Argument-Driven Planning & Autonomous Explanation Generation

Leonard Eberding[1]
Jeff Thompson[2]
Kristinn R. Thórisson[1,2]

Proceedings of Artificial General intelligence, 2024, 73-83

[1] Center for Analysis & Design of Intelligent Agents
Reykjavik University

[1,2] Icelandic Institute for Intelligent Machines
Reykjavik, Iceland

OCTOBER 23 2024

Argument-Driven Planning & Autonomous Explanation Generation

Leonard M Eberding,[1] Jeff Thompson[2] & Kristinn R. Thórisson[1,2]

[1] Center for Analysis & Design of Intelligent Agents
Reykjavik University, Iceland {leonard20,thorisson}@ru.is
[2] Icelandic Institute for Intelligent Machines (IIIM)
Reykjavík, Iceland jeff@iiim.is

Abstract. Research on general machine intelligence is concerned with building machines that are capable of performing a multitude of highly complex tasks in environments as complex as the real world. A system placed in such a world of indefinite possibilities and never-ending novelty must be able to adjust its plans dynamically to adapt to changes in the environment. These adjustments, however, should be based on an informed explanation that describes the hows and whys of interventions necessary to reach a goal. This means that explanations are at the core of planning in a self-explaining way. Using Assumption-Based Argumentation we present a way how an AGI-aspiring system could generate meaningful explanations. These explanations consist of argumentation graphs that represent proponents (i.e., solutions to the task) and opponents (contradictions to these solutions). They thus provide information on why which intervention is necessary, thus making an informed commitment to a particular action possible. Additionally, we show how such argumentation graphs could be used dynamically to adjust plans when contradicting evidence is observed from the environment.

Keywords: Argumentation · Artificial Intelligence · General Machine Intelligence · Causal Reasoning · Self-Explanation

1 Introduction

The process of explanation involves more than a transmission of information between two or more agents. An explanation can help a system, in a self-explaining manner, to identify reasons for and against achieving a goal at a future time, e.g. when deciding what actions to take or what plans to commit to [12]. For this, however, a special type of explanation is necessary. This type of explanation differs from the idea of explanations (or interpretations, two concepts often used interchangeably) in Deep Learning (DL) and explainable artificial intelligence (XAI) research. The approaches surrounding DL and XAI mainly consist of explaining the system itself either by explaining the processing of data or by explaining the representation of data inside the system [7]. What we envision

when we say explanation, however, is an explanation of the environment that the agent is in in such a way that the system can derive plans from the explanation[3].

Thórisson (2021) describes the 'Explanation Hypotheses' in autonomous general learning, which states that "explanation generation is a fundamental and necessary process for general self-supervised learning" [11]. He argues that an agent capable of self-improvement needs to keep track of and explain the "hows and whys" of failures. Thórisson (2023) extends this work, presenting a detailed description of "explicit goal-driven autonomous explanation generation" [12]. We build on this work and present a way to autonomously generate explanations usable by a system for (self-)evaluation of its knowledge's logical consistency[4] and plan generation. These explanations are not exclusively for explanations of the system's failures but rather for explaining the cause-effect patterns of the environment such that hypothetical failures of achieving a future goal can be predicted and countermeasures taken in advance. These explanations, therefore, need to be generated on the fly when a task is given to the agent.

Failures describe states from which a goal, given time and energy constraints, is impossible to reach. These conditions that forbid the reaching of a goal state are arguments against reaching the goal. Conditions that need to be met to reach a goal are arguments for a solution to the task. We use Assumption-Based Argumentation (ABA) with these arguments for and against solutions to a task to generate logically consistent plans and explanations about the "hows and whys" of decisions that the system commits to.

The paper is structured as follows: In section 2, we introduce core concepts and present related work and how it differs from our definitions (where applicable). In section 3, we present how a system can use ABA to generate explanations during planning. Section 4 highlights how these argumentation graphs generated by ABA can be dynamically adapted to changes in the environment. Section 5 gives a short insight into the implementation efforts before we conclude our work in section 6.

