood as the result of a social process of construction by logicians throughout the years. Symbol systems like formal languages and deductive systems are understood as tools constructed to mediate between logicians and the world. These mediation tools, despite being constructed to transform the world, have the main effect of transforming their users and creating a particular way of seeing, thinking, and acting on the world [2]. Following Vygotsky's perspective, we conceptualize formal languages as tools used to interact with and make sense of the world, but also as tools that shape us. In this sense learning to use a deductive system and a formal language is a matter of internalizing a social way of seeing, thinking, and acting on the world.

The main goal of this paper is to propose a conceptualization of exactly what this logical way of seeing, thinking, and acting entails. To do this, we turn to the education literature focused on the process of learning to use a symbol system like written languages. We propose the metaphor of learning to construct proofs in a deductive system as learning to read and write logically.

To do this, we take into account a bibliography on notational theory to understand the internal function of deductive systems and the rules of combination of characters to get a proof. These considerations give us a series of rules to follow that students must master to construct a proof, but not on how to master them. The bibliography on alphabetization and the general socio-constructivist theory of learning allows us to propose a conceptualization of the process of learning to construct proofs.

2 Symbol systems

By "reading logically," we mean understanding a proof in a certain way, abstracting, dividing, constructing individuals, and grouping them in a certain way. By "writing logically," we mean putting symbols on a sheet and arranging them in a particular way. These categories come from an analogy between the process of learning logic and the process of alphabetization proposed in this paper.

The use of alphabetization (or literacy) bibliography provides us with a framework, structure, or guide to propose a conceptualization of what it entails to learn logic [3], [4], [5], [6], [7]. For example, this bibliography proposes that one of the first processes in alphabetization is that of differentiating between one's drawings and a written object. In analogy, in logic, a student must learn to differentiate an image from a proof. Therefore, the process of adopting a way of reading and writing a proof is understood as the process of seeing a proof from an image. This means constructing a proof as a logical tool.

In the theory of notation, another literature that focuses on written language or symbol systems, we find the discussion of what differentiates an image from a writing. Nelson Goodman [8] answered that, where images' marks are dense, the characters in symbol systems are discontinuous because there are spaces that allow us to distinguish between one mark and the other.

Goodman proposed a technical terminology to analyse symbol schemes [8]. In a sheet, we find visual 'marks' like "p", "1" or "⊃", a mark becomes an 'inscription' when it belongs to a 'character'. In a symbol scheme, every mark is syntactically indistinguishable from the marks that belong to the same character. We can then distinguish between types of characters and bring together the letters of the alphabet as non-logical symbols "p, q, r, s", the connectives as logical symbols "⊃", "&", "≡", "∨", "∼" and the auxiliaries "(,)". These types of characters are then combined by rules of combination, in logic, the rules of well-formed formulas. A student must learn the rules that govern the construction of well-formed formulas, as well as they must learn the rules that allow them to go from one combination of characters to another: the rules of inference.

Nonetheless, this isn't enough to get a proof, as this construction is done inside a particular structure: a deductive system [9]. A deductive system forces us to take into account other characters and rules in addition to those of a formal language. For example, the deductive system proposed by Jaskowski [10], Gentzen [11], and Fitch [12] makes use of vertical and horizontal lines as marks, characters of the numeric system "1, 2, 3, 4..." and the use of marks like "-" and ",". We can call these characters 'extras' or 'particulars' of a deductive system.

Logical symbols, non-logical symbols, auxiliaries and extras are arranged according to the particular ordination rules of a deductive system. Krämer [13] expands Goodman's notation theory, focusing on the use of ordering in symbol systems. For the author, written language exploits the two-dimensionality of the surface it is written on, this means that it is informative if a formula is written at the top or bottom of the sheet and if it's written to the right or left of a character. The ordination rules of a deductive system control the arrangement of symbols in a sheet, for example, a proof in Gentzen's system is arranged horizontally while a proof in Jaskowski's system or Fitch-style system is arranged vertically.

Given this formal description of a proof in a deductive system, we must say that for a student to learn logic, they must learn to use a formal language in a particular way: use limited characters, combine and transform them in a regular way and organize them in a particular structure. Now, there is also another dimension to formal languages that comes from taking into account the use of these symbol systems. Schlimm expands Goodman's theory of notation and proposes a series of criteria for analysing a symbol system; we will only focus on one: verbalization [14]. Written formal languages are verbalized in oral natural languages in the process of teaching logic when we name and describe proofs. In this process, we construct parts of proofs: we talk about 'steps', 'formulas', 'premises', 'conclusions', 'justifications', 'suppositions', 'subproofs', etc. (for an example see [12]). Therefore, learning to read and write logically is not only a matter of putting symbols in the right place but also of grouping characters as an articulation of parts.

3 Learning and teaching logic

The literature on notational theory allows us to understand the ideal of what a student must learn and how they must read and write a proof to understand it logically. This theory specifies the composition rules of formal languages and deductive systems. This gives us a clear view of what we want students to learn, but not of how they can learn it and not of how we can teach them. We expand this analysis to the socio-constructivist theory to fill in this gap.

As well as knowledge, in the socio-constructivist perspective, the process of learning takes place in a social environment. Vygotsky's theory conceptualizes this process as an internalization of knowledge in an interpersonal dialogue [2]. Bruner understands this dialogue as narratives shared by the teacher and the students [15], [16]. Then, knowledge needs to be transformed to take place in a conversation. Following this perspective, in logic, we find the interaction between the rigid and established rules of deductive systems and the ambiguity of natural language, narratives and social context. Nonetheless, these social aspects of the process of learning and teaching logic do not distort logic's formality but rather make it accessible.

