KR ARGUMENTATION PROJECT REPORT

Vrije Universiteit Amsterdam
XM_0059: Knowledge Representation

Argumentation
Group 21

14 / 12 / 2023

ABSTRACT

This report explores the realm of Knowledge Representation with a focus on understanding and evaluating Argumentation Frameworks, which play a crucial role in various decision-making and problem-solving applications. The study involves the development and implementation of a tool to interact with AFs, allowing users to engage in interactive gameplay against a computer/proponent. The tool addresses two distinct tasks: a Preferred Discussion Game and a Credulous Decision Problem based on stable semantics. They aim to serve as instructive elements to help the reader understand and internalize the mechanisms of AFs.

***K*eywords** Argumentation Frameworks · Discussion Games · Credulous Decisions

1 Introduction

In the realm of knowledge representation (KR), a profound comprehension of argumentation frameworks (AFs) is indispensable for applications that extend far beyond conventional decision-making and problem-solving scenarios (Zafarghandi et al., 2019). AFs play a pivotal role in various real-world applications, ranging from legal reasoning, where they meticulously model and evaluate legal arguments within the structures of legal cases, to supporting lawyers in the intricate process of decision-making (Prakken, 2017).

This report delves into argumentation frameworks (AFs) and their semantics, seeking an understanding of their mechanisms and practical implementations.

We develop and implement a tool which evaluates a loaded argumentation framework. This tool serves as a medium for interaction with an AF, elucidating their dynamics through interactive gameplay against computer/proponent. The tool is designed to handle AFs inputted into the system in JSON format and is capable of addressing two distinct tasks: A Preferred Discussion Game (PDG) and a Credulous Decision Problem (CDP) based on stable semantics. Both implementations are separately discussed in sections 3 and 4 of this paper, respectively. Through these functionalities, we aim to facilitate users in exploring and understanding the dynamics of arguments within the given frameworks.

2 Argumentation Frameworks

An Argumentation Framework (AF) is a tuple of atomic arguments and binary relations (attacks) between arguments. The most well-known is Dung's (1995) abstract argumentation framework, formalised as a pair $AF = (A, R)$ with A being a set of arguments and R a set of attack relations. For example, when we construct an AF, $F = (A, R)$ s.t. $A = (a, b, c, d, e)$ and $R = \{(a, b), (c, b), (c, d), (d, c), (d, e), (e, e)\}$, 'a' attacks 'b', 'c' attacks 'b' and so on. This example AF is depicted visually in the figure below.

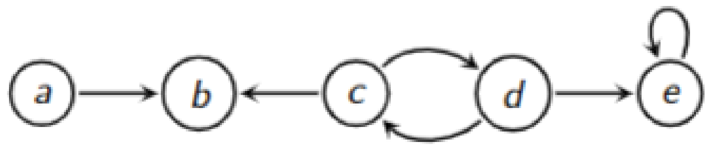

Figure 1: AF from Example

Formal argumentation theory stands out as an advantageous approach for defining non-monotonic reasoning due to its basis on concepts similar to human reasoning, such as arguments and debate (Booth et al., 2018). Motivated by the inherent complexity of decision-making processes and the need for transparent, rational discourse, these frameworks offer a unique approach to modelling and analyzing arguments. Furthermore, argumentation frameworks are not merely theoretical constructs but also imperative tools that bring transparency and efficiency to various fields.

2.1 Background

In the realm of Artificial Intelligence (AI), formal argumentation can be leveraged for creating more explainable AI (Vassiliades et al., 2021). As AI becomes more and more integrated in our daily lives, it brings forth the challenge of understanding and trusting automated decision-making processes. Well-designed AFs can help to make AI more understandable by giving clear, explainable reasoning about why AI makes certain choices.

In the work of Zhong et al. (2014), the authors define an AF that takes into consideration information from the Knowledge Base (KB) to make a decision using similar decisions. The framework first parses the text of the argument and extracts the most important features (nouns, verbs). Then, it compares with the decisions in the KB, returning the most similar, with respect to the quantity of common features. The framework can back up its decision with arguments on how similar the two cases are and uses arguments which were stated for the similar case in the KB (Zhong et al., 2014).

