An Abstraction Approach for Formal Argumentation

Iosif Apostolakis, Johannes P. Wallner

Graz University of Technology, Graz, Austria

Abstract

We present planned work on using abstraction to support the presentation of argumentative reasoning in computational argumentation. Building on our previous symbolic methods for abstracting potentially complex argumentation frameworks, we aim to extend the approach with a machine learning component. In this setting, the latter component determines the parts of the framework that are relevant based on user-input, while symbolic AI ensures that the reasoning presented remains sound at every level of abstraction.

Computational argumentation (CA) [1] is a branch of artificial intelligence (AI) concerned with the representation and structure of arguments, the relationships between them, and reasoning based on those arguments. Reasoning is a central task that refers to finding "acceptable" sets of arguments. A prominent way of defining acceptance is via so-called argumentation semantics [2]. For example an argument may be accepted if it has no counterarguments. If it does, then each counterargument must itself be countered. In this way, reasons against accepting an argument can themselves be challenged, providing defense to the initial argument.

Argumentation can be complex. While argumentative reasoning is in general complex for automated reasoning (e.g., all major argumentative reasoning tasks are NP-hard [1]), we here aim mostly at ways of presenting argumentative reasoning to users in a way that avoids unnecessary complexities.

In previous works [3], we looked at symbolic methods for utilizing abstraction for argumentative reasoning. There we developed ways to abstract complex argumentation scenarios, while preserving reasoning properties. Our plan is to combine machine learning (ML) approaches with these symbolic methods. In brief, symbolic reasoning provides justified reasoning, while ML methods can be employed to steer which part of the reasoning should be presented to the user in order to enhance understanding.

Consider Figure 1(a). Assume we have a database "DB" that contains some (formalized) knowledge. This can take many shapes, but say such a database is rule-based and may contain certain facts. Also assume that a user wants to see why a certain conclusion $\mathbf{C}$ can be drawn from the database. The idea is to present an incomplete argument structure for $\mathbf{C}$, based on the premises and rules that are most interesting to the user. In this way a user is not burdened with all of the framework's complexities.

Our idea is to utilize ML approaches, e.g. large language models (LLMs), to determine what parts of the derivation could be interesting to a user, and hide the rest. For instance, in Figure 1 we only show two particular premises X and Y leading to conclusion $\mathbf{C}$. The "full" argument might have many more premises and rules governing the derivation of the conclusion. Internally, the full derivation towards $\mathbf{C}$ is checked via symbolic reasoning, ensuring that presented argument aligns with the original case.

Now, in case a user seeks more details on how the argument concluding $\mathbf{C}$ is formulated, or why is this partial argument warranted, we proceed to unravel the framework further. In particular, we "zoom out" to show the main counter-arguments (with respect to user's interests), as shown in Figure 1(b). Arguments deriving $\mathbf{D}$ and $\mathbf{E}$ can be construed as counter-arguments to the argument for $\mathbf{C}$.

Finally, the user is skeptical about the reasons against argument claiming $\mathbf{E}$, and asks for more details. We proceed to Figure 1(c), and show the whole argument chain against $\mathbf{E}$: three further arguments, coming from an unattacked one (justified) attacking another one (defeated), defending a third one (justified), finally defeating argument claiming $\mathbf{E}$ (and defending our original argument).

Joint proceedings of the Workshop on Theory and Methods for Abstraction (THEMA 2026) and the Workshop on Modularity and Splitting Techniques for Knowledge Representation and Reasoning (MoST 2026) at the 9th Federated Logic Conference (FLoC 2026), July 13–29, Lisbon, Portugal

iosif.apostolakis@tugraz.at (I. Apostolakis); johannes.p.wallner@tugraz.at (J. P. Wallner)

0009-0003-2124-3773 (I. Apostolakis); 0000-0002-3051-1966 (J. P. Wallner)