2 Core Concepts & Related Work

The real world is too complex to define a single function to be optimized (e.g., by a reinforcement learner). A system that is to perform tasks in such an environment must be able to reason about different state transitions that could happen by either the environment's own dynamics or the system's intervention. At any time, there can exist an infinite set of possible interventions by the agent on the environment. Backward chaining restricts the search space by chaining from the goal to the current state, thus reducing the set of possible interventions (possible in the sense of "leading to the goal"). However, multiple paths can exist for a single goal at any time. The solution space is restricted by the cause-effect

[3] For a more detailed analysis of exactly what we mean by *explanations*, we refer the reader to [11,12].

[4] Something that is necessary when a system learns from experience and can therefore never know whether existing knowledge is correct - in short, when it is non-axiomatic.

structures that define any of those paths to the goal. Some of these structures can rely on external conditions (i.e., contexts) that define the constraints under which the causal model holds. This means that certain situations need to be fulfilled for the system to reach the goal. Other situations could be forbidden along any particular path to the goal (e.g., dead-ends in the path towards the goal), describing attacks (i.e., opponents) on the original plan (i.e., proponent).

In order to commit to a decision in the initial state at the initial time, the system needs to autonomously identify blocked paths (or such that will be at a future time) by contexts that make reaching the goal state impossible. During backward chaining, the system can analyze all known attacks on causal relations used in any particular chain and thus create arguments for and against the chain. By creating a dynamic graph of the cause-effect structures, argumentation theory can be applied to identify extensions (i.e., valid solutions) that are complete (i.e., there exist no attacks on the plan that are not defended by other arguments), thus creating consistent explanations about the causal structures and interventions that need to be applied to reach the goal state. Additionally, it is possible to identify all possible attacks on the extension. Since the environment changes over time, the system can, therefore, identify attacks and counterattacks and thus ensure that all counterattacks are instantiated at the right time. In short, an AI system that uses argumentation-based reasoning can 1) explain why it chooses any particular solution by providing arguments for and against the solution, including why arguments against it do not hold, and 2) identify which situations need to be enacted/ instantiated to ensure that any situations that would prohibit the system from reaching its goal state are precluded.

2.1 Argumentation Theory

The value of including argumentation theoretic approaches in AI has been described in detail (see, for example, [14]) and commonly includes four different aspects of logical reasoning and argumentation. 1) Identification is concerned with identifying the premises and conclusions of an argument. 2) Analysis is for finding implicit premises and conclusions. Both of these will not be a concern of this work since the former is the fundamental assumption of any reasoning-based AI architecture, and the latter is mainly concerned with human language and the soundness of arguments rather than logical rules. 3) Invention is the construction of new arguments that can be used to prove a specific conclusion – again, an underlying premise of reasoning-based AI systems. However, the fourth aspect of argumentation, 4) evaluation, is what we want to focus on in this work. It is the task of evaluating the weakness (or strength) of an argument by applying general criteria to it. For this, we will use Assumption-Based Argumentation (ABA) in particular.

2.2 Assumption-Based Argumentation Theory

Assumption-Based Argumentation (ABA) is an instance of abstract argumentation (AA) that handles the generation of flat, non-circular argumentation graphs

(rather than trees in other argumentation theories) [2,3,4,13]. ABA is a general-purpose argumentation framework that supports a multitude of other applications and frameworks, including (but not limited to) reasoning systems, game theory, and decision theory. In ABA, arguments are made up of deductive inferences supported by assumptions. Arguments can be attacked by other arguments when the latter (i.e., opponent) deduces the contrary of an assumption supporting the former (i.e., proponent). In ABA, a derivation starts from a claim and proceeds through backward chaining to find one or more sets of assumptions that can support the claim (i.e., a top-level maintenance goal). If the top claim is itself an assumption, then the argumentation proceeds by considering an argument that can attack the top claim and finding counterattacks.

2.3 Definitions

Goal: A goal is a state defined at a time that is to be reached by an agent performing a task [1,12]. A goal always exists in the future and can span any amount of state-space dimensions. Other than, for example, in reinforcement learning (c.f. [6]), goals do not come with rewards or any sort of additional implications about their importance. The system must reason about possible paths to (multiple) goal(s) and analyze their importance in accordance with some higher-level goal or mutually exclusive goals independently and autonomously without additional information from the developers.

