# Secondary Publication

**Steen, Alexander; Benzmüller, Christoph**

## Challenges for Non-Classical Reasoning in Contemporary AI Applications

Date of secondary publication: 15.10.2025

Version of Record (Published Version), Article

Persistent identifier: urn:nbn:de:bvb:473-irb-110814x

**Primary publication**

Steen, Alexander; Benzmüller, Christoph (2024): Challenges for Non-Classical Reasoning in Contemporary AI Applications, in: Künstliche Intelligenz : KI, Berlin: Springer Nature, Vol. 38, Nr. 1–2, pp. 7–16, doi: 10.1007/s13218-024-00855-8.

**Legal Notice**

This work is protected by copyright and/or the indication of a licence. You are free to use this work in any way permitted by the copyright and/or the licence that applies to your usage. For other uses, you must obtain permission from the rights-holders.

This document is made available under a Creative Commons license.

The license information is available online:
https://creativecommons.org/licenses/by/4.0/legalcode

KI - Künstliche Intelligenz (2024) 38:7–16
https://doi.org/10.1007/s13218-024-00855-8

TECHNICAL CONTRIBUTION

# Challenges for Non-Classical Reasoning in Contemporary AI Applications

**Alexander Steen**[1] · **Christoph Benzmüller**[2]

Received: 5 May 2024 / Accepted: 19 June 2024 / Published online: 4 July 2024
© The Author(s) 2024

**Abstract**
In knowledge representation and reasoning, a key area in artificial intelligence research, non-classical logics play a prominent double role: firstly, non-classical logic languages allow for a precise and transparent encoding of domain specific knowledge. Secondly, as the logical languages are equipped with custom-tailored rules of logical inference, they make available a principled approach to derive new knowledge from previous information. In practice, the first aspect addresses data storage and retrieval, the second aspect the utilization of available information. This article briefly surveys contemporary challenges of NCL research in AI.

## 1 Introduction

*Non-classical logics* (NCLs), in short, are logic formalisms that deviate from classical (propositional or predicate) logic in one way or another, e.g., by refusing bivalence of truth-values, truth-functionality of logical connectives, monotony of consequence, or some other of many possible options. Prominent non-classical logic families include:

- temporal logics, e.g., linear time temporal logic (LTL) [41, 64], and (full) computation tree logic (CTL/CTL*) [33, 34],
- (dynamic) epistemic logics, e.g., public announcement logic (PAL) [63], epistemic action logic (EAL) [3, 4], logic of communication and change (LLC) [83],
- non-monotonic logics, e.g., default logic (DL) [36, 68],
- deontic logics [39, 40], and
- modal and description logics [2, 20].

In knowledge representation and reasoning, a key area in Artificial Intelligence (AI) research, NCL formalisms are often preferred over classical logics. This is because NCLs are usually domain-specific formalisms designed to provide concise means to express expert knowledge in the given context (which would otherwise need to be elaborately circumscribed with classical logic means). In AI research, NCLs play a prominent double role: Firstly, they allow for a precise, natural and transparent encoding of domain specific knowledge. Secondly, they make available a principled approach to derive new knowledge from previous information using logical reasoning (i.e., rules of inference). In practice, the first aspect addresses data storage and retrieval, the second aspect the utilization of available information in AI systems.

Long standing international conferences witness the sustained interested in NCLs (and logic-based AI methods in general), including specialized conferences such as *Principles of Knowledge Representation and Reasoning (KR)*,[1] the *International Conference on Automated Planning and Scheduling (ICAPS)*,[2] the *International Workshops on Non-Monotonic Reasoning (NMR)*,[3] the *International Workshops on Description Logics (DL)*[4] and the *International Conferences on Non-Classical Logics: Theory and Applications (NCL)*; and NCLs are also in the topical portfolio of general logic automation conferences, including the *International Joint Conferences on Automated Reasoning (IJCAR)*[5] and

✉ Alexander Steen
alexander.steen@uni-greifswald.de

Christoph Benzmüller
christoph.benzmueller@uni-bamberg.de

[1] Universität Greifswald, Walther-Rathenau-Straße 47, 17489 Greifswald, Germany

[2] Otto-Friedrich-Universität Bamberg, An der Weberei 5, 96047 Bamberg, Germany

[1] See https://kr.org/.
[2] See https://www.icaps-conference.org/.
[3] See https://nmr.krportal.org/.
[4] See https://dl.kr.org/.
[5] See https://ijcar.org/.

major general AI conferences such as the *International Joint Conferences on Artificial Intelligence (IJCAI)*[6] and the *European Conferences on Artificial Intelligence (ECAI)*.[7] Relevant journals include the *Journal of Applied Non-Classical Logics (JANCL)*, the *Journal of Automated Reasoning (JAR)*, and the various general AI journals.

In logic-based AI research, it seems, the development of reasoning tools is still underrepresented. This clearly impedes the practical application of these methods in more complex AI systems.

In this brief survey, some topical applications of NCLs in AI research are highlighted, and relevant challenges for non-classical reasoning in AI are given. The article is not intended as an introduction nor an historical overview of the (early) developments in non-classical logics. A comprehensive collection of historical information is compiled in the Handbook series on the History of Logic [35], and introductions to non-classical logics are available elsewhere, e.g., by Wansing (as editor) [85], by Priest [65], and by Wasilewska [86].

## 2 Logical Reasoning in and for AI

The automation of symbolic, logical reasoning has been a central focus of AI research since its inception. In the early days of AI, until the advent of powerful subsymbolic machine learning solutions, the automation of symbolic reasoning played a pivotal role in numerous leading AI research centres worldwide. At present, there is a resurgence of interest in logical reasoning, particularly in light of the limitations of pure subsymbolic reasoning. This is evidenced by the growing interest in so-called hybrid or neuro-symbolic AI systems; see [82] for an exemplary recent success story on a hybrid system integrating a large language model with a symbolic, logical reasoning engine. The pros and cons of both sides of the spectrum are prominently discussed in a recent paper by Lenat and Marcus [56], and as stated in [11], it is *"The vision of strong AI, that is, AI that surpasses human capabilities in all or almost all domains,* [that] *requires,* [...] *the hybridization of techniques from both sides (or a convincing explanation of why symbolic reasoning techniques should suddenly and readily evolve from data-driven subsymbolic techniques)."* Rather than delving further into developments in hybrid AI or neuro-symbolic AI and the debate around them (see e.g. [18, 59, 87] for overviews), here we briefly review some exemplary recent or prospective application areas of symbolic, logical reasoning in AI and beyond.