The internalization of a logical way of seeing, thinking, and acting is not a passive process for the students, but rather an active process of construction of meaning and hypothesis [6], [7]. Students do not construct a formal language (as it was established, this tool was constructed by many people throughout many years), but they do construct an interpretation or narrative above this system [16]. Such interpretations can encounter difficulties when interacting with the rules of logical tools; for example, a logic student who holds the narrative that every argument can be proven will encounter problems with their narrative when dealing with invalid arguments. In this sense, seeing, thinking, and acting logically is a matter of adopting the interpretation that logicians construct of these systems. Finally, this adoption is not, once again, passive, but critical. Critical thinking requires giving students a certain amount of freedom in constructing their narratives, which could result in questioning established logic. Under the socio-constructivist theory of learning, teaching does not aim to form passive and repeating agents, but rather to form critical and reflective investigators [17].

The process of teaching logic is understood as the effort to bring into dialogue the narratives established by logicians and the various narratives that are in the process of being constructed by logic students. The logical way of seeing, thinking, and acting, already internalized by the teacher, needs to be in conversation with the way of seeing, thinking, and acting that students are constructing by interacting with knowledge and logical tools. A logic teacher has the role of sharing the logical way of reading and writing proofs, as well as recognising the interpretations of their students, and pointing out, when needed, the difficulties in these interpretations. It is in this sense that logical knowledge is shared by dialogue between teachers, students, their respective interpretations, and the formality and necessity of proofs themselves.

In conclusion, this approach to logic teaching is centred on the social interactions in a logical classroom. The recognition of student's active role in constructing their own understanding of the rules of a deductive system through the dialogue with their teachers, their peers, and indirectly with the rest of society allows us to take into account both the regularities of logic itself and the autonomy and critical thinking of students. Vygotsky's and Bruner's perspectives on the centrality of dialogue in the learning process allow us to take into account a cooperative perspective on classroom interactions and logic [17], as it is discussed in the feminist perspectives on argumentation [18]. This socially oriented perspective on teaching and learning logic prioritizes caring tasks that have historically been neglected. In this sense, the logic classroom is understood as a place where strict and necessary rules meet with a cooperative dialogue between humans.

References

[1] Vygotsky, L. S.: Thought and language (A. Kozulin, Trans.). Mit Press. (1968).

[2] Vygotsky, L. S.: Mind in Society: Development of Higher Psychological Processes (M. Cole, V. John-Steiner, S. Scribner, & E. Souberman, Eds.). Harvard University Press. (1978).

[3] Castedo, M., Dávalos, A., Möller, M. A., & Soto, A. (Eds.): Enseñar a leer y escribir en contextos diversos: Aportes para la formación docente [Teaching to read and write in diverse contexts: Contributions to teacher training]. Dirección General de Educación Superior.(2022).

[4] Catach, N.: La ponctuation: Histoire et système [Punctuation: History and System]. Presses Universitaires de France. (1994).

[5] Ferreiro, E.: Los niños piensan sobre la escritura [Children think about writing]. CD Multimedia. Siglo XXI Editores. (2003).

[6] Ferreiro, E.: El ingreso a la escritura y a las culturas de lo escrito [Entry into writing and written cultures]. Siglo XXI. (2013).

[7] Ferreiro, E., & Teberosky, A.: Los sistemas de escritura en el desarrollo del niño [Writing systems in child development]. Siglo XXI. (1979).

[8] Goodman, N.: Languages of Art: An approach to a theory of symbols. The Bobbs-Merrill Company. (1968).

[9] Pelletier, F. J.: A Brief History of Natural Deduction. History and Philosophy of Logic. **20**(1), 1–31 (1999), https://doi.org/10.1080/014453499298165

[10] Jaskowski, S.: On the Rules of Suppositions in Formal Logic. In S. McCall, Polish Logic 1920-1939 (Vol. **1**, 232–258). (1967). (Original work published 1934).

[11] Gentzen, G.: Untersuchungen über das logische Schliessen [Investigations into logical reasoning]. Mathematische Zeitschrift. **39**, 176–210, 405–431 (1934/5).

[12] Fitch, F. B.: Symbolic Logic An Introduction. The Ronald Press Company. (1952).

[13] Krämer, S.: Writing, Notational Iconicity, Calculus: On Writing as a Cultural Technique (A. McChesney, Trans.). **118**(3), 518–537 (2003), https://doi.org/10.1353/mln.2003.0059

[14] Schlimm, D.: Mathematical notations. Cambridge University Press. (2025).

[15] Bruner, J.: The Role of Dialogue in Language Acquisition. In A. Sinclair, R. J. Jarvella, & W. J. M. Levelt (Eds.). The Child's Conception of Language, (pp. 241–256). Springer-Verlag. (1978).

[16] Bruner, J.: Actual minds, possible worlds. Harvard University Press. (1986).

[17] Radford, L.: La teoría de la objetivación: Una perspectiva vygotskiana sobre saber y devenir en la enseñanza y el aprendizaje de las matemáticas [Objectification theory: A Vygotskian perspective on knowing and becoming in the teaching and learning of mathematics]. Ediciones Uniandes. (2023).

[18] Hundleby, C.: Feminist Perspectives on Argumentation. In E. N. Zalta & U. Nodelman (Eds.), The Stanford Encyclopedia of Philosophy (Fall 2023 Edition). (2023) https://plato.stanford.edu/archives/fall2023/entries/feminism-argumentation/

Formally Verified Verifiable Group Generators

Mina A. Cyrus[1]

Swansea University, Swansea, UK
m.a.swansea@swansea.ac.uk

Abstract

Electronic voting (e-voting) requires a trusted setup to initiate an election process. This setup must be transparent to maintain the integrity of the election. A crucial aspect of this trusted setup involves generating group generators for a finite cyclic group, which are then used in cryptographic algorithms deployed within the voting system. Although computing group generators is generally not considered a difficult problem, election verifiability – where every step can be ascertained by independent third parties – excludes many of them, as they fail to provide verifiable evidence of correctness. In this work, we present a formally verified implementation of the group generator algorithm `A.2.3` and the group generator verification algorithm `A.2.4`, specified in the National Institute of Standards and Technology (NIST), FIPS 186-4, in the Rocq theorem prover. These two algorithms are highly sought-after methods to compute and verify group generator(s), respectively, because their outcomes can be established independently by third parties. Our formalisation captures all the requirements specified in algorithms A.2.3 and A.2.4 using the expressive type system of the Rocq theorem prover.