Comparatively, Zeng et al. (2018) use a Decision Graph with Context (DGC), to understand the context of an argument and provide decision-making support. A DGC is a graph, where the decisions are represented as nodes, and the interactions between them (attacking and supporting) as edges. The authors map the DGC in an Assumption-Based Argumentation Framework (ABA) by considering decisions as arguments and the interactions between them as attack and support relations. Then, if a decision is accepted in the ABA, it is considered a good decision (Zeng et al., 2018).

Since its inception, argumentation has played a pivotal role in the development of explainable systems, evolving into one of the foremost reasoning methods for achieving explainability. Illustratively, in the legal domain, the intersection of Argumentation and Law extends beyond the realms of Computer Science and AI. Law, as a discipline, inherently involves lawyers seeking knowledge to counter each argument against their clients. Henry Prakken (2017), a pioneering computer scientist, has notably contributed by defining a formalization for AFs used in law. In his 2017 paper, Prakken (2017) outlines a process wherein a legal case undergoes scrutiny by an AF aiming to generate defeasible rules, essentially treated as arguments. This process culminates in the compilation of an AF, providing a structured representation of the legal case within Argumentation theory.

3 Preferred Discussion Games

As discussed by Booth et al. (2018), their web-based implementation of a discussion game, designed to determine the acceptance of an argument within a given semantics, can be used as a proof procedure for different argumentation semantics. This system is implemented with a nice interactive representation, providing a practical, hands-on method of interacting with, understanding, and evaluating various argumentation semantics.

This project serves as a basis for our stripped-down implementation of a PDG, which runs in a simple text-based environment directly from the command line.

3.1 Game Rules

During each round, the proponent (played by the computer) presents an argument, and the opponent (the user) can select from various options to challenge this argument. At the start, the proponent provides the argument specified as a parameter to the program. The game follows these guidelines:

- The opponent is limited to choosing arguments that counter a previously presented argument by the proponent (from the current round or any previous round).
- The proponent responds with an argument that counters the opponent's chosen argument from the immediately preceding round.
- The opponent is prohibited from using the same argument more than once, whereas the proponent is exempt from this restriction.
- If the opponent uses an argument previously employed by the proponent, or the proponent uses an argument previously used by the opponent, the opponent emerges victorious, illustrating a contradiction on the proponent's part.
- If the proponent is unable to present a counterargument, the opponent wins.
- If the opponent exhausts all available choices, the proponent is declared the winner.

3.2 Implementation

Our script simulates a simple interactive discussion game based on an argumentation framework, allowing players to make strategic choices and respond to given arguments. To begin the game, the user can call the python programme from the command line, giving the path to the JSON file and the proponent's initial argument as additional arguments.

```
python preferred_discussion.py input/ex3.json c
```

Using the *NetworkX* and *MatPlotLib* packages for Python, our implementation begins by automatically generating an AF graph based on the provided input - like the one below, generated from the example AF given in the project description. This graphical depiction should help the opponent to better visualise the system of arguments and make informed decisions during gameplay. The code also prints the set of arguments and attacks, for the player's reference.

