Abstraction In Non-Monotonic Reasoning

Doctoral Consortium

Iosif Apostolakis
Graz University of Technology
Graz, Austria
apostolakis@ist.tugraz.at

ABSTRACT

Abstraction emerges as a valuable method across diverse domains of Artificial Intelligence (AI), particularly in the field of knowledge representation and reasoning. Intuitively, abstraction maps a complicated structure to a simpler version of it. That reduces the computational complexity of the task being considered, as it provides us with the ability to focus on the parts of the problem that are relevant to the solution. In our view, such a tool can also have potential in the field of non-monotonic reasoning. Non-monotonicity is a crucial notion as it is very common when reasoning over defeasible knowledge. Adding new entries to our current knowledge, oftentimes results in restricting the conclusions that we can draw. For this form of reasoning we use certain formalisms, such as computational argumentation and Logic Programming (LP), that help us capture non-monotonicity. However, interpreting these formalisms faces hardships due to the large structures that might occur when representing the problem in question. Hence, coming up with ways to manage these structures easier is necessary. Recently, abstraction was shown to be a promising tool when dealing with Argumentation Frameworks (AFs) as well as with LP. AFs are frameworks with graph-like structure, whose nodes represent arguments with no internal structure, while edges stand for conflicts among the arguments. In our research we focus on continuing in this direction by employing structured frameworks such as Assumption-Based Argumentation Frameworks (ABAFs). Subsequently, we will extend our research to similar formalisms such as LP.

KEYWORDS

Computational Argumentation; Abstraction; Clustering; Structured Argumentation; Assumption-based Argumentation; Logic Programming

ACM Reference Format:
Iosif Apostolakis. 2024. Abstraction In Non-Monotonic Reasoning : Doctoral Consortium. In *Proc. of the 23rd International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2024), Auckland, New Zealand, May 6 – 10, 2024*, IFAAMAS, 3 pages.

1 INTRODUCTION

Abstraction serves as a useful method in various fields of Artificial Intelligence (AI) [11]. Particularly in the context of model checking [10], abstraction was successfully employed to tackle the state explosion problem. Model checking is the field of research that aims to verify whether a given system specification complies with certain properties. In this field one often encounters challenges due to the large size of the state space of these systems. Abstraction is a valuable tool as it simplifies these cumbersome representations while preserving the validity of the properties we aim to verify.

In its essence, abstraction simplifies the representation of such large structures by filtering parts of the structure that may not be relevant to the property we want to verify, or by replacing parts that could be represented in a more simple way. Abstraction can be viewed as a change of the representation of a given problem, in order to attain a "simpler" representation that bears all the vital information of the initial [20]. By applying an "appropriate" abstraction, we can avoid unnecessary information and thus make the whole interpretation process faster and clearer. Finding such an "appropriate" abstraction is in general a task related to the problem itself and the properties we want to preserve.

The concept of abstraction is visually presented in Figure 1. The process involves applying abstraction to the initial (or concrete) structure with the objective of obtaining a simpler and more manageable abstract structure. Then we reason over the simplified structure to address our simplified problem. The final step involves adapting the abstract solution back to the concrete scenario, which is a process called refinement.

Abstracting, albeit very useful, comes at a price: reasoning over an abstracted structure might yield false results due to ignoring possibly essential information that was abstracted away. In other words, a property might hold in the abstracted structure, but might fail to hold in the original one, or vice versa. Such spuriousness is unavoidable in most cases, as we can not avoid losing information in our attempt to simplify. However, we can separate our approaches into two directions: over-approximating and under-approximating. In the former we only consider abstractions that preserve the models of the original system, at the risk of generating a surplus of abstract models that do not correspond to any of the original ones. On the contrary, under-approximating aims to ensure that an abstract model corresponds to an original one, without guaranteeing that all original models are preserved.

2 STRUCTURED ARGUMENTATION

Formal argumentation [2, 3, 6] is a way to draw conclusions from incomplete knowledge. If one would take a closer look to some everyday discussions, they might realise that usually we tend not to strictly prove, but rather to provide persuasive arguments for our points. It is common that people use this sort of reasoning on a daily basis. In artificial intelligence, the most prominent approach to formalize argumentation was introduced by Dung [13]. This

This work is licensed under a Creative Commons Attribution International 4.0 License.