Classical automated and interactive theorem provers [9, 71], SAT [19, 58], SMT [7] and QBF [75] solvers have applications in AI planning, formal software & hardware verification, bioinformatics, and in various other domains; e.g., SAT solvers have been able to solve open maths problems [23, 52], and first-order and higher-order ATPs found applications even in philosophy [1, 14].

Similarly, logic programming [55], including Datalog and answer set programming [5, 24], inductive programming and inductive logic programming [28, 47, 67] have applications in areas such as robotics, vision and games, to mention just a few.

Description logics [2] are used, among other things, for modelling and reasoning with web ontologies. They have many other applications, e.g. in the medical and legal domains [73].

Multi-modal and other non-classical logics have been used, among other, for the modeling of and the reasoning with and about multi-agent systems [37, 46, 84].

*"The Need for Good Old Fashioned AI and Law"* [8] has been defended by Bench-Capon and substantiated by a recent study [27] using an implementation of an abstract reasoning framework [25]. Related is the objective to develop "Logics for New-Generation AI"[8] which, among others, aims at exploring novel bridges between automated reasoning and abstract argumentation [42].

Particularly relevant for normative and legal reasoning, where so called contrary-to-duty paradoxes may arise, are deontic logics [39, 40]. And more general inconsistency robust solutions are being studied in the area of paraconsistent and paracomplete logics [66], which have applications e.g. in robotics, intelligent control, decision-making systems, pattern recognition and classification, and medical diagnosis [30].

With reference to the aforementioned trend towards hybrid or neuro-symbolic AI, there is now a growing interest in exploring logics for exlainable AI [72].

## 3 A Practical Challenge: Automation of NCLs

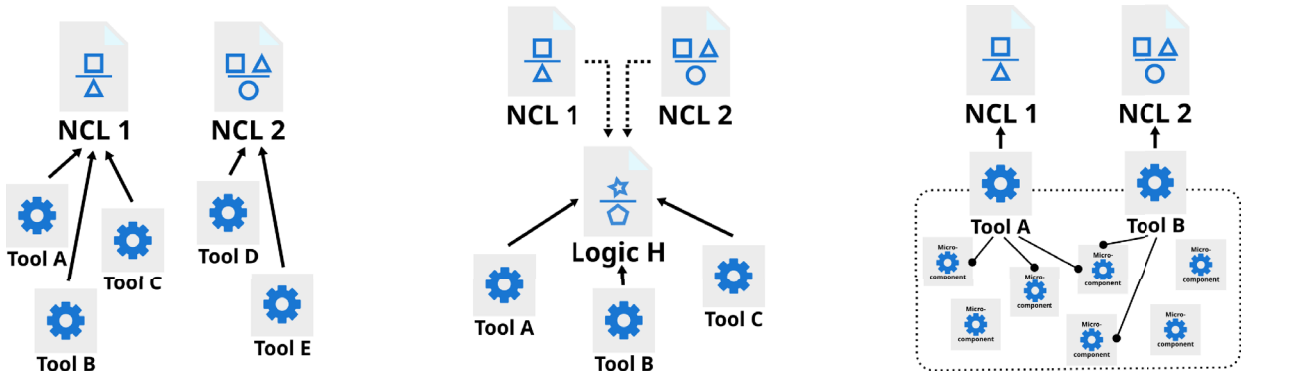
It seems that many interesting NCLs are not adopted within concrete AI systems (presumably) because there is simply no automation for many (if not most) of these logics.[9] Relevant

[6] See https://www.ijcai.org/.

[7] See https://www.eurai.org/ecai.

[8] See the international research project website at https://xixilogic.org/lngai/.

[9] It has to be noted that, of course, there exist automation methods and tools for various relevant NCLs, in particular, for propositional NCLs. Still, this does not invalidate the argument as there seem to be at least as many NCLs for which no automation exists, probably even many more (but this is difficult to acertain objectively). Note also the discrepancy between the theoretical focus on the study of individual




**A.1**) (b) Automation by Translation (**A.2**) (c) Automation by Composition (**A.3**)

**Fig. 1** Overview of three conceptually different automation approaches. Solid arrows indicate an *is-implemented-by* relation between reasoning tools and NCLs. Dotted arrows indicate an *is-translated-to* relation from one logic to another logic. Solid lines with an circular endpoint denote *is-component-in* relations between micro-service components and composite reasoning tools

contributions to logic-based AI methods, e.g., as regularly presented for multi-agent systems at IJCAI conferences, often remain on a theoretical level (e.g., providing completeness results) and hence cannot be easily leveraged into more complex software systems, let alone be empirically tested and assessed for adequacy on a larger scale than small motivating examples ("toy examples").

It is not surprising, then, that automation usually only exists for logics which are known to be mature and stable: classical propositional and predicate logics were thoroughly researched and refined into today's stable state, and they proved an adequate tool for many fields in mathematics and computer science; consequently, for classical logics, there exist many different tools for automation.[10] There are also various tools for well-established NCL families such as separation logics [61, 69], modal logics and temporal logics, including the separation logic-based program verifiers Smallfoot [17] and VeriFast [54], the modal logic provers LoTREC [32] and MetTeL [81], and the LTL resp. CTL-based model checking systems SPIN [53] and NuSMV [26].

For many other NCLs the situation is quite different: They are not as mature and stable, and it's not clear yet which logic formalism variant is best suited for a specific context (e.g., in deontic logics). Also, the research community is continuously developing new formalisms, and also further developing existing formalisms with various extensions. As a consequence no "fixed standard NCL" (for a given purpose), or a small set of those, has yet been agreed on.