Solution: Belenchia et al. (2021) [1] describe the solution space as the sum of all states from which the goal is reachable. In a similar manner, we define a solution to a task as a path from the initial state to the goal through time. It consists of sub-goals that need to be enacted and other states that need to be avoided in order to reach the goal. A solution describes the full transformation of all relevant (to the goal) variables from the initial state to the goal state. We want to differentiate here between, for example, reinforcement learners (or other function approximators) and want to emphasize that a solution is a *path*, not an estimate of the successfulness of the next interaction with the environment. This is important since only such an explicit representation of a solution allows for further analysis concerning attacks and counterattacks.

Argument: An argument is a set of assumptions and rules that derive a conclusion. In our particular case, that means a set of assumptions and their deductions that define a causal chain connecting a state from earlier in time to a state later in time. An argument can consist of multiple implications spanning time and state-space dimensions. [14]

Proponent: We take the proponent concept from ABA [4,13]. The proponent graph describes a set of arguments that reaches from the current state to the goal state, thus describing the causal structures of performing a task. In other words, the proponent is the chain of models that is generated from chaining backward from a goal that is to be achieved. For a proponent to be valid, it must defend against all attacks from opponents (see following definitions) by counteracting all opponents' assumptions.

Opponent: The opponent, on the other hand, is the argument that describes an attack on the proponent [4,13]. While proponents can consist of multiple arguments, an opponent always consists of a single argument that describes an attack. Another major difference between pro- and opponent is that while the proponent needs to defend against every attack in order to hold, the opponent only needs to attack a single assumption of the proponent. The proponents and opponents all draw from the same set of knowledge of the system. There is no difference in the models, assumptions, rules, and observations between them.

Attack: An attack is an argument that deduces the contrary of other assumption(s) at a particular time. It thus invalidates the original argument if 1) the original argument necessitates the taking of the contradicted assumption, 2) the contradiction overlaps the assumed timing of the original assumption, and 3) the attacking argument is not itself attacked by a different argument.

Counterattack: we define a counterattack as an attack on an attacking argument. Assuming that the first argument represents a proponent for a solution to a task, an attacking argument would represent an opponent to the solution, whereas the counterattack supports the proponent's argument. If all attacks on an argument are counterattacked by other conclusions, it is a defended argument.

Extension: An extension is a set of arguments that can survive together (i.e., show no contradictions that are not in turn attacked by counterattacks) and are collectively acceptable [14]. We want to emphasize here that extension in our usage differs from extensions and intentions of other reasoning systems like the Non-Axiomatic Reasoning System (NARS). An extension is not an implied logical extension of rules to hierarchically organized structures. It is simply the extent of a set of arguments that leads to a valid and logically consistent solution that is defended on all attacks.

Explanation: We define an explanation as a valid extension of the ABA graph that is formed from the experience of a learning agent performing a task. Therefore, an explanation depends on 1) the agent's experience (including false or incomplete models of the world), 2) the goal of the task, and 3) the available (i.e., computable) attacks and counterattacks to a particular solution to the task. Therefore, this does not correspond to the definition of explanation in XAI research, where the primary concern is the explanation to others (i.e., the developer) that describes which data led to the system performing in a certain way or how data is stored in the system [7]. Our definition of explanation is independent of the explainee and is only concerned with creating consistent, logic-based arguments for or against a solution to a task.

3 Explanation generation through Assumption-Based Argumentation

In this section, we will give an overview of how explanations can be generated autonomously when needed by the agent. We will discuss some examples of how

argumentation theory supports explanation generation and will shortly touch on the topic of counterfactual reasoning and explanation.

3.1 From Backward Chaining to Explanations

Reasoning systems deployed in a highly complex world (like the real world) usually use backward chaining from a goal to the current state in order to reduce the size of the search space.[5] During backward chaining, we can apply assumption-based argumentation (ABA) to identify proponents and opponents, as well as attacks and counterattacks on solutions to the task.