© 2026 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

CEUR Workshop Proceedings ceur-ws.org ISSN 1613-0073

published 2026-11-15

CEUR-WS.org/Vol-4331/abstract1.pdf

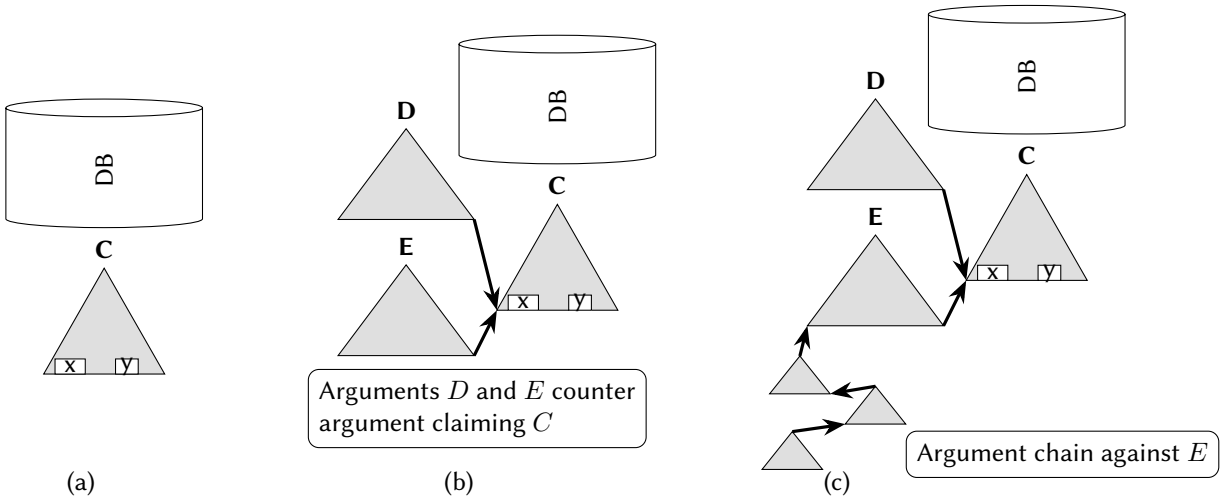


Figure 1: Different stages of presenting abstracted argumentative reasoning.

Behind the scenes, the system initially maintains the full, structured argumentation framework generated from DB. Given a user query about conclusion **C**, a symbolic reasoner constructs all arguments and counterarguments and checks their status under the desired semantics.

In our previous work, we developed ways for obtaining clustered versions of sets of arguments that preserve reasoning properties. We aim to employ this technique to obtain a higher level view of the initial framework. Building on this, we aim to employ ML techniques in order to (i) suggest what are interesting or relevant parts of arguments or what are interesting arguments in terms of presentation to a user and (ii) provide a natural-language-text presentation of an abstracted cluster of arguments.

For instance, for (i), an LLM (or other ML component) that possibly has background knowledge about the user can be prompted to decide which set of arguments or what part to present.

For (ii), counter-arguments to **E** in Figure 1(c) could also be presented differently: an ML (LLM) component can be given this set of arguments and asked to provide a textual summarization.

In summary, we aim to combine strengths of sound argumentative reasoning with ML components.

Acknowledgments

This research was supported by the Austrian Science Fund (FWF) grants P35632 and 10.55776/COE12.

Declaration on Generative AI

During the preparation of this work, the authors used ChatGPT in order to: Drafting content. After using this tool, the authors reviewed and edited the content as needed and take full responsibility for the publication's content. Figure 1 was created by prompting ChatGPT for a code draft in TikZ and subsequent editing by the authors. Generative AI was not used on other parts of this extended abstract.

References

[1] P. Baroni, D. Gabbay, M. Giacomin, L. van der Torre (Eds.), Handbook of Formal Argumentation, College Publications, 2018.

[2] P. Baroni, M. Caminada, M. Giacomin, An introduction to argumentation semantics, Knowl. Eng. Rev. 26 (2011) 365–410.

[3] I. Apostolakis, Z. G. Saribatur, J. P. Wallner, Abstraction in assumption-based argumentation, in: Proc. KR, 2024, pp. 49–59.