This constitutes a classic *hen-and-egg problem* for logic automation: Who will build a reasoning tool – which is usually a very laborious thing to do – for a logic for which it is not even clear yet whether it is the "right choice" for the task at hand? The logic may be superseded next year by another generalization, extension and/or replacement. The other way around, who can conduct an in-depth assessment on the adequacy of a logic formalism in the context of a medium- to large-scale application scenario, if there is no tool support available for that specific logic? Such an assessment could also uncover interesting pragmatics of the NCL. As an example, in contrary to known theoretical considerations (and known issues), the logic may actually work well enough in practice for some scenarios, or there may be issues that have been overlooked so far as they materialize in more complex scenarios only.

This state of affairs hampers the beneficial employment of symbolic AI techniques, and also creates non-trivial obstacles for interested audience that wishes to test such approaches (e.g., in hybrid/neurosymbolic AI methods).

A relevant challenge is thus the following:

**Challenge #1: Feasible Automation**

*How can we develop feasible (efficient, robust and maintainable) reasoning procedures and tools for many different non-classical logics?*

In principle, there are (at least) three different options how to address this practical challenge:

**A.1** Individual automation approach
**A.2** Automation by translation approach
**A.3** Automation by composition approach

Footnote 9 (continued)

propositional NCLs and the need for more expressive and combined NCLs in many real world applications.

[10] See e.g. the various classical reasoners as available at SystemOnTPTP (https://tptp.org/cgi-bin/SystemOnTPTP).

The three options are visualized in Fig. 1. Each approach comes with pros and cons, which are summarized next.

### 3.1 Individual Automation

The approach referred to as *individual automation* here seems to be the predominant approach for logic automation to date: For each logic formalism (or logic family, if possible) specific software tools are implemented, including the design development of specialized calculi. Different tools may use different (specialized) data structures, are implemented in different programming languages, possibly using different input/output formats.

*Advantages.* There are evident advantages of this approach:

+ The individual tools are (usually) optimized for their particular application use case.
+ The input/output formats are custom-tailored and hence (often) concise.
+ The development process is done locally (e.g., in a research group), so there is little communication overhead and little need to compromise about design choices.

*Disadvantages.* However, individual automation comes with several serious downsides:

- The development of an individual reasoning tool is expensive, and likely to take many years before a stable implementation is available.
- Each system will potentially use its own input and output standard, which is cumbersome and error-prone for the integration in larger software.
- The individual tools come with a varying quality of documentation resources, and their usage, limitations and parameters may not be clear.
- Individual tools, if developed in the scope of research projects, have the tendency to become neglected and unmaintained after some time, making it unattractive for third parties to consider using it.

At a first glance, it seems that in this approach the disadvantages outweigh the advantages. Nevertheless, the impact of a simple (local) process management and an agile development process should not be underestimated in practice.

Long-time existing and successful reasoning systems of the *individual automation* kind include the automated theorem provers E [74] and Vampire [70] for classical predicate logic; the satisfiability-modulo-theories (SMT) solvers CVC5 [6] and Z3 [31]; the description logic reasoners RacerPro [50], HermiT [45] and Konclude [80], and the answer set solver clasp [43].

### 3.2 Automation by Translation

The approach here termed *automation by translation* describes an indirect automation method: Reasoning in some logic $O$ (the *object* logic) is reduced to reasoning in some other logic $H$ (the *host* logic). This is done by translating the $O$-expressions into "adequate" $H$-expressions in such a way that the translation preserves correctness conditions (e.g., soundness and completeness with respect to $O$). Well-known examples of such reductions are the so-called *standard translation* of modal logics into classical predicate logic [62], of propositional intuitionistic logic into modal logic **S4** using the Gödel-McKinsey-Tarski translation [49, 60].

In general, such translations may employ *shallow embeddings* or *deep embeddings* [44]. Shallow embeddings directly encode the semantics of $O$ as terms of $H$, thus enabling off-the-shelf automation with (usually) fair reasoning performance [10]. As downside, shallow embeddings hide the syntactic structure of $O$-expressions so that explicit reasoning over $O$-terms or $O$-formulas is not easily possible. On the other hand, deep embeddings encode the syntactic expressions of $O$ within $H$ and then provide an evaluation function in $H$ to map these expressions to some $H$-denotation. This makes explicit in $H$ the syntactic structure of $O$ objects, but severely hampers automation performance (due to complex inductive definitions and recursive evaluation functions). Both the standard translation and the Gödel-McKinsey-Tarski translation are shallow embeddings. Deep embeddings are often found in formalization libraries of proof assistants (see, e.g., [21, 38, 48]).

*Advantages.*

+ The implementation of a translation is usually much simpler than the implementation of a stand-alone reasoning tool.
+ Existing well-engineered reasoning tools can be re-used for the actual automation.
+ Automation is not bound to any one specific reasoning tool for logic $H$, and users may chose which $H$-tool they prefer (or performs better).
+ Automation performance for $O$ automatically benefits from improvements of tools for logic $H$.
+ Input/output formats and documentation resources of established systems may be reused.
+ It is easier to automate the integration of different $O$-logics.
+ The approach also applies to quantified (e.g., first-order) reasoning.

*Disadvantages.*

- Depending on the logic $H$, theoretical complexity bounds of $O$ are abandoned and cannot easily be exploited as reasoning in $H$ may be less efficient (or even undecidable) in general.
- The $H$-tools will not be optimized for reasoning on $O$-inputs.
- Proofs found by $H$-tools are proofs formulated in $H$ modulo the translation from $O$. It may be complicated to reconstruct proper $O$-proofs (if relevant).

A key insight here, potentially outweighing the disadvantages, is that automation by translation can serve as an intermediate method for rapid prototyping of logic formalisms, and for medium-scale assessments in concrete applications. If, after careful assessment, some NCL is found to be adequate for the given use case, a dedicated reasoning tool can be developed to supersede the translation-based approach (using one of the other two automation approaches). In particular, the translation-based approach yields simple means of automation for logics where otherwise no automation is available at all.