We envision (and have implemented) the reasoning process of backward chaining through time as chains of causal models from effect states to their causes using a non-axiomatic knowledge representation. We adopt the motto, "All good explanations are causal explanations." In other words, knowledge is about the causal processes which transform a system from one state to another. A *cause* is a tuple of the environment's (sub-)state at time t and an event that causes the change (e.g., the action taken by the agent), called the assumption $\langle S_t, \mathbb{A} \rangle$. The effect is the environment's state after the intervention on $S_{t'}$ with $t' > t$. A model, therefore, represents a logical rule that connects the cause tuple to the effect state[6]. However, other models might provide contradicting implications that a particular model does not hold given some (other) (sub-)state $\bar{S}_t$. Applying this to assumption-based argumentation, the rules are the causal state transitions, and the assumptions are the hypothesized environment events or agent actions that cause those transitions.

During backward chaining, starting at the goal, the system chains effects (initially the goal, later sub-goals along the path) to causes and analyzes whether there exist models that would contradict the needed cause-effect structures necessary to perform the task. The original chain from the goal represents the proponent's argument for performing the task. Any contradictions represent attacks on the proponents, thus creating opponent chains. All opponents that prohibit the proponent are chained backward as well by creating anti-goals of states (i.e., cause states in the opponent chain) not to be observed. When a possibility to deflect the opponent's attack is found, the installation of this counterattack is added to the proponent argument. When all attacks are defended, the system has created a valid extension to the task and thus created an explanation for all measures that need to be taken in order to reach the goal. We want to direct the reader's attention to the fact that, during backward chaining, no information of the initial state is (at first) processed. This means that the explanation includes hypotheticals that could hinder the goal from being reached. Any hypotheticals that cannot be instantiated in the current environment (or context) are automatically dismissed since contradicting information (i.e., counterattacks) will be

[5] A purely forward-chaining system would immediately run into problems of computational explosion since the possible number of interventions with the real world is infinite (or at least extremely large).

[6] For a more detailed analysis of how noise and uncertainty can be handled in such a reasoning system, we refer the reader to [5]

found during further backward chaining. However, the explanation nevertheless includes these hypotheticals, thus making it easier for the system to adjust to them if they happen to be instantiated due to unforeseen dynamics in the environment.

4 Dynamic Argumentation Graphs

Given the described concepts of Assumption-Based Argumentation (ABA), backward chaining, and causal reasoning, it is possible to dynamically generate ABA graphs based on the system's assumption, incremental observations of the environment, and its ever-changing, non-axiomatic knowledge base. Given a goal G, the system can create a plan through backward chaining by connecting learned causal relational models (i.e., assumptions and conclusion), thus reasoning over state transitions that lead to the goal. Using argumentation, they can be evaluated for their consistency by analyzing whether the set of arguments making up the proponent are fully defended from opponent attacks.

Such a plan, however, might become unusable when new observations are added, new models learned, or old models defeated through new evidence. New observations, combined with credible assumptions and/or hypotheses (again, these are non-axiomatic and can change over time), can lead to conclusions that attack the proponent in an unforeseen way. Therefore, the system must be able to dynamically adjust its plan (i.e., argumentation graph) to include new evidence and newly learned or adjusted models. Figure 1 shows such a dynamic re-evaluation of the completeness of the argumentation graph when new evidence is available. First, the system generates a plan (e.g., going along a path to get water) to counteract an attack on its maintenance goal. After committing to the first action (e.g., going along the path), new evidence is made available in the form of an observation of something on the ground that could be a snake. Therefore, the agent dynamically re-evaluates the possible attacks on its proponent, finding new opposing conclusions that attack the action previously committed to or other assumptions taken when the original plan was created. After this argumentation graph adjustment, the agent can commit to a new action (e.g., throwing the stone to check whether it really is a snake) that keeps the proponent fully defended.

This way, the new evidence can be integrated into the plan by only analyzing its impact on the proponent. Given that a new attack can be found with this new evidence, the system only needs to focus on finding countermeasures to the new attack rather than having to generate a completely new plan. That means that, in theory, large and complex plans can be calculated at times of high resource availability and only need to be adjusted in parts when new evidence is made available, thus reducing the load on the processing unit when time and/or energy is sparse and quick plan adjustment is necessary.

At any point during planning and task performance, the system is able to answer questions like "Why are you looking at the dark shape?" "Because it may be the snake I saw earlier, which would prevent me from going to the water

hole to get water to prevent being thirsty, which would result in an unstable state." "Why did you throw the stone?" "To test the case where it doesn't move, meaning that it's not a snake, and I can continue the plan to walk to the water hole."