1 Introduction

In recent years, e-voting – using computer programs in some aspect of an election – is becoming more popular because it addresses challenges such as accessibility, efficiency, and scalability in modern elections. For example, France and Switzerland use internet voting (i-voting) to include their overseas voters, ensuring inclusivity and convenience in the decision-making process. India and Brazil use electronic voting machines (EVMs) to record and tally ballots, making the process faster and more reliable in large-scale elections. Australia uses a computer program – that implements the single transferable vote (STV) method – for declaring the results of its senate elections, ensuring accurate and efficient processing of preferential ballots. However, there is a history of bugs, varying from trivial to critical, in e-voting software programs employed in democratic elections, e.g., Scytl/SwissPost [HLPT20] (used in Switzerland), Voatz [SKW20] (used in West Virginia, USA), Democracy Live Online Voting System [SH21] (used in Delaware, West Virginia, and New Jersey, USA), Moscow Internet Voting System [GG20] (used in Russia), and iVote System [HT15, CBNT17] (used in New South Wales, Australia). Most software programs establish their correctness through testing, but it does not cover all possible scenarios. Moreover, these software programs are proprietary, so members of the general public, including researchers, are not allowed to inspect them [Aus13]. We argue that all components of e-voting software programs should be developed with utmost rigour using existing formal verification techniques. Additionally, these components should be open source, allowing anyone to inspect the code and independently verify any claims made about it.

2 Our Contribution

In this work, we focus on the problem of computing independent generators, specifically in the context of e-voting [HLPT20]. Algorithm A.2.3 is a well-established method for computing independent generators. It takes public data as input and produce generators. This public data is made available on a public bulletin board so that any third party can verify the correctness of the process. The rationale is that when two generators g and h are produced by some public data, it becomes difficult, or computationally infeasible, to determine their discrete logarithm. In this formalisation, we translate the English prose describing algorithms A.2.3 and A.2.4 into formal implementations using the Rocq theorem prover [Tea22]. Consequently, all the correctness assertions originally expressed in English prose are transformed into theorems that must be formally proven within the Rocq system. By employing the most precise system available – formal systems based on constructive logic – we implement algorithms A.2.3 and A.2.4 into computer programs, thereby significantly reducing the gap between the theory and practice. Our formalisation, available at this repository[1], comprises 5000 lines of Rocq code and proofs for the following:

1. implementation of A.2.3 with a correctness proof that it always produces correct generators.
2. implementation of A.2.4 to validate generators computed according to A.2.3.
3. implementation of the *SHA-256 (FIPS 180-4)* [Nat15] hash algorithm, required in A.2.3 and A.2.4.
4. implementation of the *Fermat's little theorem*, needed to prove the correctness of A.2.3 and A.2.4.

2.1 Proof of Correctness

Theorem `generator_in_range_2_p` asserts that the value returned by Algorithm A.2.3 (`compute_generator`) lies strictly within the interval $[2, p-1]$. Theorem `correct_compute_generator` guarantees that the generator g returned by `compute_generator` satisfies the relation $g^q \equiv 1 \mod p$, confirming that g generates a subgroup of order q within Z_p^*; this result follows as a corollary of Fermat's Little Theorem. Theorem `generator_verifier_correctness` ensures consistency between Algorithm A.2.3 and Algorithm A.2.4 (`verify_generator`): a value g is returned by `compute_generator` if and only if `verify_generator` g returns `true`.

Listing 1: Correctness of generator algorithm and its connection with verification algorithm

```
Theorem generator_in_range_2_p : forall (g : N),
  g = compute_generator -> 2 <= g < p.

Theorem correct_compute_generator : forall g,
  g = compute_generator -> g^q mod p = 1.

Theorem generator_verifier_correctness : forall g,
  g = compute_generator <-> verify_generator g = true.

Theorem fermat_little_simp : forall a p : nat,
 prime p ->  a^p mod p = a mod p.
```

[1] https://github.com/Artun1s/Formally_Verified_Verifiable_Group_Generator

References

[Aus13] Australian Electoral Commission. Letter to Mr Michael Cordover, LSS4883 Outcome of Internal Review of the Decision to Refuse your FOI Request no. LS4849, 2013. avaliable via http://www.aec.gov.au/information-access/foi/2014/files/ls4912-1.pdf, retrieved February 8, 2022.

[CBNT17] Andrew Conway, Michelle Blom, Lee Naish, and Vanessa Teague. An Analysis of New South Wales Electronic Vote Counting. In *Proceedings of the Australasian Computer Science Week Multiconference*, ACSW '17, New York, NY, USA, 2017. Association for Computing Machinery.

[GG20] Pierrick Gaudry and Alexander Golovnev. Breaking the Encryption Scheme of the Moscow Internet Voting System. In Joseph Bonneau and Nadia Heninger, editors, *Financial Cryptography and Data Security*, pages 32–49, Cham, 2020. Springer International Publishing.

[HLPT20] Thomas Haines, Sarah Jamie Lewis, Olivier Pereira, and Vanessa Teague. How not to prove your election outcome. In *2020 IEEE Symposium on Security and Privacy (SP)*, pages 644–660, 2020.

[HT15] J. Alex Halderman and Vanessa Teague. The New South Wales iVote System: Security Failures and Verification Flaws in a Live Online Election. In Rolf Haenni, Reto E. Koenig, and Douglas Wikström, editors, *E-Voting and Identity*, pages 35–53, Cham, 2015. Springer International Publishing.

[Nat15] National Institute of Standards and Technology. Secure Hash Standard (SHS), 2015. avaliable via https://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.180-4.pdf, retrieved February 8, 2022.

[SH21] Michael Specter and J. Alex Halderman. Security Analysis of the Democracy Live Online Voting System. In *30th USENIX Security Symposium (USENIX Security 21)*, pages 3077–3092. USENIX Association, August 2021.