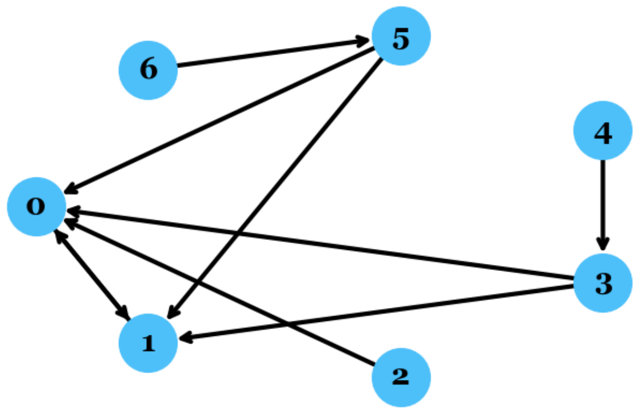

Figure 2: Example AF Graph

Starting the first round, the computer outputs the proponent's first argument, given in the function call, and asks for the player's input. The player much then input one of the argument's from the framework, which can be checked against all of the implemented rules. If all the rules are passed, the programme prints the lists of the arguments used by both the proponent and opponent, for the player's reference. The programme then begins onto the next round, choosing a new argument on behalf of the proponent, requesting an input argument from the player, and so on. These rounds are repeated until either the opponent or proponent wins, based on the conditions described above. An entire winning gameplay cycle can be seen in the following example figure.

```
PS C:\Users\yoshm\VU\Knowledge Representation\ARG> python .\preferred_discussion.py input/ex3.json c

The Argumentation Framework
Arguments: ['a', 'b', 'c', 'd', 'e']
Attacks: ['a -> b', 'b -> a', 'b -> c', 'c -> d', 'd -> e', 'e -> c']

Round 1
Proponent's Chosen Argument: c
Choose an attacking argument: b
Proponent's Used Arguments: ['c']
Arguments You've Used: ['b']

Round 2
The proponent attacks your argument with: a
Choose an attacking argument: e
Proponent's Used Arguments: ['c', 'a']
Arguments You've Used: ['b', 'e']

Round 3
The proponent attacks your argument with: d
Choose an attacking argument: c
Proponent's Used Arguments: ['c', 'a', 'd']
Arguments You've Used: ['b', 'e', 'c']

You show that the proponent's argument contradicts itself.
You win!
```

Figure 3: Example Gameplay

4 Credulous Acceptance

The semantics of argumentation frameworks play a pivotal role in determining which sets of arguments are deemed acceptable, preferred, or stable, offering different perspectives into the dynamics of reasoning and decision-making processes. A given semantics provides us with the rules and criteria for various decisions, one of which is credulous acceptance. For an argument to be credulously acceptable under a certain semantic in a given AF, it must be contained in at least one of the extensions of AF which are valid under the same semantics.

In the second part of the project, we have implemented a decision process for credulous acceptance under stable semantics. This semantics was chosen due to its relatively simple rules and intuition.

A stable set of arguments must be conflict-free, and also every argument outside this set must be attacked by at least one argument inside the set. This captures a notion of conflict avoidance and stability, which can be intuited as separating all the arguments into two conflicting parts with no argument left in between.

We shoukd note that not all AFs have stable extensions, and if there is no stable extension in a given AF, then straightforwardly no argument is credulously accepted.

4.1 Implementation

Our implementation of the credulous decision program prioritizes simplicity over efficiency for the sake of being instructive. The first part of the program receives as input an argumentation framework and an argument, and utilizes Python's *itertools* to create every possible stable extension of the given set of arguments. This last part is not necessary when deciding credulous acceptance as it is enough to prove that one stable set of the given AF contains the input argument. If it is contained in one, then it is credulously acceptable. The second part of the program checks whether the input argument is contained in any of the stable sets. If it is found, the program terminates.

```
PS C:\Users\yoshm\VU\Knowledge Representation\ARG> python stable_credulous_decision.py input/ex3.json b
Arguments: ['a', 'b', 'c', 'd', 'e']
Attacks: ['a -> b', 'b -> a', 'b -> c', 'c -> d', 'd -> e', 'e -> c']

Stable sets:  [{'b', 'd'}]

All stable sets found in 0.41039999632630497 milliseconds

+ Positive, the argument is in at least one stable set.

Elapsed time: 0.685 milliseconds
```

Figure 4: Example CDP Output

4.2 Results

As instructed, several credulous decision runs are presented in Table 1 for the following AFs:

$F_1 = (A, R), A = \{a, b, c, d, e, k\}, R = \{(e, d), (d, b), (b, a), (c, b), (a, k)\}$

$F_2 = (\{a, b, c, d, e\}, \{(a, b), (c, b), (c, d), (d, c), (d, e), (e, e)\})$

$F_3 = (\{a, b, c, d, e\}, \{(a, b), (b, a), (b, c), (c, d), (d, e), (e, c)\})$

$F_4 = (\{a, b, c, d, e, f, g, h, i\}, \{(a, b), (b, a), (a, c), (d, b), (b, g), (g, h), (h, c), (c, f), (f, e), (e, h), (i, e)\})$

The fourth AF was added by hand to serve as a larger example.