Proc. of the 23rd International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2024), N. Alechina, V. Dignum, M. Dastani, J.S. Sichman (eds.), May 6 – 10, 2024, Auckland, New Zealand. © 2024 International Foundation for Autonomous Agents and Multiagent Systems (www.ifaamas.org).

formalization is based on abstract arguments and an attack relation that arises in between arguments.

Formally, an abstract argumentation framework (AF) [13] is a pair $F = (A, R)$, where A is a set of arguments, and $R \subseteq A \times A$ describes an attacking relation among these arguments. In contrast to structured argumentation, in abstract argumentation, arguments have no internal structure. That is, each argument stands for an abstract statement, of which we have no information of the way it is derived. However, through the attacking relation, we know how it interacts with the rest of the arguments. As mentioned, this is not the case for structured argumentation, where for each argument, we have information about how this argument is derived.

It is often the case that we are dealing with an AF with an argument set of a very sizeable size. This "argument explosion" poses computational barriers, since dealing with such large structures can be computationally costly, but also poses interpretational barriers, i.e. if a set of accepted arguments is too large, it can be hard to find sufficient explanations for its acceptance. A form of abstraction (clustering) has been used to address this problem [22]. Abstraction on AFs is carried out in the form of clustering over the set of arguments. Let $m : A \rightarrow \hat{A}$ be a surjective mapping. Then m is called a clustering, and $\hat{A}$ contains clustered arguments. This clustering induces a new framework $\hat{F} = (\hat{A}, \hat{R})$ that contains as arguments the clustered arguments of $\hat{A}$. A cluster attacks another cluster if and only if there is an element in the former cluster that attacks an element of the latter in the initial framework.

We aim to extend abstraction to structured argumentation. As mentioned previously, AFs are graphs whose nodes have no internal structure. So far, we have formalized abstraction on Assumption-Based Argumentation Frameworks (ABAFs) [7, 12]. These frameworks are defined as a tuple $D = (\mathcal{L}, \mathcal{R}, \mathcal{A}, \bar{\ })$, where $\mathcal{L}$ is a set of atoms of which we distinguish the assumptions $\mathcal{A} \subseteq \mathcal{L}$. Each argument of D has internal structure that is inherited by a rule in the set $\mathcal{R}$. Finally, $\bar{\ }$ assigns a contrary to each assumption. In case the contrary of an assumption a is derived from a set of assumptions A, we say that set A attacks a. In our view, lifting the abstraction applied on AFs to ABAs, is non-trivial and also can bear benefits to explaining accepted sets of assumptions.

Our approach of abstracting on ABAFs focuses on clustering elements of the assumption set $\mathcal{A}$, by introducing a surjective mapping $m : \mathcal{A} \rightarrow \hat{\mathcal{A}}$ (clustering). Then the other parts of the ABAF are altered, forming a clustered ABAF (cABAF) denoted by $\hat{D} = (\hat{\mathcal{L}}, \hat{\mathcal{A}}, \hat{\mathcal{R}}, \hat{\bar{\ }})$, where: *i)* the non-assumption elements remain unchanged, and thus $\hat{\mathcal{L}} \setminus \hat{\mathcal{A}} = \mathcal{L} \setminus \mathcal{A}$, *ii)* rules in $\hat{\mathcal{R}}$ occur after replacing the assumptions of a rule in $\mathcal{R}$ with their respective cluster, *iii)* the contrary of a cluster is the set that contains the contraries of each assumption that lies in the cluster. We remark that a clustered ABAF is not a "classical" ABAF. The contrary function of a classical ABAF maps elements of its assumption set to a particular atom in its formal language. Instead, the contrary function of a clustered ABAF maps clusters to subsets of the clustered formal language.