[SKW20] Michael A. Specter, James Koppel, and Daniel Weitzner. The Ballot is Busted Before the Blockchain: A Security Analysis of Voatz, the First Internet Voting Application Used in U.S. Federal Elections. In *29th USENIX Security Symposium (USENIX Security 20)*, pages 1535–1553. USENIX Association, August 2020.

[Tea22] The Coq Development Team. The coq proof assistant, version 8.15.0, October 2022.

A Default Logic Model of Successful Speech Acts

Yan Xu[1*a] Xue Ge[2 b] Sara L. Uckelman[1 c]

[a] *Durham University*

[b] *Sun Yat-sen University & Durham University*

[c] *Durham University*

Abstract

Language is not only a tool for communication but also a way of acting. In his theory of speech acts, Austin [1] argued that people can do things by speaking, which can bring about effects in our lives. However, most existing research approaches adopt the listener's perspective, contemplating how speakers should express themselves to ensure the effectiveness of their speech acts. As a result, there is a failure to adequately explain why listeners often feel being led in response to the speaker's intention, and why they might agree with the speaker outwardly when they actually disagree inwardly. To address this gap, this paper adopts the phenomenological framework of self-consciousness to explain how speakers capture and shape listeners' responses in order to promote successful speech acts. We further introduce a dynamic logic model to formalize this process and to examine the capturability of listener responses. The main contribution of this study lies in the construction of a logical system—based on default logic—that enables discourse agents to engage in inside-out analogical reasoning grounded in common sense. The ultimate goal is to simulate how external stimuli trigger responses within the agent's internal mental world.

1 Extended abstract

The complexity of human natural language transcends mere sound production, expression, and information transmission; it fundamentally encompasses intentional human behavior. Austin [1] describes this behaviour, which is implemented through speech via "speech acts," underscoring that speaking not only serves to communicate information but also functions as a tool for executing specific social actions. He argues that when people speak, they simultaneously engage in three different levels of speech acts: (1) The locutionary act, where the speaker expresses words, conveying the meaning defined in the language. (2) The illocutionary act, which means achieving a specific action through the speech, such as making a promise, taking an oath, or issuing a warning. (3) The perlocutionary act through the utterance, that is, the consequences or effects a speech act brings about. Since the meaning that people convey is not always literal, the speaker's illocutionary acts are not always understandable through the semantic content of the sentence content alone. The difficulty that arises is that different interpretations of the speaker's illocutionary acts may affect the successful performance of the speaker's speech acts and their intended outcomes. Therefore, it is important for speakers to ensure that their illocutionary acts are accurately understood by the listeners. This raises the question: what makes a speech act successful? We identify four essential steps involved in the process: (i) Triggering the listener's internal reaction through specific utterance stimuli. (ii) Transforming personal psychological intentionality into the listener's psychological experience. (iii) Fostering a sense of consensus with a goal. (iv) Revenging on her husband as a response to the words she heard.

In this way, the speaker infers inner mental states, such as beliefs and emotions, through external reactions, by understanding how such states give rise to observable behavior [2, p. 488].

Dialogue is an interactive process that requires participants to respond instantly, often relying on their own knowledge bases to interpret others' verbal and non-verbal behaviors. Speakers must quickly assess whether their communicative intentions have been correctly understood. When new information suggests a mismatch between intention and interpretation, it becomes necessary to adjust one's speech strategy in order to achieve goals such as persuasion. Such reasoning, by nature, is uncertain and context-sensitive. To formally model this mechanism, we introduce modal default logic, a form of nonmonotonic logic, to account for how speakers draw conclusions under incomplete knowledge conditions. Originally introduced by [3], default logic allows agents to make tentative inferences based on consistent assumptions, which are retained unless contradicted by new evidence. This framework captures the defeasible and revisable nature of human reasoning in dialogue. Default logic has been widely applied in fields such as knowledge representation, logic programming ([4]), and medical diagnosis ([5]). In this study, we adapt its modalized form to represent how speakers reason about belief, intention, and response during real-time interaction.

2 Syntactic system

2.1 Language

Definition 2.1 [Formulaes] The set of formulas (terms) $\mathcal{F}$ is defined inductively as follows:

$$\mathcal{F} \ni \phi ::= p \mid \top \mid \bot \mid \neg\phi \mid \phi \wedge \psi \mid \phi \vee \psi \mid \phi \rightarrow \psi \mid \phi \overset{\Box}{\rightarrow} \psi \mid K_a\phi \mid B_a\phi \mid !_{<a,b>}[\phi]\psi$$

where $p \in \mathbf{Var}$. I is a finite set of agents, a, $b \in I$ are two agents in a private conversation.

Propositional variables can be facts and prerequisite information grounded in common sense; $\overset{\Box}{\rightarrow}$ [1] denotes the default implication concluded from using the default rules, its binding strength is weaker than that of all other connectives and doesn't satisfy the law of exportation; $K_a(\phi)$ denotes that agent a knows ϕ; $B_a(\phi)$ denotes that agent a believes ϕ; ! is a dynamic operator; $!_{<a,b>}[\phi]\psi$ denotes the possibility that action ψ will be executed in the future or now after agent a tells agent b proposition ϕ, in which $< a, b >$ is an ordered pair. ! can be introduced into different types of speech act, such as commanding, requesting, asserting, etc.

K and B here are treated merely as cognitive conditions in the current state and have truth values only relative to this current state. K indicates the agent has the ability of introspection, $K\phi$ is true if and only if ϕ is a fact that can be observed. $B\phi$ means that the agent believes that ϕ is consistent with his own current knowledge base. And $B_a\phi$ is true doesn't mean that ϕ is true.

