On Explanation of Propositional Logic-based Argumentation System

Teeradaj Racharak and Satoshi Tojo
School of Information Science, Japan Advanced Institute of Science and Technology, Ishikawa, Japan

Keywords: Deductive Argumentation, Argumentation System, Explainable Artificial Intelligence, Natural Deduction.

Abstract: We present a characterization about argumentation and proof in logic. Indeed, we show that proof for a claim α from a set of premises Φ can be deemed as a structured form of an argument for that claim. Due to the expressivity of classical propositional logic (PL), this work considers that the knowledge-base is represented in PL, in which the semantics and proof systems for individual arguments are studied and utilized. We show that natural deduction (ND) can be used as a basis of proof for an argument and also for modeling counterarguments in the form of canonical undercut. We reveal that ND does not merely enable for the construction of arguments but also paves the way naturally for a human-understandable form of argumentative reasoning. Finally, we demonstrate that our approach gives the feasibility of developing explainable artificial intelligence systems that can offer human-friendly explanations to the users.

1 INTRODUCTION

Argumentation is an important aspect of human intelligence. Humans always search for pros and cons of arguments as well as their consequences when attempting to understand a facing situation for making decisions. This argumentative reasoning can be formalized by utilizing a logical language for the premises and an appropriate consequence relation for showing that claims logically follow from the premises (*a.k.a. logic-based argumentation*).

There are a number of proposals for logic-based formalization of argumentation (*cf.* (Besnard and Hunter, 2018; Chesñevar et al., 2000; Vreeswijk and Prakken, 2001) for the existing literature). These works allow the representation of arguments for claims, the representation of counterarguments against them, and the relationships between the arguments. Despite the diversity, an argument in logic-based argumentation is commonly defined as a pair of which the first item is a set of formulae that proves the second item (*i.e.* a logical formula). There have been several investigation of and success with the use of proof techniques in logic. For instance, (Prakken and Sartor, 1997) developed proof procedures to find acceptable arguments in Dung's semantics from a defeasible logic knowledge-base. As an example in propositional logic knowledge-base, (Efstathiou and Hunter, 2011) proposed to generate arguments and counterarguments using the resolution principle and connected graph (Kowalski, 1975; Kowalski, 1979). Unfortunately, these approaches do not concretely offer computational content of an argument in a form that is understandable by naive users. This is a vital aspect of developing explainable artificial intelligence systems; reasoners should provide understandable explanations in order to facilitate the process of evolving the theory between explainers and explainees (*i.e.* a group of people who receive the explanations).

Here, we suppose that a knowledge-base Δ is represented by classical propositional logic (PL); thereby proof theories in PL are investigated for construction of arguments and counterarguments from Δ. Formally, finding an argument for claim α involves seeking for a consistent subset Φ of Δ which can logically derive α, *i.e.*, one can prove the validity of α from Φ using some proof systems. This basically amounts to investigate well-established proof theories of the base logic towards arguments' construction. Figure 1 (a) depicts an example of applying the resolution technique (as adopted in (Efstathiou and Hunter, 2011)) to prove that q is a valid consequence from $\{p, \neg p \vee q\}$, in which $\{\}$ denotes the empty clause.

Researchers have put an essentially great deal of effort into development of structured argumentation framework, but addressing understandable computed content of argumentation models have received less attention. Nevertheless, we might utilize procedures which offer to generate an adequate explanation for a developed argument. Considering Figure

Racharak, T. and Tojo, S.
On Explanation of Propositional Logic-based Argumentation System.
DOI: 10.5220/0010318103230332
In *Proceedings of the 13th International Conference on Agents and Artificial Intelligence (ICAART 2021) - Volume 2*, pages 323-332
ISBN: 978-989-758-484-8

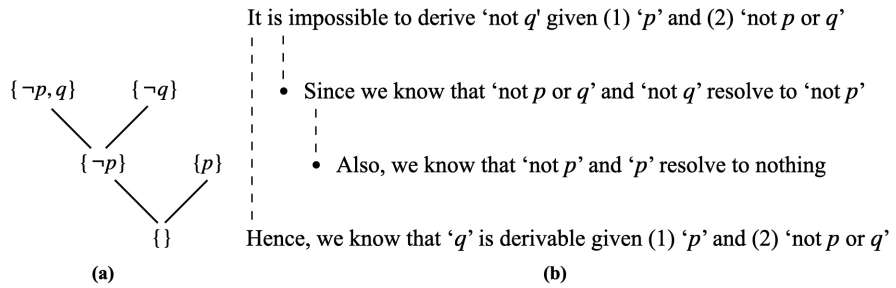

Figure 1: (a) A resolution proof for claim q given assumptions p and $\neg p \vee q$; (b) A corresponding explanation of the proof.

1 (b) as a motivating example, one may observe that an argument developed from the resolution carry too much information than the need for a meaningful explanation of why 'q' given assumptions 'p implies q' and 'q'. While there exist more diverse procedures in logic (*e.g.* analytic tableaux (Smullyan, 1995) and sequent calculus (Takeuti, 2013)), this work argues for using natural deduction (ND) (Gentzen, 1935), taken as a mean to identify an argument's structure from proof. We demonstrate that the pattern represented by ND is close to what humans can perceive as an argument drawing a conclusion from any conjunction that it contains. We elaborate upon our formalization based on ND proof in Section 3.

It is worth mentioning that current studies on logic-based argumentation have mostly concerned on exploiting logic for modeling structured argumentation such as (Besnard and Hunter, 2018); however, how it contributes to the development of explainable artificial intelligence (XAI) systems is not fully investigated. Thus, this work aims at bridging this gap between argumentation and its applications on XAI.

The contribution in this paper is that we introduce an approach to modeling arguments based on ND calculus towards the development of XAI systems. Our approach offers three main advantages: (1) explicit information used to build up arguments, (2) a transparent connection between the supports and the claim corresponding to the consequence relation, and (3) obvious translation for generating a human-friendly argument from the proposed formalization. The use of ND supports explanation generation from the computed deductive arguments; and also, the use of argumentative proof procedure coincides with everyday explanations used by humans (*cf.* Section 4). We review the basic elements in (Dung's) abstract argumentation and classical propositional logic including the natural deduction in Section 2. Section 5 relates our approach to others. Finally, Section 6 provides a conclusion and discussion of future directions.