Automation by translation is intensively exploited by the authors in the LogiKEy methodology [15] and the logic embedding tools [76, 79], which make use of classical higher-order logic (HOL) as host logic $H$. Maybe somewhat surprisingly, even with an undecidable host logic like HOL, the automation performance for many different quantified modal logics is comparable to, and sometimes even better, than using native modal logic reasoning systems [78].

### 3.3 Automation by Composition

Instead of developing specialized, monolithic reasoning tools, the *automation by composition* approach aims at providing many small tools with limited functionality each. These tools are then composed into full-functioning reasoning systems (cf. a microservice architecture from software engineering). In a compositional setting, ideally, reasoning systems for different logics would be constructed by combining these different components in a specific way. Ideally, assertions and properties of the individual components translate into suitable correctness properties of the composite reasoning system.

*Advantages.*

+ New reasoning tools can be created by composing available (micro-)components, which increases re-use of stable, established and mature libraries.
+ New components can be developed much faster than big, monolithic systems.
+ Each individual component can be thoroughly specified and documented more easily.
+ Existing composite systems can be adopted to other needs by exchanging one or more individual components.

*Disadvantages.*

- It is unclear how general interfaces and capabilities of general purpose components should look like in order to allow flexible reasoning combinations.
- A composite reasoning tool made up from individual components will not be as efficient as a specialized monolithic tool.
- The input/output standards of such tools would need to be very general in order to capture many different application scenarios, potentially leading to bloated and cumbersome representation and/or file formats.

The reasoning by composition approach has been the focus of attention of Dagstuhl seminar 23471 ("*The Next Generation of Deduction Systems: From Composition to Compositionality*")[11] for addressing what the seminar organizers referred to as "*software crisis*" [22] in automated reasoning technology. In particular, the following insights are mentioned by Bonacina et al. [22]:

> *"The existing dichotomy, between short-lived prototypes and powerful, but big, monolithic, unwieldy systems, was discussed as an automated reasoning software crisis. The need for modularity was recognized, and a distinction between industry powertools and pedagogical platforms was outlined. The latter will have to give up on a unique programming language and programming style, as well as on award-winning efficiency, but will facilitate the entrance of new students, currently discouraged by the impossibility of competing with established tools. Thanks to such platforms, the building of new systems will be less expensive in terms of human time and labor. The risk of new ideas being forgotten without having been properly implemented and tested will be reduced."*

Up to the authors' knowledge, no comprehensive collection of reasoning microservices exists at the time of writing of this article. Still, ventures such as the Logic ToolKit [29] or Deduction-as-a-Service [51] seem to point in the right direction, if carefully designed at maintained for long-term use.

[11] See https://www.dagstuhl.de/23471.

## 4 A Communication Challenge: Zoo of NCLs

As outlined further above, every year at major AI conferences, and in relevant journals, new NCL formalisms are presented; usually extending or generalizing earlier systems, but also introducing new NCLs. As an example, at the major IJCAI conferences of the last six years, **89** NCL formalisms have been presented in the "Knowledge Representation and Reasoning" category[12]:

- IJCAI 2023: 12
- IJCAI 2022: 16
- IJCAI 2021: 20
- IJCAI 2020: 9
- IJCAI 2019: 13
- IJCAI 2018: 19

Of course, the actual number of new formalisms is much higher. Firstly, many NCLs are introduced in the multi-agent systems category of IJCAI (not counted here); secondly, they are also regularly presented at conferences resp. workshops such as KR, NMR, or IJCAR (to only name a few); thirdly, journal articles of the last years are not even considered here. The general picture should become clear nevertheless: Many NCLs are proposed and published every year, at different venues, for different application areas.

A relevant strategic challenges for NCL research in AI is thus the following:

**Challenge #2: Communication about NCLs**

*How can we improve the presentation, communication and teaching about NCLs in AI and beyond?*

This seems to be primarily a topic of science and research strategy and policy making, but also has scientific methodological dimensions.

Firstly, the *internal* communication strategy dimension of the challenge concerns how NCL development and use in AI is perceived within the community. In particular, there may be need to discuss about how to structure and array the different research strands, and how to make them more accessible to newcomers (e.g., prospective students), allowing them to navigate through the large amount of different NCL formalisms without being discouraged from the vast amount of (seemingly unrelated) systems. For established research areas handbooks or textbook series often meet this task; but it is an open question whether this is a feasible route for the dynamic and fast-paced research area of NCLs in AI (at least for those fields where the formalisms are not settled upon, yet).

Secondly, the *external* communication strategy dimension of the challenge concerns how logic-based AI methods are perceived from outside of the community. It seems likely that simpler access to automation methods and tools, as discussed in Sec. 2, might increase the visibility of NCL developments within other areas of AI research, and fosters collaboration across the board. Strong evidence is provided by fields such as description logics and answer set programming (where many such automation tools exists).