2.2 Axioms and inference rules

Let Δ =(**D,W**) is a closed default theory where **D** is a set of defaults and **W** is a set of closed wffs. The system includes all the standard axioms of propositional logic, modal logic S5 and is closed under uniform substitution and modus ponens. The system also contains the following axioms and rules:

Axioms:

(1) $B_a(\phi \rightarrow \psi) \wedge B_a\phi \rightarrow B_a\psi$

(2) $!_{<a,b>}[\phi](\psi \wedge \chi) \rightarrow (!_{<a,b>}[\phi](\psi) \wedge !_{<a,b>}[\phi](\chi))$

(3) $!_{<a,b>}[\phi](\psi \vee \chi) \rightarrow (!_{<a,b>}[\phi](\psi) \vee !_{<a,b>}[\phi](\chi))$

(4) $!_{<a,b>}[\phi](\psi \rightarrow \chi) \rightarrow (!_{<a,b>}[\phi](\psi) \rightarrow !_{<a,b>}[\phi](\chi))$

Rules:

$$\frac{\phi}{K_a\phi}(\text{Nec}) \qquad \frac{\phi \rightarrow \psi}{K_a\phi \rightarrow K_a\psi}(\text{Mon})$$

$$\frac{\phi\ (prerequisite) : \psi_1, \cdots, \psi_n\ (justifications)}{\chi\ (conclusion)}(n \geq 1)(\text{General default rule})$$

[1] It was first introduced by [6].

References

[1] Austin, J.L.: How to do things with words. Harvard university press. (1962).

[2] Dodds, A.E., Lawrence, J.A., Valsiner, J. : The personal and the social: Mead's theory of thegeneralized other'. Theory & Psychology. **7** (4), 483–503 (1997), https://journals.sagepub.com/doi/abs/10.1177/0959354397074003?casa_token=e93KxyIcrLAAAAAA:blULmkIEKBba9N02SwExd5MW70G25hsphT7NuJ_4W8428LgUnYEdYIyTfXfB28RsbutdrmNionQ.

[3] Reiter, R.: A logic for default reasoning. Artificial intelligence. **13** (1-2), 81–132 (1980), https://www.sciencedirect.com/science/article/abs/pii/0004370280900144.

[4] Antoniou, G.: A tutorial on default logics. ACM Computing Surveys (CSUR). **31** (4), 337–359 (1999), https://dl.acm.org/doi/abs/10.1145/344588.344602.

[5] Lifschitz, V.: Success of default logic. Logical Foundations for Cognitive Agents: Contributions in Honor of Ray Reiter. Springer. 208–212 (1999). https://link.springer.com/chapter/10.1007/978-3-642-60211-5_16.

[6] Ben-David, S., Ben-Eliyahu, R.: A modal logic for subjective default reasoning. Proceedings ninth annual ieee symposium on logic in computer science. 477–486 (1994). https://ieeexplore.ieee.org/abstract/document/316043?casa_token=2bdkFrzSyu8AAAAA:hJi-u_pifVOPnOqB_-ydXKHOllcOksbHSRCOhUs9VElNKpwIV5qUbfvnxPoMBOT118ei-Fyf.

[7] Avramides, A.: Knowing others as persons. Inquiry. **67** (4), 1125–1147 (2024). https://www.tandfonline.com/doi/full/10.1080/0020174X.2020.1837235?casa_token=36HS_9iZdpoAAAAA%3Ajx5xTYRmQXfhTbP7LjwWVBp5JWF1bvS4LiTXBGlRQkdYOQgSr-1vYF6zehNe1gaEFExUv84iYmU.

[8] Caudal, P., Roussarie, L.: Aspectual viewpoints, speech act functions and discourse structure. In Aspectual inquiries. Dordrecht: Springer Netherlands. 265–291 (2005). https://link.springer.com/chapter/10.1007/1-4020-3033-9_12.

[9] Howell, R.J.: Self-awareness and the elusive subject. Oxford University Press. (2023). https://books.google.co.uk/books?hl=zh-CN&lr=&id=i16vEAAAQBAJ&oi=fnd&pg=PP1&dq=Howell,+R.J.+(2023).+Self-awareness+and+the+elusive+subject.+Oxford+University+Press.&ots=GWTOpLl-Ts&sig=oody4yWYtaSdUTuRoF_nh-Mfq0o&redir_esc=y#v=onepage&q&f=false.

[10] Guiraud, N., Longin, D., Lorini, E., Pesty, S., Rivi'ere, J.: The face of emotions: a logical formalization of expressive speech acts. 10th internationale conference on autonomous agents and multiagent systems (aamas 2011). Vol. 3, 1031–1038 (2011). https://inria.hal.science/hal-00950871/.

A Defense of Modus Ponens as a Valid Rule of Reasoning

Xue Ge

Sun Yat-sen University & Durham University

Abstract

Modus ponens is one of the most basic reasoning rules in both formal logical systems and argumentation. Like conjunction and disjunction in logic, it serves as a formal representation of the abstract laws of human thought. It is a widely used reasoning rule, applied explicitly in disciplines such as logic and computer science, and implicitly in daily reasoning, writing, dialogue, and thought processes. However, in 1895, Lewis Carroll put forward the "Carroll Paradox" [3] arguing that any reasoning rule used in the process, including modus ponens, would lead to an infinite regress in reasoning. This sparked debates in the field of logic about whether modus ponens is a valid rule of reasoning, especially in the face of challenges from the philosophy of non-classical logic.

In this paper, we will defend the validity of modus ponens as a fundamental rule of reasoning. We begin with the definition of the validity of modus ponens: it is a truth-preserving inference rule, meaning that if the premise $p \to q$ and its antecedent p are both true, then the succedent q is not false [10] [4]. Only cases where true premises yield a false conclusion serve as effective demonstrations of its invalidity. To provide a more comprehensive defense of the validity of modus ponens, we classify and analyze various types of counterexamples, including those involving indicative conditionals [8], counterfactuals [7] [11], and subjunctive conditionals [3]. Although some scholars have offered defensive responses to certain counterexamples [9] [5] [1] [6], no existing work has provided a systematic refutation of all types. Moreover, several important aspects of these counterexamples have not been fully analyzed or addressed in the literature. This paper fills this gap by classifying the known counterexamples into distinct types, offers new insights into overlooked issues and examining, in detail, the specific reasons why each of them fails. In doing so, the paper argues that both critics and defenders of the validity of modus ponens must first clarify the meaning of validity using methods from analytic philosophy to engage in a meaningful discussion. Since validity is a logical term, this clarification should be made from a logical perspective and adhere to logical consistency.