2 PRELIMINARIES

2.1 Abstract Argumentation

Abstract argumentation (AA) provides a good starting point for formalizing argumentation in human reasoning. In (Dung, 1995), an AA framework is a pair $\langle \mathcal{A}, \mathcal{R} \rangle$ of which $\mathcal{A}$ represents a set of arguments and $\mathcal{R} \subseteq \mathcal{A} \times \mathcal{A}$ represents attack between arguments. Arguments may attack each other and thereby their statuses are subject to an evaluation. Semantics for AA returns sets of arguments called *extensions*, which are *conflict-free* and *defend* themselves against attacks.

Formally, a set $S \subseteq \mathcal{A}$ of arguments is conflict-free iff there are no arguments $A, B \in S$ such that $(A,B) \in \mathcal{R}$. Moreover, S defends $A \in \mathcal{A}$ iff, for any argument $B \in \mathcal{A}$, $(B,A) \in \mathcal{R}$ implies an existence of $C \in S$ such that $(C,B) \in \mathcal{R}$. A conflict-free set S is *admissible* iff each argument $A \in S$ is defended by S. These conflict-freeness and admissibility properties form the basis of all AA semantics as follows. Let $\mathsf{Defended}(S) := \{A \mid S \text{ defends } A\}$ be a function which yields a set of arguments defended by a certain set. Then, set S is a *complete extension* iff S is conflict-free and $S = \mathsf{Defended}(S)$; set S is a *grounded extension* iff it is the minimal complete extension (w.r.t. set inclusion); set S is a *preferred extension* iff it is a maximal complete extension (w.r.t. set inclusion); and set S is a *stable extension* iff S is conflict-free and S attacks every argument which is not in S.

2.2 Propositional Logic and Proof

In AA, the structure and meaning of arguments and attacks are abstract. On the one hand, the abstract definition enables to study properties which are independent of any specific aspects (Baroni and Giacomin, 2009). On the other hand, this generality features a limited expressivity and can be hardly adopted to model practical target situations. To fill out this gap, less abstract formalisms were considered, dealing in particular with the construction of arguments

and the conditions for an argument to attack another *e.g.* ASPIC$^+$ (Modgil and Prakken, 2014), DeLP (García and Simari, 2004), and assumption-based argumentation (ABA) (Dung et al., 2009). There have also been some investigation of logic-based argumentation in which well-established logical frameworks are used to model arguments and the relationship between them (*cf.* Section 5). Since the expressivity of classical propositional logic (PL) allows for modeling sufficient knowledge-base, this work focuses on a knowledge-base expressible in PL. Some preliminary definitions are provided below for self-containment.

Let $\mathcal{L}$ be a PL language obtained from a given set of atoms with $\neg$, $\wedge$, $\vee$, and $\rightarrow$ as connectives. Though all PL formulae can be formulated by using only $\neg$ and $\wedge$, we also include others to simplify the presentation of our approach. We also assume that $\mathcal{L}$ contains the special symbol $\perp$ representing inconsistency. To show that a sentence is derivable (or provable), we use the following natural deduction (ND) rules for any $\phi, \psi \in \mathcal{L}$ (Van Dalen, 2004):

$$\frac{\phi \quad \psi}{\phi \wedge \psi}\ (\wedge I) \qquad \frac{\phi \wedge \psi}{\phi}\ (\wedge E) \qquad \frac{\phi \wedge \psi}{\psi}\ (\wedge E)$$

$$\frac{\phi \quad \neg\phi}{\perp}\ (\neg E) \qquad \frac{\perp}{\phi}\ (\perp) \qquad \frac{\begin{array}{c}[\neg\phi]\\ \vdots \\ \perp\end{array}}{\phi}\ (RAA)$$

$$\frac{\begin{array}{c}[\phi]\\ \vdots \\ \psi\end{array}}{\phi \rightarrow \psi}\ (\rightarrow I) \qquad \frac{\phi \rightarrow \psi \quad \phi}{\psi}\ (\rightarrow E)$$

$$\frac{\phi}{\phi \vee \psi}\ (\vee I) \qquad \frac{\psi}{\phi \vee \psi}\ (\vee I) \qquad \frac{\phi \vee \psi \quad \begin{array}{c}[\phi]\\ \vdots \\ \sigma\end{array} \quad \begin{array}{c}[\psi]\\ \vdots \\ \sigma\end{array}}{\sigma}\ (\vee E)$$

Derived consequences result from applying these ND rules in sequence and we denote $\Phi \vdash \phi$ if ϕ is derivable from Φ. This search can be performed in the forward direction, from Φ to ϕ, in the backward direction, from ϕ to Φ, or even from both directions concurrently (Ferrari and Fiorentini, 2015). Our definition of argument (*cf.* Section 3) insists on the backward generation of arguments by applying ND rules with formulae in a knowledge-base.

Example 2.1. For a knowledge-base $\Delta := \{b \rightarrow a, c \rightarrow b, c \wedge b\}$, where a, b, c represents 'avoid steroids', 'get vaccine against hepatitis B', and 'plan to visit Africa', respectively. In the following, we show that $\Delta \vdash a$:

$$\frac{b \rightarrow a \qquad \dfrac{c \wedge b}{b}}{a}$$

Naturally, reading this deduction tree from top to bottom corresponds to the following explanation; noted that one can also read the tree from bottom to top to obtain a similar natural language sentence:

1. It is assumed that c and b;
2. So is b from #1;
3. It is also assumed that b implies a;
4. So is a from #2 and #3.

Hence, we show that assuming 'b implies a' and 'c and b' derives 'a'.