Table 1: Credulous Acceptance Under Stable Semantics

AF	Size (#A)	Input Argument	Credulous Acceptance	Run Time (ms)
F_1	6	a	Positive	0.29
F_1	6	k	Negative	0.30
F_2	5	c	Negative	0.21
F_2	5	d	Positive	0.22
F_2	5	a	Positive	0.22
F_3	5	b	Positive	0.20
F_3	5	d	Positive	0.19
F_3	5	e	Negative	0.21
F_4	9	g	Positive	0.56
F_4	9	c	Negative	0.58
F_4	9	h	Negative	0.55

Since the algorithm at hand creates all stable sets of the given AF by going over every possible subset, the run times are correlated with the number of arguments present in the AF. F_2 and F_3 contain 5 arguments each, the fewest among all, and consequently have the shortest run times while F_4 takes the longest to run because it contains the most arguments.

5 Conclusion

This report has explored the area of Knowledge Representation with a specific focus on Argumentation Frameworks (AFs). The development and implementation of an interactive tool for AFs have been discussed, addressing two main tasks: a Preferred Discussion Game and a Credulous Decision Problem based on stable semantics.

The Preferred Discussion Game provides an interactive platform for users to engage in strategic gameplay against a computer/proponent, demonstrating the dynamics of argumentation within a given framework. Subsequently, the Credulous Decision program serves as an instructive example discussing which arguments can be credulously accepted under stable semantics in a given AF.

The study emphasizes the significance of AFs in decision-making processes, with applications ranging from legal reasoning to explainable AI. The interactive tool and implemented decision process serve as valuable building blocks for exploring and understanding argumentation dynamics in various contexts.

As future work, the tool's functionalities could be extended to incorporate more advanced argumentation semantics and further enhance user interactivity and interface. Additionally, the decision program should be optimized for efficiency, especially when dealing with larger argumentation frameworks.

References

Booth, R., Caminada, M., & Marshall, B. (2018). DISCO: a Web-Based implementation of discussion games for grounded and preferred semantics. *Computational Models of Argument*, 453–454. https://doi.org/10.3233/978-1-61499-906-5-453

Dung, P. M. (1995). On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games. *Artificial Intelligence*, *77*(2), 321–357. https://doi.org/10.1016/0004-3702(94)00041-x

Prakken, H. (2017). Logics of argumentation and the law. In P. Glenn & L. Smith (Eds.), *Law and the new logics* (pp. 3–31). Cambridge University Press. https://doi.org/10.1017/9781316227329.002

Vassiliades, A., Bassiliades, N., & Patkos, T. (2021). Argumentation and explainable artificial intelligence: A survey. *The Knowledge Engineering Review*, *36*. https://doi.org/10.1017/S0269888921000011

Zafarghandi, A. K., Verbrugge, R., & Verheij, B. (2019). *Discussion games for preferred semantics of abstract dialectical frameworks*. https://doi.org/10.1007/978-3-030-29765-7\{_}6

Zeng, Z., Fan, X., Chen, M., Leung, C., Jih, C. J., & Soon, O. Y. (2018). Context-based and Explainable Decision Making with Argumentation. *Adaptive Agents and Multi-Agents Systems*, 1114–1122. https://dr.ntu.edu.sg/bitstream/10356/88889/1/aamas-2018-camera-final.pdf

Zhong, Q., Fan, X., Toni, F., & Luo, X. (2014). Explaining best decisions via argumentation. *European Conference on Social Intelligence*. https://api.semanticscholar.org/CorpusID:5282734