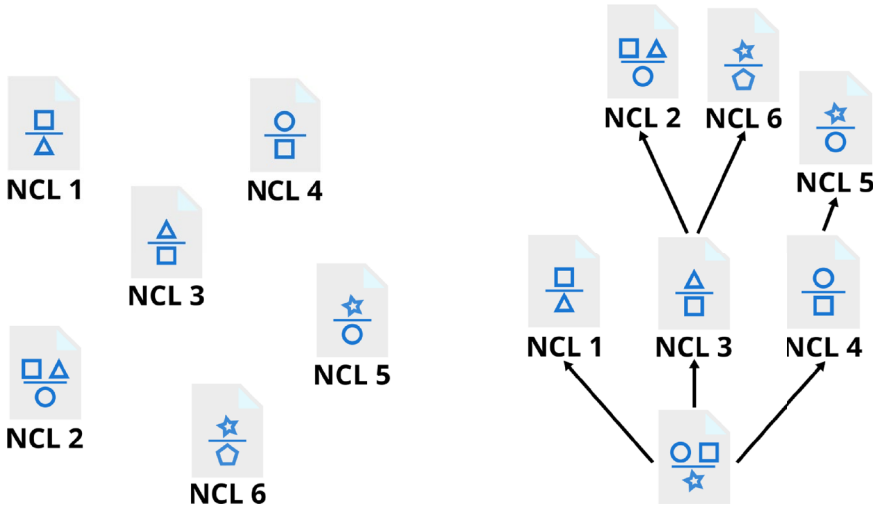
Both dimensions above are not necessarily distinct from topical research questions. As pointed out by Leon van der Torre in this special issue's interview [77], and prominently put forward by David Makinson in his textbook "Bridges from classical to nonmonotonic logic" [57], communication about NCLs and their conceptual development can go hand-in-hand: In his book Makinson starts from classical logic, and from there develops different non-monotonic inference relations using classical logic notions. This clarifies the connection of both systems, but also provides an intuitive development process (the *bridge*) from one system to another. The former aspect might provide interesting technical insights, the latter makes non-monotonic notions of inferences much more understandable to students.

Can this also be done for other NCL formalisms? The starting point need not be classical logic, but may be another NCL. Then, the field of (stable and mature) NCL formalisms could be introduced as a connected and directed graph of such Makinson-like development steps. This would, in turn, transform the unstructured *Zoo of Non-Classical Logics* into a structured *Bouquet of (consecutive) Bridges* – from the common root of classical logic, as visualized in Fig. 2. In a talk given at King's College London in around 2002, Makinson himself gave a metaphor of a solar system, where classical logic is the sun, and the different NCLs are the planets in its orbit.

## 5 An Interdisciplinary Challenge: Fragmentation of Logic

The field of logic emerged from philosophy at a time when philosophy saw itself as a universal science, with the aim of questioning the foundations of all other disciplines. This no longer seems to be the case, and over the centuries the field of logic has split into subfields of philosophy, mathematics,

[12] The numbers are collected as follows: For each IJCAI, accepted submissions in the KR &R category are considered. Then, papers are counted towards the given number if and only if their abstract explicitly mentions the introduction of a new, enhanced, augmented or otherwise adapted formalism. This is a lower bound of new NCL systems presented at IJCAI, as some may have been assigned to other categories, or their abstracts do not mention the fact explicitly.




(a) Zoo of Logics (b) Bouquet of Bridges

**Fig. 2** Methodological development of NCL formalism. On the left, NCL systems are introduces as isolated systems; on the right, NCL systems are developed via bridges (indicated by arrows) from other logics, the root possibly being classical logic

computer science, and artificial intelligence; cf. Figure 3.[13] As a result, the importance, relevance, and impact of the field as a whole is now largely unrecognised, both in public reception and in any of these fields.

This fragmentation particularly affects the development of NCL in these different areas, with the result that young researchers in particular are often unaware of relevant ongoing work in these interrelated areas of logic development. Heterogeneous terminology has emerged, different methods and techniques are preferred in education and research, different criteria are used to evaluate research, and the field as a whole does not receive adequate recognition and strategic support.

This naturally yields a third challenge:

**Challenge #3: Overcome Fragmentation**

*How can the fragmentation of NCL research and education in the fields of AI, computer science, philosophy and mathematics be mitigated?*

The challenge is therefore to promote interdisciplinary logic education and research, especially with regard to the development and use of NCLs in current application areas. The UNESCO *World Logic Day (WLD)*, proclaimed in 2019, and now annually celebrated on January 14, tries to address exactly this challenge. As written on the UNESCO WLD web page,[14] the goal is to...

> "*[...] bring the intellectual history, conceptual significance and practical implications of logic to the attention of interdisciplinary science communities and the broader public.*"

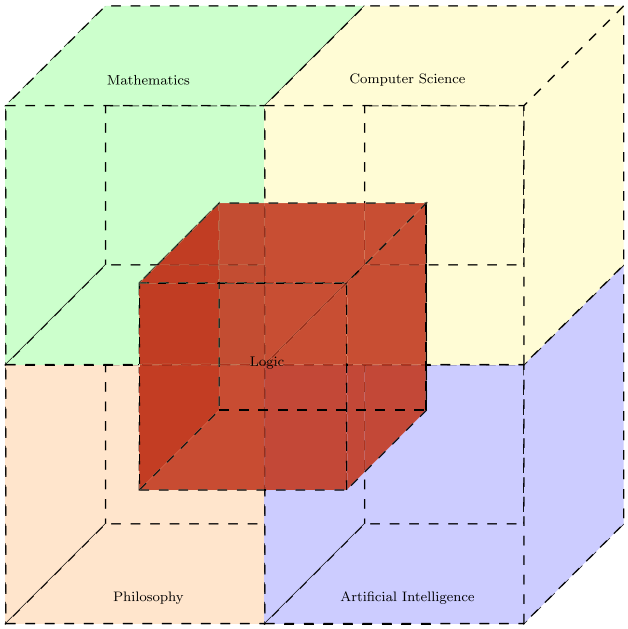



**Fig. 3** Logic as core of AI, CS, philosophy, and maths.

[13] Side remark: The LaTeX tikz code of Fig. 3 was generated by ChatGPT 4.0 in an impressive "semantical" dialogue with the second author about 3D shapes & structures, orientation, colouring, relative positioning, etc. However, despite the very good result of ChatGPT from this dialogue, there are apparently still some semantic errors in the picture, which supports the initial statements made in §2: ChatGPT is still only simulating semantic understanding, so errors may still occur.

[14] See https://unesco.org/days/world-logic.