3 NDSA FRAMEWORK: ND FOR STRUCTURED ARGUMENTATION

Observe that a derivation in Example 2.1 corresponds to querying 'should we avoid steroids and why if so?'. Hence, it is quite natural to deem that ND for formula α represents a logical argument for claim α supported by a corresponding set of premises. Indeed, reading a ND tree from top to bottom (also, from bottom to top) yields an interpretation of a logical argument, allowing to extract naturally a human-friendly explanation as to why the claim α is so.

Definition 3.1 (ND Argument)**.** Given a PL knowledge-base Δ, an *argument* for claim α supported by $\Phi \subseteq \Delta$ (denoted by $\langle \Phi, \alpha \rangle$) is a ND proof tree such that α is derivable (backwards) from α to Φ and $\neg\alpha$ is not derivable from Φ.

Set Φ is called *supports* or assumptions; and also, α is called a *claim* of an argument. Observe that Definition 3.1 imposes consistency constraint to avoid the construction of illogical arguments (such as via *ex falso quodlibet*).

Unlike several other work, *e.g.* those of (Besnard and Hunter, 2018) and (García and Simari, 2004), we do not impose the restriction that the support of an argument be minimal. For instance, the same consequence as in Example 2.1 can be derived; however more verbose, as follows:

$$\frac{b \rightarrow a \qquad \dfrac{c \rightarrow b \qquad \dfrac{c \wedge b}{c}}{b}}{a}$$

This ND proof tree corresponds to the following logical argument for explaining why we should avoid steroids (a) given the assumptions (reading from top to bottom of the ND proof tree):

1. We know that c and b by our assumptions;

2. So is c from #1;
3. We also know that c implies b by our assumptions;
4. So is b from #2 and #3;
5. We also know that b implies a by our assumptions;
6. So is a from #4 and #5.

Though the above argument is not minimal, it is also relevant in the sense that their supports contribute to deducing the conclusion. Minimal checking is one way to ensure relevancy but may come at a computational cost. Nonetheless, our arguments are guaranteed to be relevant without imposing on minimality due to the backward generation of ND proof trees.

It is worth mentioning that applying ND is advantageous for us since the hypotheses appear only on top layers of a deduction tree, that suffices our prime goal of yielding human-friendly arguments. In comparison to other formalisms, a Hilbert-style axiomatization requires us to supply many axioms in the midst of a proof tree. As for the analytic tableau method, we need to show our goal to prove first on the top line, that is against our objective. Gentzen's sequent calculus (Kleene et al., 1952) might be the most polished style of deduction; however, each sequent becomes a long and messy sequence of formulae and is thus difficult for proof's visualization.

Given two arguments, it is possible to compare which argument is more general than one another. The following definition captures this relation between two arguments from a knowledge-base.

Definition 3.2. An argument $\langle \Phi, \alpha \rangle$ is *comparable to and more concise* than an argument $\langle \Psi, \beta \rangle$ iff $\Phi \subset \Psi$ and $\alpha \equiv \beta$.

From the above definition, one can say that argument $\langle \{b \to a, c \wedge b\}, a \rangle$ is comparable to and more concise than argument $\langle \{b \to a, c \to b, c \wedge b\}, a \rangle$.

Equipping argumentation into knowledge-base reasoning enables to deal with existence of inconsistent premises; derived conclusions of a knowledge-base are the claims of arguments in a concerned extension. Since a knowledge-base may be inconsistent, logical arguments constructed from the knowledge-base may be conflicting with each other. To define counterarguments, we consider the notion *classical direct undercut* (Besnard and Hunter, 2018), which is largely applied in the literature.

Definition 3.3. Let $A \coloneqq \langle \Phi, \alpha \rangle$ and $B \coloneqq \langle \Psi, \beta \rangle$ be arguments. Then, we say that argument A *attacks* argument B iff $\exists \phi \in \Psi$ such that $\alpha \equiv \neg \phi$.

It is worth noticing that if an argument is attacked by another argument, then other arguments which are less concise than it will also be attacked by the same counterargument. Hence, it is redundant to account for less concise arguments in argumentative reasoning and is omitted to show in our running examples. Following this idea, we are now ready to instantiate an abstract argumentation framework from a (possibly inconsistent) PL knowledge-base.

Figure 2 illustrates an instantiated abstract argumentation framework, in which each triangle represents an argument corresponding to each ND derivation. At each argument, the bottom part represents the claim, the top part denotes its supports, and arrows represent argument-counterargument relationship.

The proposed approach gives a straightforward way for instantiating an abstract argumentation framework from a PL knowledge-base. However, applying these definitions may cause infinite construction of arguments; thereby causing the attack relation among them to be also infinite. For instance, if an instantiated abstract argumentation framework contains argument $\langle \{a\}, a \rangle$, it also means that the instantiated framework contains arguments $\langle \{a\}, a \vee b \rangle$, $\langle \{a\}, a \vee b \vee c \rangle$, and so on using the $\vee I$-ND rule. This kind of infinite abstract argumentation frameworks may be hardly used and requires a special treatment to deal with the infinite construction. In the following, we consider a *core* of an argumentation framework (Amgoud et al., 2011) which can be identified by the following notions.

Definition 3.4 (Structural Equivalence of Arguments)**.** Arguments $A \coloneqq \langle \Phi, \alpha \rangle$ and $B \coloneqq \langle \Psi, \beta \rangle$ are *structurally equivalent* iff $\Phi = \Psi$ and $\alpha \equiv \beta$.

Definition 3.5. Let $\mathcal{F}' \coloneqq \langle \mathcal{A}', \mathcal{R}' \rangle$ and $\mathcal{F} \coloneqq \langle \mathcal{A}, \mathcal{R} \rangle$ represent different AA frameworks. Then, $\mathcal{F}'$ is a *core* of $\mathcal{F}$ iff $\mathcal{A}'$ and $\mathcal{R}'$ are finite; and, for any $A \in \mathcal{A}$, there exists $A' \in \mathcal{A}'$ such that A' and A are structurally equivalent and A' satisfies the following conditions:

- For any argument $B \in \mathcal{A}$ such that $(B, A) \in \mathcal{R}$, there also exists argument $B' \in \mathcal{A}'$ such that $(B', A') \in \mathcal{R}'$, and
- For any argument $B \in \mathcal{A}$ such that $(A, B) \in \mathcal{R}$, there also exists argument $B' \in \mathcal{A}'$ such that $(A', B') \in \mathcal{R}'$.