3 INTERPRETING AND OBTAINING CABAFS

Interpreting abstract frameworks is a key component of our work. In AFs, ABAFs and LPs interpretation is achieved by different semantics [7, 13, 15]. In the case of ABAFs, a semantics of a framework

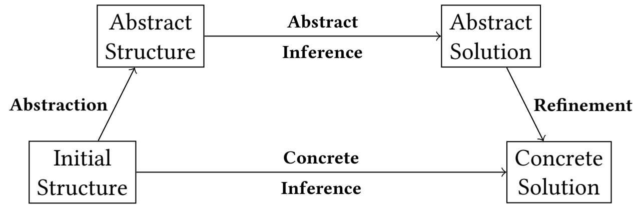

Figure 1: Process of abstraction

D is a set that contains sets of assumptions $\sigma(D) \subseteq 2^{\mathcal{A}}$. A set in a semantics represents a set of assumptions that can be accepted together under some specific condition. For example, a well-known condition for accepting a set is conflict-freeness, that is to contain assumptions that are not in conflict with each other.

Defining semantics to interpret abstract frameworks is strictly related to the way we interpret the respective notion in the original framework. We already mentioned two approaches one can follow, i.e. over-approximation or under-approximation. In our work so far we have introduced abstract semantics to interpret cABAFs by over-approximating the classical ones. We also have theoretical results regarding the optimality of the abstract semantics we define, i.e. they allow the least spurious sets that is possible, as well as complexity results that prove that in some cases, over-approximating a specific semantics is a challenging task.

Additionally, we came up with two ways of obtaining faithful clusterings, i.e. clusterings that have no spurious sets under a specific semantics. One way is to start with a "coarse" abstraction, e.g., with clustering all assumptions into one big cluster—abstracting as much as possible—and refine, upon user requests using answer set programming (ASP) [15, 19]. The other approach computes a clustering faithful under certain semantics by starting with singletons and iteratively clusters following the grounded semantics [12].

4 FUTURE DIRECTIONS

We believe that our approach can be beneficial for supporting explainability, by providing foundational work towards abstracting certain parts of argumentative reasoning in a faithful manner. Interactive tools that give users the ability to "zoom in" or "zoom out" can be useful to improve understanding.

We think more research is needed to make non-monotonic reasoning formalisms [1] more accessible and to help users to digest their results. Among interesting avenues for future works are, e.g., extending our approach to other formal approaches to structured argumentation, such as ASPIC$^+$ [18], defeasible logic programming (DeLP) [14], deductive argumentation [5], Carneades [16], or Gorgias [17]. Currently, as a next step we aim to utilize the connection between ABAFs and Logic Programming (LP) [8, 9, 23] to extend current abstraction approaches [21]. Moreover, extending abstraction with a recently proposed notion of forgetting parts of an ABA knowledge base [4] is intriguing. In the future, we also aim to study how under-approximating could be of use.

ACKNOWLEDGMENTS

This research was funded in whole or in part by the Austrian Science Fund (FWF) P35632.

REFERENCES

[1] Grigoris Antoniou and Mary-Anne Williams. 1997. *Nonmonotonic reasoning*. MIT Press.

[2] Katie Atkinson, Pietro Baroni, Massimiliano Giacomin, Anthony Hunter, Henry Prakken, Chris Reed, Guillermo Ricardo Simari, Matthias Thimm, and Serena Villata. 2017. Towards Artificial Argumentation. *AI Mag.* 38, 3 (2017), 25–36.

[3] Pietro Baroni, Dov Gabbay, Massimiliano Giacomin, and Leendert van der Torre (Eds.). 2018. *Handbook of Formal Argumentation*. College Publications.

[4] Matti Berthold, Anna Rapberger, and Markus Ulbricht. 2023. Forgetting Aspects in Assumption-Based Argumentation. In *Proc. KR*, Pierre Marquis, Tran Cao Son, and Gabriele Kern-Isberner (Eds.). ijcai.org, 86–96.

[5] Philippe Besnard and Anthony Hunter. 2018. A Review of Argumentation Based on Deductive Arguments. In *Handbook of Formal Argumentation*, Pietro Baroni, Dov Gabbay, Massimiliano Giacomin, and Leendert van der Torre (Eds.). College Publications, Chapter 9, 437–484.

[6] Alexander Bochman. 2017. Argumentation, Nonmonotonic Reasoning and Logic. *IfCoLog* 4, 8 (2017).