A well-known summer school series dedicated to interdisciplinary education in that context is the *European Summer School in Logic, Language and Information (ESSLLI)*.[15] ESSLLI's concept is to offer various courses that are centered around one of the three interdisciplinary categories "language and computation", "logic and language", and "logic and computation". From the perspective of automation and tools for teaching, current developments are discussed by the established *International Workshops on Theorem Proving Components for Educational Software (ThEdu)*[16] and the *International Congresses on Tools for Teaching Logic (TTL)*.[17]

With a teaching concept in mind bringing together logic in mathematics, computer science and philosophy the authors (among others) have developed and proposed a methodology for interdisciplinary logic education using modern higher-order proof assistants [12, 13] that originates from an award winning lecture course proposal on *Computational Metaphysics* [16], which adopts a particular focus on the study, practical development and use of expressive NCLs in the context of inspiring but crisp foundational questions e.g. in philosophy. Still many open questions remain, in particular for interdisciplinary research. How can we foster the alignment of different research areas around common logic vocabulary and methodologies? How can we present logic more prominently as a field in its own right?

## 6 Summary and Outlook

Non-classical logics, and knowledge representation and reasoning, are a core topical area of AI research. While subsymbolic AI approaches are currently ubiquitous in media and also in the general AI research landscape, symbolic approaches are still essential for reasoning tasks, and – given the visible limitations of pure subsymbolic approaches – are bound to receive renewed and focused interesting in the future.

Of course, in AI research it is not about either using symbolic or subsymbolic approaches. On the contrary, hybrid approaches that make use of both sides' strengths will be essential for major progress steps in the field. For this, we need flexible automation for non-classical logic formalisms, more young scientists in NCL research, and strengthened interdisciplinary research in logic. These three aspects have been briefly highlighted in this article.

[15] See https://esslli.eu/

[16] See http://www.uc.pt/en/congressos/thedu/ThEdu24.

[17] See https://toolsforteachinglogic23.weebly.com/ for the most recent edition.

**Acknowledgements** The authors would like to thank the anonymous reviewers for their valuable comments that led to significant improvements of this manuscript.

**Funding** Open Access funding enabled and organized by Projekt DEAL.

## Declarations

**Conflict of Interest** The authors declare that they have no Conflict of interest.

**Open Access** This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit http://creativecommons.org/licenses/by/4.0/.

## References

1. Alama J, Oppenheimer PE, Zalta EN (2015) Automating Leibniz's theory of concepts. In Amy P. F and Aart M, (Eds), *Proc. of the 25th international conference on automated deduction (CADE-25)*, volume 9195 of *LNCS*, pages 73–97. Springer
2. Baader F, Calvanese D, McGuinness DL, Nardi D, Patel-Schneider PF (2003) (Eds) *The Description Logic Handbook: Theory, Implementation, and Applications*. Cambridge University Press
3. Baltag A (1999) A logic of epistemic actions. CWI, Amsterdam
4. Baltag A, Renne B (2016) Dynamic epistemic logic. In Edward N. Zalta, (Eds), *The Stanford Encyclopedia of Philosophy*. Metaphysics research lab, Stanford University, Winter 2016 edition
5. Chitta B (2010) Knowledge representation reasoning and declarative problem solving. Cambridge University Press
6. Barbosa H, et al (2022) cvc5: a versatile and industrial-strength SMT solver. In Dana F, Grigore R (Eds) *Proc. of the 28th international conference on tools and algorithms for the construction and analysis of systems (TACAS)*, volume 13243 of *LNCS*, pages 415–442. Springer
7. Barrett C, Tinelli C (2018) Satisfiability modulo theories. In: Clarke EM, Henzinger TA, Veith H, Bloem R (eds) Handbook of model checking. Springer International Publishing, Cham, pp 305–343
8. Bench-Capon T (2020) The need for good old fashioned AI and law. *Jusletter IT*, 21
9. Bentkamp A, Blanchette J, Nummelin V, Tourret S, Vukmirovic P, Waldmann U (2023) Mechanical mathematicians. Commun ACM 66(4):80–90
10. Benzmüller C (2019) Universal (meta-)logical reasoning: recent successes. Sci Comput Program 172:48–62
11. Benzmüller C (2022) Symbolic AI and Gödel's ontological argument. *Zygon(r)* 57:953–962
12. Benzmüller C, Fuenmayor D (2023) Mathematical proof assistants for teaching logic: the LogiKEy methodology. In *Book of Abstracts — V Congress Tools for Teaching Logic*, Madrid, Spain, https://doi.org/10.13140/RG.2.2.24708.74888