Overall, the defense presented in this paper consists of the following points. First, the misunderstanding of the validity of modus ponens and lacking sufficient justification due to the subjectivity of rational cognition frameworks and their inability to provide clear, objective criteria for evaluating logic. Second, the inherent vagueness and ambiguity of natural language, misinterpreting non-contradictory statements as contradictory or making inferences based on incorrect or incomplete conditional statement. This underscores how much of the debate surrounding modus ponens is rooted in the complexities of natural language and the failure to articulate propositions rigorously, rather than flaws in the logical rule itself. Third, placing the premises and conclusion in different contexts undermines the principle of identity. Upon formalizing such a counterexample, it becomes evident that the succedent of the conditional statement prior to the contextual shift and the consequent after the shift cannot be denoted by the same symbol. As a result, the falsity of the succedent cannot be established, and thus the invalidity of modus ponens cannot be proven. Lastly, intriguing discussions about artificially constructed possible worlds, which they mistakenly attribute to problems with modus ponens. This both constitutes a violation of the principle of identity and unnecessarily complicates the issue.

This defense concerns the very foundations of logic as a discipline. Modus ponens is akin to an axiom within logical systems, abstracted from natural language into a universal formal structure. Therefore, modus ponens serves as an indispensable foundation for reasoning, argumentation, and the construction of formal proofs, and such a fundamental logical principle should not be subject to skepticism.

1 Introduction

Ancient Greek skeptics believed that all principles must be proven through other means and that nothing in the world is self-evident. So, does modus ponens, which serves as a starting point for logical reasoning, also require proof to establish its validity, as the skeptics would argue? Looking at the research results of previous scholars, it is clear that they are based on different philosophical backgrounds, for example, such as epistemology and possible worlds, but more often due to the ambiguity of natural language expressions. It should be noted that the scholars who have proposed counterexamples have not completely denied the validity of modus ponens. They all acknowledge that, in certain specific contexts, the validity of modus ponens in formal logic—where a valid argument guarantees that, in every case where all premises are true, the conclusion must also be true—then the focus shifts to derive and verify conclusions based on the axioms and rules within the logical system. This process involves determining whether the conclusion necessarily follows from the premises, rather than assessing its truth or falsity in terms of external semantics. In this case, the modus ponens rule is obviously a valid reasoning rule. However, if we step away from a fine-grained formal concept of validity and instead adopt other coarse-grained concepts rooted in subjective judgments of understanding and belief within the context of natural language, then the validity of modus ponens indeed becomes a question open to challenge.

Modus ponens, also known as the rule of detachment or the rule of affirming the antecedent or the conditional elimination rule, dates back to ancient Greece [1] . In *The Oxford Dictionary of Philosophy* and *The Philosopher's Dictionary*, modus ponens is defined as "Any argument taking the form: If p, then q; p; Therefore, q." [2] and "A rule of correct deduction of the form: If p then q; p; therefore q." [3] That is, in the following form.

$$\frac{p \to q,\ p}{q}$$

It is important to note that when the antecedent p is false, the modus ponens rule remains valid regardless of the truth value of the succedent q. This is because, by definition, the validity of modus ponens is solely determined by the truth-table of the conditional statement. In a conditional statement, if the antecedent is false, the entire statement is considered true.

[1] It is generally believed that Theophrastus and Eudemus were the earliest Greek philosophers to clearly describe and use the modus ponens rule. See: Bobzien, Susanne, "Ancient Logic", The Stanford Encyclopedia of Philosophy (Summer 2020 Edition), Edward N. Zalta (ed.), URL=¡https://plato.stanford.edu/archives/sum2020/entries/logic-ancient/¿.

[2] Blackburn S. The Oxford Dictionary of Philosophy [M]. OUP Oxford, 2005. p.238.

[3] Martin R M. The Philosopher's Dictionary-Third Edition [M]. Broadview Press, 2002. p.200.

References

[1] Bledin, J.: Modus ponens defended. J. The Journal of Philosophy. **112**(2), 57–83 (2015), https://www.jstor.org/stable/43820889.

[2] Brogaard, B, Salerno, J.: Counterfactuals and context. J. Analysis. **68**(1), 39–46 (2008), https://www.jstor.org/stable/25597849.

[3] Lewis, C.: What the tortoise said to Achilles. J. Mind. **4**(14), 278–80 (1895), http://fair-use.org/mind/1895/04/what-the-tortoise-said-to-achilles.

[4] Hurley, P. J.: A concise introduction to logic.M. Cengage Learning. 2011.

[5] Lowe, E. J.: Not a counterexample to modus ponens. M. Thinking about Logic. Routledge. 79–84 (2018), https://www.taylorfrancis.com/chapters/edit/10.4324/9780429495687-13/counterexample-modus-ponens-lowe. .

[6] Lutskanov, R.: Is modus ponens a valid inference rule. J. BAS. Humanities and Social Sciences. **5**(2), 227–236 (2018), http://www.papersofbas.eu/images/papers/Papers-2-2018/Lutskanov-2-2018.pdf.

[7] Lycan, W. G.: MPP, Rip. J. Philosophical perspectives. **7**, 411–428 (1993), https://www.jstor.org/stable/2214132.

[8] McGee, V.: A Counterexample to Modus Ponens 1. M. Thinking about Logic. Routledge. 63–77 (2018), https://www.taylorfrancis.com/chapters/edit/10.4324/9780429495687-12/counterexample-modus-ponens-1-vann-mcgee.

[9] Sinnott-Armstrong, W. Moor, J. Fogelin, R.: A defense of modus ponens. J. The Journal of Philosophy. **83**(5): 296–300 (1986), https://www.jstor.org/stable/2026144.

[10] Smith, P.: An introduction to formal logic. M. Cambridge University Press. 2003.

[11] Wen, X. Validity Under Assumptions and Modus Ponens. C. Logic and Argumentation: 4th International Conference, CLAR 2021, Hangzhou, China, October 20–22, 2021, Proceedings 4. Springer International Publishing. 533–542 (2021), https://link.springer.com/chapter/10.1007/978-3-030-89391-0_33.