When multiple arguments are structurally equivalent, it is adequate to choose exactly one argument of them in an instantiated abstract argumentation framework. Restricted our attention on structurally equivalent arguments, one can identify a core of an argumentation framework. For instance, it can be shown that a core of an instantiated abstract argumentation framework in Example 2.1 contains three arguments as in Figure 3 in which A_1 denotes a derivation for c from $c \wedge b$, A_2 denotes a derivation for b from $c \wedge b$, and A_3 denotes a derivation for a from $c \wedge b$ together with $b \to a$. At implementation, this imposition can be re-

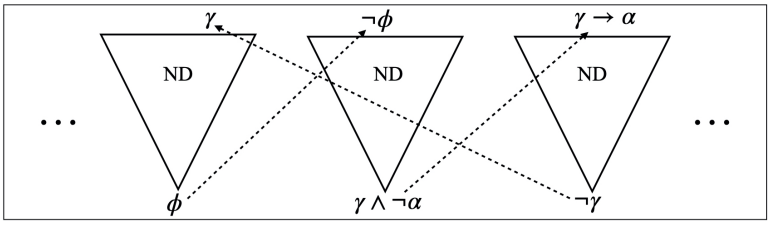

Figure 2: An instantiated abstract argumentation framework.

duced to checking whether $\alpha \wedge \neg\beta$ and $\beta \wedge \neg\alpha$ are unsatisfiable by any Boolean satisfiability (SAT) solver, although checking if two logical formulae are equivalent may come with a computational cost. Some more heuristic techniques for performance improvement may also be investigated and are remained as our future tasks.

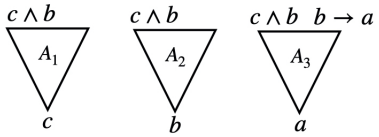

Figure 3: A core of an instantiated abstract argumentation framework.

To complete our earlier definitions of argument and attack, we formally give a definition of our natural deduction-based structured argumentation (NDSA) framework as follows.

Definition 3.6 (NDSA)**.** A NDSA framework is a triple $\langle \mathcal{L}, \Delta, \vdash_{ND} \rangle$, where $\mathcal{L}$ is a PL language, Δ is a knowledge-base modeled based upon language $\mathcal{L}$, and $\vdash_{ND}$ is a consequence relation represented by the natural deduction proof calculus.

Proposition 3.1. Let $\mathcal{F} := \langle \mathcal{A}, \mathcal{R} \rangle$ be an AA framework built according to NDSA. Then, $\mathcal{F}' := \langle \mathcal{A}', \mathcal{R}' \rangle$ is a core of $\mathcal{F}$ if it is constrained from $\mathcal{F}$ as follows:

- For each set $\mathsf{A} \subseteq \mathcal{A}$ of structurally equivalent arguments, only one argument A of A is non-deterministically included in $\mathcal{A}'$ and all attacks of arguments $\mathcal{A} \setminus A$ are excluded from $\langle \mathcal{A}, \mathcal{R} \rangle$ to yield $\langle \mathcal{A}', \mathcal{R}' \rangle$.

Proof. We show that the above construction yields a core of any argumentation framework $\langle \mathcal{A}, \mathcal{R} \rangle$. Fix any $\mathsf{A} \subseteq \mathcal{A}$ of structurally equivalent arguments, we show if an argument A of A is non-deterministically in $\mathcal{A}'$, then the following conditions hold:

- (Condition 1) Fix any argument $B \in \mathcal{A} \setminus \mathsf{A}$ such that $(B, A) \in \mathcal{R}$, we show that there exists an argument $B' \in \mathcal{A}'$ such that $(B', A) \in \mathcal{R}'$. Since A is a set of structurally equivalent arguments, then $(B, A) \in \mathcal{R}'$. This means $B' = B$. Therefore, this condition trivially holds.
- (Condition 2) Fix any argument $B \in \mathcal{A} \setminus \mathsf{A}$ such that $(A, B) \in \mathcal{R}$, we show that there exists an argument $A' \in \mathcal{A}'$ such that $(A', B) \in \mathcal{R}'$. Since A is a set of structurally equivalent arguments and A is a singleton, then we know $A' = A$. Therefore, this condition also trivially holds.

The above proposition provides us an algorithmic procedure to indicate a core of an abstract argumentation framework. In the following, we illustrates another (but less trivial) example about identifying a core of an abstract argumentation framework. It also highlights that the proposed framework can be used in multi-agent reasoning, in which all agents possess their own consistent set of the knowledge (but, their integration is inconsistent). For instance, it often occurs that witnesses in jurisdictions may hold different consistent sets of beliefs but integration among those beliefs turns to be inconsistent. This shows that NDSA allows to represent an argumentation dialogue and to detect the argument-counterargument interaction between each agent's utterance.

Example 3.1. Let $\Delta_A := \{a, a \rightarrow b\}$ and $\Delta_B := \{\neg b, a \rightarrow b\}$ represent different knowledge-bases possessed by Agents A and B, respectively, in which a denotes 'plan to visit Africa' and b denotes 'get vaccine against hepatitis B'. For the integrated knowledge-base $\Delta := \Delta_A \cup \Delta_B$, it can be shown that a core of an AA framework built from Δ according to NDSA is presented as in Figure 4.

Noted that, in the figure, A_1 represents a derivation for $\langle \{a\}, a \rangle$, A_2 represents a derivation for $\langle \{\neg b\}, \neg b \rangle$, A_3 represents a derivation for $\langle \{a \rightarrow b\}, a \rightarrow b \rangle$, A_4 represents a derivation for $\langle \{a, \neg b\}, a \wedge \neg b \rangle$, A_5 represents a derivation for $\langle \{\neg b, a \rightarrow b\}, \neg a \rangle$, and A_6 represents a derivation for $\langle \{a, a \rightarrow b\}, b \rangle$.