13. Benzmüller C, Fuenmayor D (2024) The LogiKEy methodology: applications in AI ethics & prospects for logic education. In Formal methods and science in philosophy IV, Book of Talk Abstracts
14. Benzmüller C, Woltzenlogel Paleo B (2016) The inconsistency in Gödel's ontological argument: a success story for AI in metaphysics. In Subbarao K, (Eds) *Proc. of the 25. international joint conference on artificial intelligence (IJCAI)*, pages 936–942. IJCAI/AAAI Press
15. Benzmüller C, Parent X, van der Torre LWN (2020) Designing normative theories for ethical and legal reasoning: LogiKEy framework, methodology, and tool support. *Artif. Intell.*, 287:103348
16. Benzmüller C, Wisniewski M, Steen A (2015) Computational metaphysics. https://doi.org/10.13140/RG.2.1.3535.2568; this lecture course proposal received the 2015 central teaching award of FU Berlin
17. Berdine J, Calcagno C, O'Hearn PW (2005) Smallfoot: modular automatic assertion checking with separation logic. In Frank S. de Boer et al., editors, *4th Int. symposium on formal methods for components and objects (FMCO)*, volume 4111 of *LNCS*, pages 115–137. Springer
18. Besold TR, et al (2021) Neural-symbolic learning and reasoning: a survey and interpretation. In Pascal H, Md. Kamruzzaman S (Eds) *Neuro-symbolic artificial intelligence: the state of the art*, volume 342 of *Frontiers in artificial intelligence and applications*, pages 1–51. IOS Press
19. Beyersdorff O, Biere A, Ganesh V, Nordström J, Oertel A (2022) Theory and practice of SAT and combinatorial solving (dagstuhl seminar 22411). Dagstuhl Rep 12(10):84–105
20. Blackburn P, Benthem JFAK van Frank W (2006) Handbook of modal logic, Elsevier
21. Bohrer R (2017) Differential dynamic logic. *Archive of Formal Proofs*, February . Formal proof development https://isa-afp.org/entries/Differential_Dynamic_Logic.html
22. Bonacina MP, Fontaine P, Nalon C, Schon C, Desharnais M (2024) The next generation of deduction systems: from composition to compositionality (Dagstuhl Seminar 23471). Dagstuhl Rep 13(11):130–150
23. Brakensiek J, Heule M, Mackey J, Narváez DE (2022) The resolution of Keller's conjecture. J Autom Reason 66(3):277–300
24. Brewka G, Delgrande J, Romero J, Schaub T (2023) A general framework for preferences in answer set programming. Artif Intell 325:104023
25. Brewka G, Strass H, Ellmauthaler S, Wallner JP, Woltran S (2013) Abstract dialectical frameworks revisited. In Francesca R (Eds) *Proc. of the 23rd international joint conference on artificial intelligence (IJCAI)*, pages 803–809. IJCAI/AAAI
26. Cimatti A, et al (2002) NuSMV Version 2: an opensource tool for symbolic model checking. In *Proc. of the international conference on computer-aided verification (CAV 2002)*, volume 2404 of *LNCS*, Copenhagen, Denmark, July . Springer
27. Collenette J, Atkinson K, Bench-Capon T (2023) Explainable AI tools for legal reasoning about cases: a study on the european court of human rights. Artif Intell 317:103861
28. Cropper A, Dumancic S, Evans R, Muggleton SH (2022) Inductive logic programming at 30. Mach Learn 111(1):147–172
29. Cruanes S (2014) Logtk: a logic toolkit for automated reasoning and its implementation. In Stephan S, Leonardo de M, Boris K (Eds) *Proc. of the 4th workshop on practical aspects of automated reasoning (PAAR)*, volume 31 of *EPiC series in computing*, pages 39–49. EasyChair
30. de Arnaldo JC, Justo JF, de Oliveira AM, da Filho JIS (2024) A comprehensive review on paraconsistent annotated evidential logic Algorithms, applications and perspectives. Eng Appl Artif Intell 127:107342
31. de Moura LM, Bjørner NS (2008) Z3: an efficient SMT solver. In Ramakrishnan CR, Jakob R (Eds), *14th international conference on tools and algorithms for the construction and analysis of systems (TACAS)*, volume 4963 of *LNCS*, pages 337–340. Springer
32. del Cerro LF, et al (2001) Lotrec: the generic tableau prover for modal and description logics. In Rajeev G, Alexander L, Tobias N, (Eds), *First international joint conference on automated reasoning (IJCAR)*, volume 2083 of *LNCS*, pages 453–458. Springer
33. Allen EE, Clarke EM (1982) Using branching time temporal logic to synthesize synchronization skeletons. Sci Comput Program 2(3):241–266
34. Emerson EA, Halpern JY (1982) Decision procedures and expressiveness in the temporal logic of branching time. In *Proceedings of the fourteenth annual ACM symposium on Theory of computing*, pages 169–180
35. Gabbay D et al (Eds) *Handbook of the history of logic (in 11 volumes)*. Elsevier B.V., 2004–
36. Etherington DW, Reiter R (1983) On inheritance hierarchies with exceptions. In *Aaai*, volume 83, pages 104–108
37. Fagin R, Halpern JY, Moses Y, Vardi MY (1995) *Reasoning About Knowledge*. MIT Press
38. From AH, Villadsen J (2022) Soundness and completeness of implicational logic. *Arch. Formal Proofs*
39. Gabbay D, Horty J, Parent X, van der Meyden R, van der Torre LWN (Eds) *Handbook of deontic logic and normative systems*, volume 1. College Publications, 2013
40. Gabbay D, Horty J, Parent X, van der Meyden R, van der Torre LWN (Eds). *Handbook of deontic logic and normative systems*, volume 2. College Publications, 2021
41. Gabbay D, Pnueli A, Shelah S, Stavi J(1980) On the temporal analysis of fairness. In *Proceedings of the 7th ACM SIGPLAN-SIGACT symposium on Principles of programming languages*, pages 163–173
42. Gabbay D, Giacomin M, Liao B, van der Torre LWN (2018) Present and future of formal argumentation (Dagstuhl Perspectives Workshop 15362). *Dagstuhl Manifestos*, 7(1):69–95
43. Gebser M, Kaufmann B, Schaub T (2012) Multi-threaded ASP solving with clasp. Theory Pract Log Program 12(4–5):525–545
44. Gibbons J, Wu N (2014) Folding domain-specific languages: deep and shallow embeddings (functional pearl). In Johan J, Manuel MTC, (Eds) *Proceedings of the 19th ACM SIGPLAN international conference on Functional programming, Gothenburg, Sweden, September 1-3, 2014*, pages 339–347. ACM
45. Glimm B, Horrocks I, Motik B, Stoilos G, Wang Z (2014) Hermit: an OWL 2 reasoner. J Autom Reason 53(3):245–269
46. Goranko V (2023) Logics for strategic reasoning of socially interacting rational agents: an overview and perspectives. Logics 1(1):4–35
47. Gulwani S, Hernández-Orallo J, Kitzelmann E, Muggleton SH, Schmid U, Zorn B (2015) Inductive programming meets the real world. Commun ACM 58(11):90–99
48. Guzman LPG (2022) Stalnaker's epistemic logic. *Archive of Formal Proofs*, September, Formal proof development https://isa-afp.org/entries/Stalnaker_Logic.html
49. Gödel K (1933) Eine interpretation des intuitionistischen aussagenkalküls. Ergebnisse eines mathematisches Kolloquiums, 4:39–40, Reproduced and translated with an introductory note by A. S. Troelstra in Gödel 1986:296–304
50. Haarslev V, Hidde K, Möller R, Wessel M (2012) The racerpro knowledge representation and reasoning system. Semantic Web 3(3):267–277
51. Hassona M, Schulz S (2016) Deduction as a service. In Pascal F, Stephan S, Josef U, (Eds) *Proceedings of the 5th workshop on practical aspects of automated (PAAR)*, volume 1635 of *CEUR workshop proceedings*, pages 32–40. CEUR-WS.org