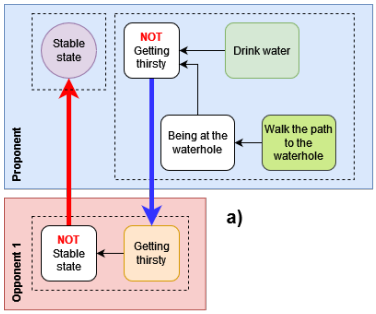
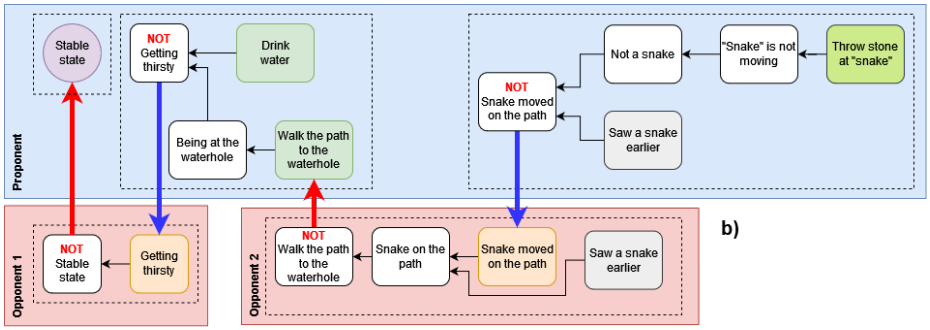


Fig. 1. Violet: Maintenance goal. Red rectangle: Opponent attacking the goal. Blue rectangle: Proponent defending the goal. Green and orange boxes: Assumptions. White boxes: Conclusions. Red arrows: Attacks. Blue Arrows: Counterattacks. Olive green rectangle: Committed decision. **a)** Commitment to the first action that leads to the goal. **b)** After the first commitment, the agent observes something on the path that could be a snake. The agent has seen a snake nearby earlier. A new decision is made to throw a stone to check whether the snake has moved onto the path.



4.1 Counterfactual reasoning

We want to shortly highlight that counterfactual reasoning and counterfactual explanation generation could be easily implemented in a system leveraging assumption-based argumentation theory for explanation generation. All that is necessary would be the introduction of attacks and/or counterattacks through external injection in the system, and it would immediately generate a new explanation given the newly set variables and their implications on the task. Counterfactual reasoning could be processed the same way as contradicting observations are handled by finding the attacks and counterattacks this counterfactual information introduces to the system's explanation.

5 Argumentation in AERA

The Autocatalytic Endogenous Reflective Architecture (AERA) [9,8] is a real-time architecture for learning, knowledge representation, and reasoning in a cog-

nitive agent (e.g., a robot). It uses ampliative reasoning, including deduction, abduction, causal discovery, and analogy-making in its reasoning process. AERA defines the general view of knowledge as models of causal processes and defines a syntax to represent this. Similar to ABA, it uses backward chaining to reason from a future goal state to infer one or more actions to achieve the goal, but AERA currently doesn't handle multiple conflicting goals or competing ways to achieve them. This is where we are integrating ABA. We have adapted the abagraph[7] software to use AERA syntax, which allows AERA's sister application, the AERA Visualizer[8], to display argument graphs.

Similar to Figure 1, instead of pursuing a single goal, AERA can use ABA with a high-level maintenance goal with an explicit explanation for the importance of each subgoal. AERA already tracks confidence levels of its knowledge to choose action sequences more likely to achieve a subgoal. Using ABA, AERA can be extended to not only use confidence as a proxy for which actions to commit to but rather use an informed (i.e., reasoned about) explanation of why and how interventions influence future goals. This includes 1) the management of mutually exclusive goals by identifying their attacks against each other, 2) the preparation for future interventions to preclude possible failure states, 3) identifying attacks within its own knowledge base (i.e., logical inconsistencies in the non-axiomatic knowledge base of AERA), and 4) keeping track of future influences that actions regarding one goal could have on other goals. Finally, using ABA, AERA can be extended to reason about counterfactual, that is, hypothetical scenarios that could be of concern when performing tasks. The current state of our implementation efforts will be presented at the "International Conference on Computational Models of Argument" (COMMA) later this year [10].