4 ACCEPTABILITY AND EXPLANATIONS OF NDSA

Since NDSA instantiates an AA framework from a knowledge-base, all semantics for determining the 'acceptability' of arguments in AA also apply to ND arguments. As a common approach in logic-based argumentation, consequences of a knowledge-base are claims of those arguments in a concerned extension.

Definition 4.1. Let $\langle \mathcal{L}, \Delta, \vdash_{ND} \rangle$ be a NDSA framework and $\mathsf{ext}(\mathcal{F})$ be an extension of an AA frame-

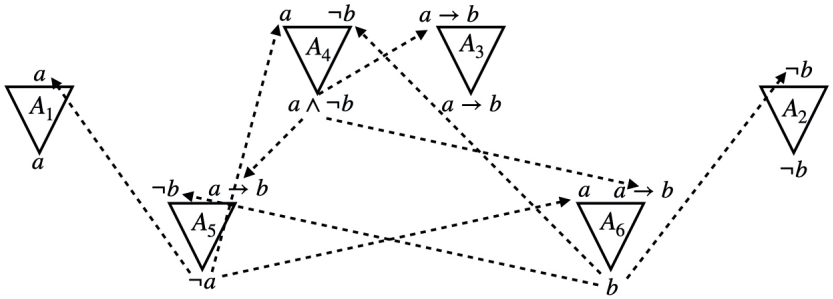

Figure 4: A core of an instantiated abstract argumentation framework.

work $\mathcal{F}$ built from the NDSA. Then, a set of consequences from Δ w.r.t. $\mathsf{ext}(\mathcal{F})$ (denoted by $\mathsf{Con}_{\mathsf{ext}(\mathcal{F})}$) is defined as: $\mathsf{Con}_{\mathsf{ext}(\mathcal{F})} := \{\alpha \mid \langle \Phi, \alpha \rangle \in \mathsf{ext}(\mathcal{F})\}$.

For instance, considering Example 3.1, applying stable semantics (*cf.* Subsection 2.1) yields three extensions: $\{A_1, A_3, A_6\}$, $\{A_2, A_3, A_5\}$, and $\{A_1, A_2, A_4\}$. Hence, there are three sets of accepted consequences w.r.t. the stable semantics from this knowledge-base: $\{a, a \to b, b\}$, $\{\neg b, a \to b, \neg a\}$, and $\{a, \neg b, a \wedge \neg b\}$.

Proposition 4.1. Let $\mathcal{F}' := \langle \mathcal{A}', \mathcal{R}' \rangle$ and $\mathcal{F} := \langle \mathcal{A}, \mathcal{R} \rangle$ represents AA frameworks. If $\mathcal{F}'$ is a core of $\mathcal{F}$, then the following holds for any concerned extension:

- For any $\alpha \in \mathsf{Con}_{\mathsf{ext}(\mathcal{F}')}$, there exists $\beta \in \mathsf{Con}_{\mathsf{ext}(\mathcal{F})}$ such that $\alpha \equiv \beta$; and
- For any $\beta \in \mathsf{Con}_{\mathsf{ext}(\mathcal{F})}$, there exists $\alpha \in \mathsf{Con}_{\mathsf{ext}(\mathcal{F}')}$ such that $\alpha \equiv \beta$.

Proof.

- (Condition 1) Fix any $\alpha \in \mathsf{Con}_{\mathsf{ext}(\mathcal{F}')}$, we show that there exists $\beta \in \mathsf{Con}_{\mathsf{ext}(\mathcal{F})}$ such that $\alpha \equiv \beta$. By assumption, we know that, for any $A \in \mathcal{A}$, there exists $A' \in \mathcal{A}'$ such that A' and A are structurally equivalent. Therefore, this condition trivially holds by Definitions 3.4 – 3.5.
- (Condition 2) Fix any $\beta \in \mathsf{Con}_{\mathsf{ext}(\mathcal{F})}$, we show that there exists $\alpha \in \mathsf{Con}_{\mathsf{ext}(\mathcal{F}')}$ such that $\alpha \equiv \beta$. By assumption, we know that, for any $A \in \mathcal{A}$, there exists $A' \in \mathcal{A}'$ such that A' and A are structurally equivalent. Therefore, this condition trivially holds by Definitions 3.4 – 3.5.

Notice that incorporating the computation of acceptability in AA together with ND in PL knowledge-base naturally represents formal reasoning used by humans, as (Gentzen, 1935)[1] and (Dung, 1995) aimed at this. Furthermore, separating levels of proof offers non-monotonic behavior for PL (which is a monotonic logic) since adding/removing arguments can surpass the acceptance of some arguments and thereby the corresponding claims are prevailed. Due to the well-investigated computational models of argumentation and ND proof in PL, we can outline an implementation of our proposed approach as shown in Algorithm 1. This algorithm finds all acceptable arguments w.r.t. a concerned extension.

Algorithm 1 : Finding accepted arguments in a NDSA framework $\langle \mathcal{L}, \Delta, \vdash_{ND} \rangle$.

```
1: input: a knowledge-base Δ, an AA semantics s
2: output: sets of acceptable arguments w.r.t. s
3: function ACCEPTEDARGUMENTS(Δ, s)
4:     Let G be an empty directed graph.
5:     G := Construction of an abstract argumentation framework based on Definitions 3.1 – 3.3.
6:     Indicate a core of G based on Proposition 3.1.
7:     Remove irrelevant arguments and attack relations (that are not part of the core) in G.
8:     exts := Sets of acceptable arguments w.r.t. the semantics s in G.
9:     return exts
10: end function
```

Regarding the tasks of providing their explanations, NDSA considers two-level interpretation on a core of an argumentation framework:

1. Explain why arguments are acceptable w.r.t. AA semantics;
2. Explain why accepted arguments are logically derived based on ND.

These two levels correspond to macro-scoping and micro-scoping explanations, respectively, for an accepted argument. The first level interpretation can be viewed as a debate between two fictitious agents (*i.e.* the proponent and the opponent) arguing why arguments in an extension should be accepted. On the other hand, for the second level interpretation, one can specifically zoom into a ND derivation of an argument as a basis for serving an explanation.