[7] Andrei Bondarenko, Phan Minh Dung, Robert A. Kowalski, and Francesca Toni. 1997. An Abstract, Argumentation-Theoretic Approach to Default Reasoning. *Artif. Intell.* 93 (1997), 63–101.

[8] Martin Caminada, Samy Sá, João F. L. Alcântara, and Wolfgang Dvořák. 2015. On the equivalence between logic programming semantics and argumentation semantics. *Int. J. Approx. Reason.* 58 (2015), 87–111.

[9] Martin Caminada and Claudia Schulz. 2017. On the Equivalence between Assumption-Based Argumentation and Logic Programming. *J. Artif. Intell. Res.* 60 (2017), 779–825.

[10] Edmund M. Clarke, Thomas A. Henzinger, Helmut Veith, and Roderick Bloem (Eds.). 2018. *Handbook of Model Checking*. Springer.

[11] Mika Cohen, Mads Dam, Alessio Lomuscio, and Francesco Russo. 2009. Abstraction in model checking multi-agent systems. In *Proc. AAMAS*, Carles Sierra, Cristiano Castelfranchi, Keith S. Decker, and Jaime Simão Sichman (Eds.). IFAAMAS, 945–952.

[12] Kristijonas Čyras, Xiuyi Fan, Claudia Schulz, and Francesca Toni. 2018. Assumption-Based Argumentation: Disputes, Explanations, Preferences. In *Handbook of Formal Argumentation*, Pietro Baroni, Dov Gabbay, Massimiliano Giacomin, and Leendert van der Torre (Eds.). College Publications, Chapter 7, 365–408.

[13] Phan Minh Dung. 1995. On the Acceptability of Arguments and its Fundamental Role in Nonmonotonic Reasoning, Logic Programming and n-Person Games. *Artif. Intell.* 77, 2 (1995), 321–358.

[14] Alejandro Javier García and Guillermo Ricardo Simari. 2018. Argumentation Based on Logic Programming. In *Handbook of Formal Argumentation*, Pietro Baroni, Dov Gabbay, Massimiliano Giacomin, and Leendert van der Torre (Eds.). College Publications, Chapter 8, 409–435.

[15] Michael Gelfond and Vladimir Lifschitz. 1988. The Stable Model Semantics for Logic Programming. In *Proc. ICLP*, Robert A. Kowalski and Kenneth A. Bowen (Eds.). MIT Press, 1070–1080.

[16] Thomas F. Gordon, Henry Prakken, and Douglas Walton. 2007. The Carneades model of argument and burden of proof. *Artif. Intell.* 171, 10-15 (2007), 875–896.

[17] Antonis C. Kakas, Pavlos Moraitis, and Nikolaos I. Spanoudakis. 2019. *GORGIAS*: Applying argumentation. *Argument Comput.* 10, 1 (2019), 55–81.

[18] Sanjay Modgil and Henry Prakken. 2018. Abstract Rule-Based Argumentation. In *Handbook of Formal Argumentation*, Pietro Baroni, Dov Gabbay, Massimiliano Giacomin, and Leendert van der Torre (Eds.). College Publications, Chapter 6, 287–364.

[19] Ilkka Niemelä. 1999. Logic Programs with Stable Model Semantics as a Constraint Programming Paradigm. *Ann. Math. Artif. Intell.* 25, 3-4 (1999), 241–273.

[20] Lorenza Saitta. 2013. *Abstraction in artificial intelligence and complex systems*. Springer.

[21] Zeynep G. Saribatur, Thomas Eiter, and Peter Schüller. 2021. Abstraction for non-ground answer set programs. *Artif. Intell.* 300 (2021), 103563.

[22] Zeynep G. Saribatur and Johannes P. Wallner. 2021. Existential Abstraction on Argumentation Frameworks via Clustering. In *Proc. KR*, Meghyn Bienvenu, Gerhard Lakemeyer, and Esra Erdem (Eds.). ijcai.org, 549–559.

[23] Claudia Schulz and Francesca Toni. 2015. Logic Programming in Assumption-Based Argumentation Revisited - Semantics and Graphical Representation. In *Proc. AAAI*, Blai Bonet and Sven Koenig (Eds.). AAAI Press, 1569–1575.