6 Conclusion

We have presented a strategy for AI systems to generate powerful explanations that can be used for self-explanation by the system itself based on the causality-based experiences of the agent. Using ABA, possible paths to the goal can be evaluated for their consistency (i.e., the consistency of the agent's knowledge) and possible disturbances can be predicted and countermeasures identified. With this work, we present a possible way how explanations as described in Thórisson (2021, 2023) [11,12] can be created and used for autonomous planning and self-improvement. We believe that the dynamic generation and adaptation of argumentation graphs provide a well-described way to improve AI reasoning systems. By bridging argumentation theory and non-axiomatic reasoning, we believe that such a dynamic argumentation graph generation is possible and present one possible architecture to which this work can be applied. The presented work focuses on the backward-chaining process of reasoning systems. Thus, systems

[7] Described in [13], available on GitHub at https://github.com/robertcraven/abagraph — *accessed 25th of April 2024*

[8] Code accessible on GitHub: https://github.com/IIIM-IS/AERA_Visualizer — *accessed 25th of April 2024.*

using backward chaining in their reasoning apparatus can be extended to include ABA in their reasoning process with little to no interference in other reasoning steps like forward-chaining or induction, making this a valuable extension to the existing work in reasoning-based AI and artificial general intelligence as a whole.

Acknowledgments. This work was funded in part by the Icelandic Research Fund (IRF) (grant number 228604-051) and a research grant from Cisco Systems, USA. The authors would like to thank the rest of the GMI research team at CADIA, Reykjavik U., for extensive discussions on the topics of this paper.

Disclosure of Interests. The authors have no competing interests to declare.

References

1. Belenchia, M., Thórisson, K.R., Eberding, L.M., Sheikhlar, A.: Elements of task theory. In: Proceedings of the International Conference on Artificial General Intelligence. Springer (2021)
2. Bondarenko, A., Dung, P.M., Kowalski, R.A., Toni, F.: An abstract, argumentation-theoretic approach to default reasoning. Artificial intelligence **93**(1-2), 63–101 (1997)
3. Bondarenko, A., Toni, F., Kowalski, R.A.: An assumption-based framework for non-monotonic reasoning (1993)
4. Dung, P.M., Kowalski, R.A., Toni, F.: Assumption-based argumentation. Argumentation in artificial intelligence pp. 199–218 (2009)
5. Eberding, L.M., Thórisson, K.R.: Causal reasoning over probabilistic uncertainty. In: International Conference on Artificial General Intelligence. pp. 74–84. Springer (2023)
6. Li, Y.: Deep reinforcement learning: An overview. arXiv preprint arXiv:1701.07274 (2017)
7. Linardatos, P., Papastefanopoulos, V., Kotsiantis, S.: Explainable AI: A review of machine learning interpretability methods. Entropy **23**(1), 18 (2020)
8. Nivel, E., Thórisson, K.R.: Towards a programming paradigm for control systems with high levels of existential autonomy. In: International Conference on Artificial General Intelligence. pp. 78–87. Springer (2013)
9. Nivel, E., Thórisson, K.R., Steunebrink, B., Dindo, H., Pezzulo, G., Rodriguez, M., Corbato-Hernandez, C., Ognibene, D., Schmidhuber, J., Sanz, R., Helgason, H.P., Chella, A., Jonsson, G.K.: Bounded recursive self-improvement. RUTR 13006 (2013)
10. Thompson, J., Thórisson, K.R.: ABA argument graphs with constraints. In: International Conference on Computational Models of Argument (in press, 2024)
11. Thórisson, K.R.: The 'Explanation Hypothesis' in general self-supervised Learning. Proceedings of Machine Learning Research **159**, 5–27 (2021)
12. Thórisson, K.R., Rörbeck, H., Thompson, J., Latapie, H.: Explicit goal-driven autonomous self-explanation generation. In: International Conference on Artificial General Intelligence. pp. 286–295. Springer (2023)
13. Toni, F., Craven, R.: Argument graphs and assumption-based argumentation (2016)
14. Walton, D.: Argumentation theory: A very short introduction. In: Argumentation in artificial intelligence, pp. 1–22. Springer (2009)