The following ND deduction proof exemplifies this intuition by explaining why argument A_6 (in Ex-

[1] "First, I wished to construct a formalism that comes as close as possible to actual reasoning. Thus, arose a *calculus of natural deduction*", quoted from (Gentzen, 1935).

ample 3.1) for claim 'get vaccine against hepatitis B' is derivable in the first stable extension:

$$\frac{a \quad a \to b}{b}$$

This inferential step enables to yield the following explanations to the users. As aforementioned, it corresponds to the micro-scoping level of explanation:

1. Given that 'plan to visit Africa' (a) and 'planning to visit Africa implies to get vaccine against hepatitis B' $(a \to b)$;
2. Hence, 'get vaccine against hepatitis B' (b).

We discuss more about the generation process of explanations on our two levels of explanations (*cf.* Subsection 4.1) and the evaluation of explanations in an aspect of good explanations based upon the viewpoint of human cognition in social science (*cf.* Subsection 4.2) in the following subsections.

4.1 Generation of Explanations for Claims of NDSA Arguments

This subsection briefly introduces the idea of providing explanations from NDSA, which basically comprise two forms of explanation as follows.

4.1.1 Dialogical Explanations

This explanation corresponds to the macro-scoping interpretation of our proposed framework. Indeed, given an abstract argumentation framework $\langle \mathcal{A}, \mathcal{R} \rangle$ instantiated from NDSA, one can explain the outcome of a concerned extension dialogically by reinterpreting a *dispute tree* $\mathcal{T}$ of an argument $A \in \mathcal{A}$, which can be constructed by the following procedure:

1. Every node of $\mathcal{T}$ is of the form $[L : B]$ where L is either *proponent* (P) or *opponent* (O) and $B \in \mathcal{A}$,
2. The root node of $\mathcal{T}$ is always labeled by $[\mathsf{P} : A]$,
3. For every node $[\mathsf{P} : B]$ of $\mathcal{T}$ with $B \in \mathcal{A}$, and for every $C \in \mathcal{A}$ with $(C,B) \in \mathcal{R}$, there exists a child of $[\mathsf{P} : B]$ which is labeled by $[\mathsf{O} : C]$,
4. For every node $[\mathsf{O} : B]$ of $\mathcal{T}$ with $B \in \mathcal{A}$, there exists exactly one child of $[\mathsf{O} : B]$ which is labeled by $[P : C]$ with $(C,B) \in \mathcal{R}$,
5. There are no other nodes in $\mathcal{T}$ except #1 – #4.

The proponent wins if he/she can counter-attack against every attacking argument by the opponent. The set of all arguments belonging to the proponent nodes in $\mathcal{T}$ is called the *defence set* of $\mathcal{T}$ (Dung et al., 2006). This defence set represents a reason for why a certain claim should be accepted.

Note that a branch in a dispute tree may be either finite or infinite. A finite branch represents a winning sequence of arguments that ends with an argument by the proponent in which the opponent is unable to attack. An infinite branch represents a winning sequence of arguments that the proponent counter-attacks every attack of the opponent ad infinitum. Several studies (Dung et al., 2006; Modgil and Caminada, 2009) have put an essentially great deal of effort into investigation of its winning strategies in order to help determining the membership of arguments in an extension of any abstract argumentation framework.

Definition 4.2. Let $\langle \mathcal{A}, \mathcal{R} \rangle$ be an abstract argumentation framework. A dispute tree $\mathcal{T}$ for $A \in \mathcal{A}$ is *admissible* iff no argument labels both P and O;

Theorem 4.1 ((Dung et al., 2009))**.** Let $\langle \mathcal{A}, \mathcal{R} \rangle$ be an abstract argumentation framework. We know:

1. If $\mathcal{T}$ is an admissible dispute tree for an argument $A \in \mathcal{A}$, then the defence set of $\mathcal{T}$ is admissible;
2. If $A \in S$ for an admissible set $S \subseteq \mathcal{A}$, then there exists an admissible dispute tree for A with defence set S' such that $S' \subseteq S$ and S' is admissible;

This work does not focus on the strategies; but rather, we employ this notion for explanation in a contrafactual argumentative situation for the consequences derived by NDSA (cf. Definition 4.1).

Intuitively, this form explains as a debate between a proponent P seeking to establish the acceptance of an argument in an extension and an opponent O seeking to withdraw such acceptance. For example, one can unfold the debate for the acceptance of A_1 in a stable extension $\{A_1, A_3, A_6\}$ in Example 3.1 as follows (*cf.* Figure 5). First, P moves argument A_1 representing the acceptability of A_1. Then, O puts forward argument A_5 representing the attack on a support of A_1. Then, P has to counter-argue O's argument by putting forward argument A_6 as it defends A_1. Next, O puts forward argument A_4 representing the attack on a support of A_6. It is in turn provided that this argument is counter-attacked by the same argument A_6 of P. We note that the same arguments A_4, A_5 put forward by O can also counter-attack A_6 of P; however, this attack and counter-attack relationship represents two winning sequences of arguments that the proponent counterattacks every attack of the opponent ad infinitum. We can handle this situation of an infinite branch by disabling its repetition on the dispute tree (*cf.* the dot lines in Figure 5). Since the arguments forwarded by P are unattacked by O, these sequences of argument moves indicate the acceptance of argument A_1. Dialogical explanations for other arguments in the extension can be obtained similarly.

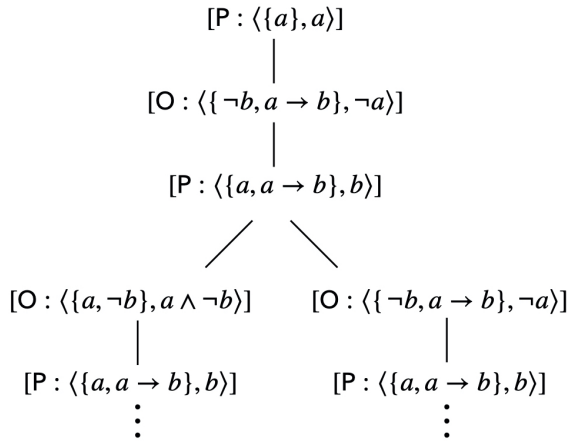

Figure 5: A dispute tree for argument $\langle\{a\}, a\rangle$ w.r.t. the NDSA framework in Example 3.1.