Fuzzy Logic in Ethical AI

Ziba Assadi and Paola Inverardi

Gran Sasso Science Institute, Viale Francesco Crispi, 7 - 67100 L'Aquila, Italy.
ziba.assadi@gssi.it and paola.inverardi@gssi.it

Abstract

In the context of automating ethical rules in digital machines, in this paper, we investigate the dissimilarity between possibility and probability of ethical rules and re-evaluate the possibility/probability principle introduced by Zadeh in light of ethical evidence. Then, we consider a fuzzy logic interpretation to represent the precision and fluidity of ethical behaviors.

Keywords: Ethics, Formalism, Fuzzy Logic, Possibility, Probability, Robotics

1 Ethics, Possibility or Probability

The degree of compatibility of a property with elements of a set can be best described by possibility. Some properties are technically feasible and possible, but their chance of occurrence is unlikely, or, in other words, they have a very low likelihood and are not probable. The deep-seated nature of ethical rules in natural language is imprecision, gradability, and being inclined to possibilities rather than probabilities. The system needs to know the possibility of making ethical rules happen rather than their likelihood. Moreover, uncertainty about the dependency of ethical rules confined us with the particularly challenging decision about the union of events, and it seems that choosing a powerful ethical rule is more compatible with $\mathrm{Possibility}(\cup_{i=1}^{n}(e_i)) = \vee_{i=1}^{n}\mathrm{Possibility}(e_i) = \max_{i=1}^{n} \mathrm{Possibility}(e_i)$ than with $\mathrm{Probability}\,(\bigcup_{i=1}^{n} e_i) = \sum_{k=1}^{n}(-1)^{k+1}\sum_{1\leq i_1<\cdots<i_k\leq n}\mathrm{Probability}(\bigcap_{j=1}^{k} e_{i_j})$. In the ethical context of users' privacy and their preferences, consider the following example of healthcare robotics. A user has manifested the requirement to not be undressed when the curtains are open. Let us define `Poss(x)` as the degree of possibility and `Prob(x)` as the probability that a user is dressed in x when the curtains are open, respectively. Let the user choose a threshold of 0.8 for expressing the state of being dressed and let her express the grading for her type of clothing at home according to her convenience with open curtains as follows: Poss(Tops such as T-shirt, shirt, blouse, sweatshirt, etc.)=0.4, Poss(Bottoms such as skirt, pants, leggings, capri pants, etc.)=0.5, Poss(Dresses such as sundress, evening dress, gown, etc.)=1, Poss(Sleepwear such as nightgown, robe, etc.)=0.8, Poss(Accessories such as jewelry, sunglasses, watch, etc.)=0, Poss(Others such as socks, hat, belt, tie, etc.)=0.1. Now we can observe that using the possibility is closer to reality than using probability. The chance of wearing socks is more likely than the chance of wearing a sundress in winter, while the user considers a sundress as a complete dress and socks very far from the threshold. Moreover, wearing 10 pairs of socks does not mean getting dressed; the possibility of dressing style means the maximum amount of possibilities that is assigned to the garments on the user's body.

1.1 *Inconsistency*

We believe that in the heuristic observations leading to reach the possibility/probability consistency by Zadeh[6], ethical considerations that diverge the consistency were not taken into account. Not always a reduction in possibility reduces the probability. The diminution of possible concrete choices may correspond to a diminution of probability, but human preferences are not concrete. The degree of consistency $\gamma = \Sigma_{\mathrm{x}} \mathrm{Poss(x)} \times \mathrm{Prob(x)}$ may change depending on user's changing preferences. Restricting the wardrobe of the mentioned user to a finite universe, consider the probability of her dressing style, which may vary by climate change. The probability of wearing socks in cold weather is much greater than its probability in hot weather, while the possibility has been graded a constant by the user. The same goes for the other elements of the universe, and by dropping the relation between possibility and probability due to unstable probabilities, we meet divergency in the degree of consistency, or rather, inconsistency of probability and possibility as a consequence.

2 The Need for Fuzzy Logic

Now consider that the user wants the robot to open the curtains when she is highly distressed, although she has not reached her defined threshold. Expressing ethical requirements from natural language to a formal and logic based language needs accuracy and precision. Words such as "highly" and "distressed" are not definite and obvious. "Highly distressed" is an expression in natural language that is not precise, and its possible existence depends on some circumstances [2]. For mapping imprecision to precision, we use fuzzy logic that can assign all possibilities to a degree between zero and one. Fuzzy logic has been extended to handle the concept of partial truth, where the truth value may range between completely true and completely false. According to the non-absolute and partially true nature of ethics, we propose fuzzy logic, which can handle infinite degrees of truth. We tried to formalize ethical rules according to the fuzzy-rule based system by changing the pivotal logical connective and avoiding ambiguity [3]. [1] discusses the results of the implementation of a questionnaire composed of some scenarios with ethical implications that users may encounter in their daily life to make decisions. Users reply to scenarios by *Yes* or *No* and justify it by weighting four parameters. Consider the following scenario (*Fruit scenario*) and its four justification parameters. *There are trees with ripe fruit in a private park with private access. The gate is open, and there are no people around. Do I go in and steal some? p_1:How much did the potential consequences of the action on others weigh on my choice? p_2: How much did the potential consequences of the action on me weigh on my choice? p_3: How much did my personal experiences weigh on my choice? p_4: How much did respect for the law weigh on my choice?* [5] has reported two questionnaires of [1] and implicitly employed possibility rather than probability in weighting their parameters. Users can weight the parameters as their preferences (possibilities). The scenario may stress on one or some of the parameters and the singleton set of parameters that the scenario puts the most pressure on, is the maximum of parameters. Weighting parameters justifies users `Yes` or `No` answers, and this justification is judged by soundness checking through the soundness function $\mathrm{SOUNDJ(SCENARIO, RESPONSE, JUSTIFICATION)} \equiv \{\top, \bot\}$. In *Fruit scenario*, consider two justifications of parameters as follows, where weights can be integers ranging from 1 to 5: $(1, 1, 1, 1)$ and $(5, 3, 2, 5)$. $\mathrm{SOUNDJ(Fruit, No, (1, 1, 1, 1))} \equiv \top$ elicits answer `No` and $\mathrm{SOUNDJ(Fruit, Yes, (5, 3, 2, 5))} \equiv \bot$ has no elicitation. Our Fuzzy approach is not limited to binary outcomes. For instance, truth values could come from a more nuanced set of options, such as true, not true, very true, less true, almost true, and more. We improve the process by fuzzy logic and reduce the complexity by refining the soundness verification. A fuzzy system is composed of fuzzification, a section of fuzzy rules and inference engine, and defuzzification. In fact, in our mentioned *Fruit scenario*, decision-making depends on the result of the defuzzification step, and there is no need for sound justification. The fuzzification process assigns some linguistic variables such as *Low*, *Medium*, and, *High* to each possibility by the suitable (we propose the trapezoidal) membership functions. For the *Fruit scenario*, in case of weighting parameters on the same scale, the membership functions of four parameters in the states of low, medium, and high are as follows:

$$\mu_{p_{ij_{Low}}}(x) = \begin{cases} 0 & x < 0 \\ \frac{x-0}{1-0} & 0 \le x < 1 \\ 1 & 1 \le x \le 2 \\ \frac{4-x}{4-2} & 2 < x \le 4 \\ 0 & x > 4 \end{cases} \qquad \mu_{p_{ij_{Medium}}}(x) = \begin{cases} 0 & x < 1 \\ \frac{x-1}{3-1} & 1 \le x < 3 \\ 1 & 3 \le x \le 3 \\ \frac{5-x}{5-3} & 3 < x \le 5 \\ 0 & x > 5 \end{cases}$$

$$\mu_{p_{ij_{High}}}(x) = \begin{cases} 0 & x < 2 \\ \frac{x-2}{4-2} & 2 \le x < 4 \\ 1 & 4 \le x \le 5 \\ \frac{6-x}{6-5} & 5 < x \le 6 \\ 0 & x > 6 \end{cases}$$

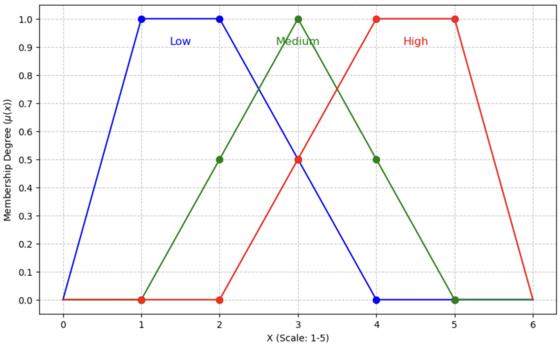

Fig. 1. Fuzzy Trapezoidal membership functions

In this scenario, we can have at most 3^4 (four parameters and three linguistic variables) fuzzy rules based on our proposed trapezoidal membership functions such as $R_i : \mu_{p_{i1}} \wedge \mu_{p_{i2}} \wedge \mu_{p_{i3}} \wedge \mu_{p_{i4}} \to \mu_{p_{i5}}$, where $\mu_{p_{ij}} \in \{\mu_{p_{ij_{Low}}}, \mu_{p_{ij_{Medium}}}, \mu_{p_{ij_{High}}}\}$ for $\{i = 1, \cdots, n\}$ (with $n \le 3^4$). Since our fuzzy control rule is a function of input parameters, we use the defuzzification method introduced by [4] and when the user in a similar scenario weights these parameters as $\{w_1, w_2, w_3, w_4\}$, the defuzzification process gives a crisp value as $\frac{\Sigma_{i=1}^{n} \min\left(\mu_{p_{i1}}(w_1),\mu_{p_{i2}}(w_2),\mu_{p_{i3}}(w_3),\mu_{p_{i4}}(w_4)\right) \times \mu_{p_{i5}}(w_1,w_2,w_3,w_4)}{\Sigma_{i=1}^{n} \min\left(\mu_{p_{i1}}(w_1),\mu_{p_{i2}}(w_2),\mu_{p_{i3}}(w_3),\mu_{p_{i4}}(w_4)\right)}$ and interpretation of this value to a linguistic output can decide an answer `YES` or `NO` for the user.

References

[1] Alfieri, C., Donati, D., Gozzano, S., Greco, L., Segala, M.: Ethical Preferences in the Digital World: The EXOSOUL Questionnaire. In: HHAI 2023: Augmenting Human Intellect - Proceedings of the Second International Conference on Hybrid Human-Artificial Intelligence, IOS Press, Frontiers in Artificial Intelligence and Applications, vol. **368**, pp. 290–299 (2023), https://doi.org/10.3233/FAIA230092

[2] Assadi, Z.: Logical Formalisms for Ethics. GoodIT'24: Proceedings of the 2024 International Conference on Information Technology for Social Good, pp. 416–419 (2024), https://doi.org/10.1145/3677525.3678691

[3] Assadi, Z. and Inverardi, P.: Fuzziness of Ethics: towards overcoming some dilemmas (2024), https://dx.doi.org/10.2139/ssrn.5035017

[4] Berenji, H.R.: Fuzzy Logic Controllers. In: Yager, R.R., Zadeh, L.A. (eds.) An Introduction to Fuzzy Logic Applications in Intelligent Systems, pp. 69–96. Kluwer Academic Pub (1992), https://doi.org/10.1007/978-1-4615-3640-6_4

[5] Donati, D., Assadi, Z., Gozzano, S., Inverardi, P., Troquard, N.: On Representing Humans' Soft-Ethics Preferences As Dispositions. Italia Intelligenza Artificiale (Ital-IA), pp. 135–140 (2024), https://ceur-ws.org/Vol-3762/551.pdf

[6] Zadeh, L.A.: Fuzzy sets as a basis for a theory of possibility. Fuzzy Sets and Systems **1**(1), 3–28 (1978), https://doi.org/10.1016/0165-0114(78)90029-5