52. Heule MJH, Kullmann O (2017) The science of brute force. Commun ACM 60(8):70–79
53. Holzmann GJ (2004) *The SPIN model checker: primer and reference manual*, volume 1003. Addison-Wesley Reading
54. Jacobs B, Smans J, Philippaerts P, Vogels F, Penninckx W, Piessens F (2011) VeriFast: a powerful, sound, predictable, fast verifier for C and Java. In *NASA formal methods symposium*, pages 41–55. Springer
55. Kifer M, Liu YA (Eds) (2018) *Declarative logic programming: theory, systems, and applications*, volume 20 of *ACM Books*. ACM / Morgan & Claypool
56. Lenat D, Marcus G (2023) Getting from generative AI to trustworthy AI: what LLMs might learn from Cyc. arXiv:2308.04445
57. Makinson D (2005) *Bridges from classical to nonmonotonic logic*, volume 5 of *Texts in computing*. College Publications
58. Marques-Silva J, Malik S (2018) Propositional SAT solving. In Edmund M C, Thomas A H, Helmut V, Roderick B (Eds) *Handbook of model checking*, pages 247–275. Springer
59. Marra G, Dumancic S, Manhaeve R, De Raedt L (2024) From statistical relational to neurosymbolic artificial intelligence: a survey. Artif Intell 328:104062
60. Mints G (2012) The Gödel-Tarski translations of intuitionistic propositional formulas. *Correct reasoning: essays on logic-based AI in honour of Vladimir Lifschitz*, pages 487–491
61. O'Hearn P (2019) Separation logic. Commun ACM 62(2):86–95
62. Hans JO (1991) Semantics based translation methods for modal logics. J Log Comput 1(5):691–746
63. Plaza J (2007) Logics of public communications. Synthese 158:165–179
64. Pnueli A (1977) The temporal logic of programs. In *Proceedings of the 18th IEEE symposium on foundations of computer science*, pages 46–67. IEEE
65. Priest G (2012) *An introduction to non-classical logic: from if to Is*. Cambridge introductions to philosophy. Cambridge University Press, 2 edition
66. Priest G, Tanaka K, Weber Z (2022) Paraconsistent logic. In Edward N Z, (Eds) *The stanford encyclopedia of philosophy*. Metaphysics Research Lab, Stanford University, Spring 2022 edition
67. De Raedt L, Schmid U, Langer J (2023) Approaches and applications of inductive programming (dagstuhl seminar 23442). Dagstuhl Rep 13(10):182–211
68. Reiter R (1980) A logic for default reasoning. Artif Intell 13(1–2):81–132
69. Reynolds JC (2002) Separation logic: a logic for shared mutable data structures. In *Proceedings 17th annual IEEE symposium on logic in computer science*, pages 55–74. IEEE
70. Riazanov A, Voronkov A (2002) The design and implementation of VAMPIRE. AI Commun 15(2–3):91–110
71. Robinson JA, Voronkov A (Eds) (2001) *Handbook of automated reasoning (in 2 volumes)*. Elsevier and MIT Press
72. Schmid U, Wrede B (2022) Explainable AI. Künstliche Intell 36(3):207–210
73. Schneider T, Simkus M (2020) Ontologies and data management: a brief survey. Künstliche Intell 34(3):329–353
74. Schulz S (2002) E - a brainiac theorem prover. AI Commun 15(2–3):111–126
75. Shukla A, Biere A, Pulina L, Seidl M (2019) A survey on applications of quantified boolean formulas. In *31st IEEE international conference on tools with artificial intelligence (ICTAI)*, pages 78–84. IEEE
76. Steen A (2022) An extensible logic embedding tool for lightweight non-classical reasoning (short paper). In Boris K, Claudia S, Alexander S, (Eds) *Proceedings of the workshop on practical aspects of automated reasoning (PAAR)*, volume 3201 of *CEUR workshop proceedings*. CEUR-WS.org
77. Steen A, Benzmüller C (2024) What are non-classical logics and why do we need them? An extended interview with Dov Gabbay and Leon van der Torre, Künstliche Intell
78. Steen A, Sutcliffe G, Benzmüller C (2024) Solving quantified modal logic problems by translation to classical logics. *J Logic Comput*, Submitted. Preprint available at arXiv:2212.09570 [cs. LO]
79. Steen A, Sutcliffe G, Scholl T, Benzmüller C (2023) Solving modal logic problems by translation to higher-order logic. In Andreas H, Jieting L, Pere P, (Eds) *5th international conference on logic and argumentation (CLAR)*, volume 14156 of *LNCS*, pages 25–43. Springer
80. Steigmiller A, Liebig T, Glimm B (2014) Konclude: system description. J Web Semant 27–28:78–85
81. Tishkovsky D, Schmidt RA, Khodadadi M (2012) The Tableau prover generator MetTeL2. In Luis Fariñas del C, Andreas H, Jérôme M (Eds) *Proc. of the 13th European conference on logics in artificial intelligence (JELIA)*, volume 7519 of *LNCS*, pages 492–495. Springer
82. Trinh Trieu W, Tony Y, Quoc L, He H, Thang L (2024) Solving olympiad geometry without human demonstrations. Nature 625:476–482
83. van Benthem J, van Eijck J, Kooi B (2006) Logics of communication and change. Inf Comput 204(11):1620–1662
84. van der Hoek W, Wooldridge MJ (2012) Logics for multiagent systems. AI Mag 33(3):92–105
85. Wansing H (2001) *Essays on Non-classical Logic*. Advances in logic. World Scientific
86. Wasilewska A (2018) *Logics for computer science: classical and non-classical*. Springer International Publishing
87. Dongran Y, Yang B, Liu D, Wang H, Pan S (2023) A survey on neural-symbolic learning systems. Neural Netw 166:105–126