4.1.2 Logical Explanations

This explanation corresponds to the micro-scoping interpretation of our proposed framework. As we demonstrate earlier, the computational content of ND can facilitate to generate human-friendly logical explanations on theorem hood in a theory. It is worth noting that other proof systems may also be used as a basis to explanation generation; nonetheless, we observe that they may carry more information and express less intuitively to human interpretation. The idea to unfold explanations from a ND derivation is as follows. We trace on a ND proof tree from top to bottom (toward the conclusion of the proof) by a procedure that generates text corresponding to each inference step and tries not to repeat conjunctive particles (*e.g.* if – then, thus, hence, *etc.*). After that, we put together phrases derived from each subproof.

4.2 Evaluation of Explanations for Claims of NDSA Arguments

To evaluate the explanations generated from NDSA, we compare and relate with major findings on characteristics of good explanations in philosophy, cognitive psychology/science, and social psychology. Early on, most works devoted for explainable artificial intelligence use only the researchers' intuition of what constitutes a good explanation (Miller, 2019) and overlook the important insights from these research fields.

We fill in this gap by reviewing relevant work and demonstrating that NDSA can support XAI systems to develop human-like explanations, *i.e.* contrastive explanations (Subsubsection 4.2.1) and selected explanations (Subsubsection 4.2.2). In other words, an important outcome of this work is to draw a closer connection between argumentation theory and its applications towards the development of XAI systems.

4.2.1 Contrastive Explanations

Humans always seek in response to particular counterfactual cases (Hilton, 1990; Lipton, 1990). Research in related fields especially the social science shows that people do not explain the causes for an event *per se*, but rather explain the cause of an event relative to some other event that does not occur; that is, an explanation is of the form 'why α rather than β?', in which α represents the final conclusion and β represents an opposite outcome.

It is worth observing that our dialogical explanations are inherently *contrastive* by the construction; that is, to explain that α must occur, the explainer (the proponent) deliberates that a counterfactual β is not necessarily the case to the explainee (the opponent) by finding a counterargument to β. For instance, given a question "why should we plan to visit Africa?", the explainer may try to answer "why should not we plan to visit Africa?". To answer this negated question, the explainer can answer it by showing that "not visiting Afriva $(\neg a)$" is not necessarily the case, *i.e.*, with an answer "because we plan to visit Africa (a) and if we plan to visit Africa, then we get vaccine against hepatitis B $(a \rightarrow b)$". This indicates that our explanations satisfy this form of good explanations by naturally reinterpreting from the dialogical explanation.

4.2.2 Selected Explanations (w.r.t. a Context)

Research results on social psychology and cognitive science show that humans rarely expect an explanation to consist of both an actual and complete cause of a decision. But rather, they select one or two causes from a (possibly infinite) number of causes to be the explanation for the explainee (Hilton, 2017). Indeed, while a decision may have many argumentative claim-backings, often the explainee cares only about a small

subset (relevant to the context). That is, the explainer selects a subset of the possible explanations (based on different criteria), and the explainer and the explainee may interact and argue about these explanations.

It is worth observing that the generation of dialogical explanations can be tailored to selected explanations if the context of the explainee is given. For that, we consider only a branch in a dispute tree for the argument in which the selected branch coincides with the criteria specified in the context; the context may mean a concerned formula indicated by the explainee, the length of a considering branch of the dispute tree, and *etc.* For instance, according to Figure 5, there might be two possible selected explanations to be transmitted to a receiver of the explanation for why argument $\langle\{a\},a\rangle$ as the dispute tree has two branches. This indicates that our forms of explanation also enable to create selected explanations by manipulating on the generation of dialogical explanations.

5 COMPARISON WITH THE STATE OF THE ART WORK

This section compares our approach described in this paper with existing work on logic-based argumentation where the knowledge-base is formulated by PL.

(Efstathiou and Hunter, 2011) adopt the idea of connection graph (Kowalski, 1975; Kowalski, 1979) to model arguments from a knowledge-base Δ of clauses where each claim is a literal. At a high level, each node in a connection graph represents a clause in Δ; and, each edge connects nodes ϕ,ψ if there is a disjunct in $b \in \phi$ with its complement being a disjunct $\neg b \in \psi$. To find an argument for claim α, the authors considered the set of complements of the disjuncts of α together with Δ. Then, for any clause ϕ in the graph, if there is a disjunct $b \in \phi$ and there are no edges connecting ϕ,ψ where the complement of b is a disjunct in ψ, then the clause ϕ is deleted together with edges involving ϕ. This process of deletion is continued until no more clauses can be identified for deletion. If the resulting graph is non-empty, then it contains a set of formulae that entails α. Though the idea of connection graph can be used to find arguments, our proposed approach can model logical arguments more naturally where connections between the claim and its supporting premises are visualized explicitly; thereby our approach enables to extract explanation as more human-friendly arguments for non-technical users.

(Dung et al., 2009) introduced a more concrete framework for instantiating abstract argumentation called *assumption-based argumentation*, in which arguments can be constructed deductively from inference rules and the notion of attack is defined based on the contrary of an argument's claim. It is worth observing that the authors' proposal was also similar to ours in a sense that their arguments were modeled by applying *modus ponens* on inferential knowledge-base whereas our approach models arguments by applying ND rules to prove a claim (*i.e.* ND proof trees as logical arguments). The fact that we do not only focus on just one rule enables to model deductive arguments in a way that can be understood by naive users.

(Kakas et al., 2014) developed *argumentation logic* (AL), which can be viewed as an extension of PL. AL re-interpreted and extended PL to deal with inconsistency by modeling arguments from a set of PL formulae. Attack between two arguments in AL was defined based on a proof for inconsistency (inspired by *Reductio ad Absurdum* in ND) between two sets of PL formulae representing arguments. As for entailment, an argument is said to hold if it can be successfully defended and it cannot be successfully objected against. AL does not follow conventional semantics in AA; hence, the framework may infer different results to ours. For instance, let $\Delta := \{\alpha,\beta,\alpha\wedge\beta,\neg\alpha\vee\neg\beta\}$, the knowledge-base does not entail α in AL; however, our approach flavors ND for modeling arguments from sets of PL formulae and evaluates the acceptability of arguments based on semantics in AA. For example, α may be inferred from Δ if stable semantics is used or otherwise if grounded semantics is considered.

6 CONCLUSION AND FUTURE DIRECTIONS

This work presents an approach to a logic-based argumentation framework for reasoning with an (inconsistent) PL knowledge-base, especially in multi-agent reasoning in which each agent holds different set of (mutually inconsistent) knowledge, with an aim at the introduction of human-friendly explanations in argumentative reasoning. We show that good explanations investigated in cognitive science and social psychology can be formalized as a NDSA framework to develop an explainable artificial intelligence system.

Our approach exploits two main aspects of formalisms: (1) the naturalness of natural deduction and (2) argumentative semantics in AA. First, we utilize natural deduction proof in PL for finding valid arguments in a knowledge-base. While we are inspired by ND for modeling arguments from the knowledge-base, our approach can also be applied with other proof systems even though verbose explanations may be generated due to the proof procedure (*cf.* Figure

1). Indeed, a derivation for a formula is re-interpreted as an argument supporting that formula; and also, arguments supporting the contrary of the premises are seen as its attack. Second, when modeled arguments are in conflict, the notion of acceptability and semantics in AA are used to handle inconsistency. We believe that the computational content which brings together these two formalisms can generate human-friendly explanations on theorem hood in a theory.

It is widely accepted by now that answers of an intelligent systems should be able to explain for why to the users. Therefore, in the future, we would like to extend this idea for other logics (*e.g.* description logic and modal logic) and develop argumentation-based reasoning engines that offer human-friendly explanations to naive users for applying on real-world applications such as legal reasoning and ontology merging.

ACKNOWLEDGEMENTS

The authors would like to thank anonymous reviewers for valuable comments. This study was supported by JSPS KAKENHI Grant Number 17H02258.

REFERENCES

Amgoud, L., Besnard, P., and Vesic, S. (2011). Identifying the core of logic-based argumentation systems. In *2011 IEEE 23rd International Conference on Tools with Artificial Intelligence*, pages 633–636. IEEE.

Baroni, P. and Giacomin, M. (2009). *Semantics of abstract argument systems*, pages 25–44. Springer US, Boston, MA.

Besnard, P. and Hunter, A. (2018). A review of argumentation based on deductive arguments. *Handbook of Formal Argumentation*, pages 437–484.

Chesñevar, C. I., Maguitman, A. G., and Loui, R. P. (2000). Logical models of argument. *ACM Computing Surveys (CSUR)*, 32(4):337–383.

Dung, P. M. (1995). On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games. *Artif. Intell.*, 77(2):321–358.

Dung, P. M., Kowalski, R. A., and Toni, F. (2006). Dialectic proof procedures for assumption-based, admissible argumentation. *Artif. Intell.*, 170(2):114–159.

Dung, P. M., Kowalski, R. A., and Toni, F. (2009). Assumption-based argumentation. In Simari, G. R. and Rahwan, I., editors, *Argumentation in Artificial Intelligence*, pages 199–218. Springer.

Efstathiou, V. and Hunter, A. (2011). Algorithms for generating arguments and counterarguments in propositional logic. *International Journal of Approximate Reasoning*, 52(6):672–704.

Ferrari, M. and Fiorentini, C. (2015). Proof-search in natural deduction calculus for classical propositional logic. In *International Conference on Automated Reasoning with Analytic Tableaux and Related Methods*, pages 237–252. Springer.

García, A. J. and Simari, G. R. (2004). Defeasible logic programming: An argumentative approach. *Journal of Theory and Practice of Logic Programming*, 4(2):95–138.

Gentzen, G. (1935). Untersuchungen über das logische schließen. i. *Mathematische zeitschrift*, 39(1):176–210.

Hilton, D. (2017). Social attribution and explanation.

Hilton, D. J. (1990). Conversational processes and causal explanation. *Psychological Bulletin*, 107(1):65.

Kakas, A. C., Toni, F., and Mancarella, P. (2014). Argumentation logic.

Kleene, S. C., De Bruijn, N., de Groot, J., and Zaanen, A. C. (1952). *Introduction to metamathematics*, volume 483. van Nostrand New York.

Kowalski, R. (1975). A proof procedure using connection graphs. *Journal of the ACM (JACM)*, 22(4):572–595.

Kowalski, R. (1979). *Logic for problem solving*, volume 7. Ediciones Díaz de Santos.

Lipton, P. (1990). Contrastive explanation. *Royal Institute of Philosophy Supplement*, 27:247–266.

Miller, T. (2019). Explanation in artificial intelligence: Insights from the social sciences. *Artificial Intelligence*, 267:1–38.

Modgil, S. and Caminada, M. (2009). *Proof Theories and Algorithms for Abstract Argumentation Frameworks*, pages 105–129. Springer US.

Modgil, S. and Prakken, H. (2014). The ASPIC$^+$ framework for structured argumentation: A tutorial. *Argument and Computation*, 5(1):31–62.

Prakken, H. and Sartor, G. (1997). Argument-based extended logic programming with defeasible priorities. *Journal of applied non-classical logics*, 7(1-2):25–75.

Smullyan, R. M. (1995). *First-order logic*. Courier Corporation.

Takeuti, G. (2013). *Proof theory*, volume 81. Courier Corporation.

Van Dalen, D. (2004). *Logic and structure*. Springer.

Vreeswijk, G. and Prakken, H. (2001). Logical systems for defeasible argumentation. *Handbook of Philosophical Logic,*, 4:219–318.