Title	AN ARGUMENTATION-BASED FRAMEWORK FOR PRACTICAL REASONING
Author(s)	Gu, Chengwei
Citation	
Issue Date	2021-03
Type	Thesis or Dissertation
Text version	author
URL	http://hdl.handle.net/10119/17162
Rights	
Description	Supervisor: 東条　敏，先端科学技術研究科，修士（情報科学）

Master's Thesis

AN ARGUMENTATION-BASED FRAMEWORK FOR PRACTICAL REASONING

1810220 Gu Chengwei

Supervisor	Satoshi Tojo
Main Examiner	Nguyen Le Minh
Examiners	Kiyoaki Shirai
	Kazuhiro Ogata

School of Information Science
Japan Advanced Institute of Science and Technology
(Information Science)
February 2021

Abstract

Polarization, especially political polarization is becoming a severe problem all over the world. One of its peaks is at the 2020 United States presidential debates. Through watching the presidential debates, audience usually seek for information related to topics they concern about to help them decide which candidate they will vote for. However, the debates in 2020 were probably the most chaotic one in history. Audience might find themselves drown in quarrels and interruptions without perceiving what each candidate trying to say. We want to propose an application that can help the audiences to have a better understanding about the narrative of both candidates by visualizing how they arrange their own arguments and counterattack each other's, therefore, contributing to bridging the gap of polarization.

Researchers have introduced a various of ways to analyze the debate or, in general, dialogues. Traditionally, there is rhetoric analysis that done by expect focusing on the nuance, implicit messages as well as strategies a speaker might have. With the help of machine learning techniques, in recent years word frequency analysis and quazar graph come into existence. The word frequency analysis counts the occurrence of key words in a dialogue and visualize it in a tag cloud where the font size of each key word is directly proportional to its frequency. Quazar graph on the other hand treat key words as entities and visualize them in a graph where entities are nodes and each edge between two nodes indicates their co-occurrence within a speech.

Although these aforementioned approaches have their irreplaceable advantages, they may not serve our purpose as a whole. Rhetoric analysis is precise and profound, but it is overwhelming for people without sufficient background knowledge. Machine learning based approaches are automatic or semi-automatic but lack explicit explanation about how two entities interact.

Thus, we turn to logic based methods for they can present their reasoning procedure explicitly which not only suit our aim but also satisfy the requirements of explainable artificial intelligence (XAI). However, commonly used expert system such as ontology cannot deal with information that contains conflicts which is exactly the characteristics of a debate. Bear this in mind, we finally decide to build a system based on abstract argumentation frameworks that is instantiated following natural deduction based structured argumentation (NDSA) frameworks.

Our system is called NDSA based visualization. When an inconsistent knowledge base written in propositional logic that developed from a prepossessed debate or more generally a dialogue is inputted, out system will first model the knowledge base as a set of arguments and their attack relations and visualize it as a directed graph with arguments as nodes and attacks as edges. Moreover, following the insights in NDSA, we draw dispute trees as an explanation for an argument and its relationship with other arguments and derive a natural deduction proof as an explanation from the premises to the claim within an argument. We also translate the natural deduction proof to natural language explanations improving intelligibility.

We further apply our system to a fragment of the second presidential debate showing that our system can model real debate and provide human-friendly explanations to the users.

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

Acknowledgment

I would like to express my deepest gratitude to Associate Professor Teeradaj Racharak, Professor Satoshi Tojo and all lab members for their continues support and encouragement. Without them, I would not be able to complete my research.

List of Figures

2.1 An argumentation framework 16
2.2 Relation among extensions 20
2.3 Dispute trees for example Example 2.1 26
2.4 Fitch-style proof example 30
2.5 Block tableaux formats 33
2.6 Tableaux example 35

3.1 A core of an NDSA instantiated abstract argumentation framework. 39

4.1 layout of NDSA Framework Visualization 52

5.1 Layout for non-expert version 64

List of Tables

4.1 Layout for NDSA Visualization 53

5.1 Format of the knowledge base and a sample proposition 59
5.2 System Usability Scale Questionnaire 63
5.3 Layout for non-expert version 65

C.1 Full knowledge base 77

Contents

Abstract **I**

Acknowledgment **III**

List of Figures **V**

List of Tables **VII**

Chapter 1 Introduction **1**

1.1 Motivation . . . 1
1.2 Background . . . 3
1.2.1 XAI . . . 3
1.2.2 Nonmonotonic Reasoning . . . 5
1.3 Objectives . . . 6
1.4 Contributions . . . 7
1.5 Thesis Structure . . . 7

Chapter 2 Preliminaries **9**

2.1 Overview . . . 9
2.2 Abstract Argumentation . . . 12
2.2.1 Basic Concepts . . . 12
2.2.2 Semantics . . . 15
2.3 Assumption-based Argumentation . . . 21
2.4 Dispute Trees . . . 23
2.5 Natural Deduction For Propositional Logic . . . 27
2.5.1 Basic Concepts . . . 27
2.5.2 Fitch Notation . . . 30
2.6 Tableaux . . . 31

Chapter 3 NDSA Framework **37**

3.1 Overview . . . 37
3.2 Concepts . . . 37
3.3 Two-level Interpretation . . . 41
3.3.1 Dialogical Explanations . . . 41
3.3.2 Logical Explanations . . . 42

Chapter 4 System Design **43**
4.1 Overview . . . 43
4.2 Natural Deduction Prover . . . 44
4.3 Tableaux prover . . . 46
4.4 Argumentation Reasoner . . . 46
4.5 Natural language proof transformer . . . 51
4.6 Web interface . . . 51

Chapter 5 Evaluation **55**
5.1 Use case . . . 55
5.1.1 Annotation guideline . . . 56
5.1.2 Knowledge base description . . . 59
5.2 Unit Test . . . 60
5.2.1 Natural Deduction Prover . . . 60
5.2.2 Tableaux Prover . . . 61
5.3 Further evaluation proposal . . . 62

Chapter 6 Conclusion **67**
6.1 Conclusion . . . 67
6.2 Future Direction . . . 68

Appendices **71**

Appendix A Use Case **71**

Appendix B Transcript **73**

Appendix C Knowledge Base **77**

References **81**

Chapter 1

Introduction

1.1 Motivation

Polarization, especially political polarization is becoming a severe problem all over the world. With the advance of technology, growing number of people around the world now have easy access to the internet. At the same time, many of them watch and read news through internet and, to be specific, social media. Artificial Intelligence (AI), a long existing research field gain a new life, thanks to the major breakthrough in neural network and deep learning, has been helping social network platform such as Facebook and Twitter gaining billions and millions users with its unprecedented recommendation systems. Although initially created with good intentions, these AI-based recommendation systems are arguably the main cause of the division among the public opinions. In order to encourage users to spend more time on the platform, these systems only recommend articles and videos that they believe the users desire for. These systems even actively push controversial materials or misinformation to users as long as they know those things are intoxicating. It takes years for people to finally become acquainted with how devastating these AI-based recommendation systems can be. However, when people try to examine these system to better understand their behaviors and make modifications, they find that it is almost impossible to fix these systems due to the nature of neural network.

Black-box models is what we refer to these kind of models. As its name implies, black-box models are models that we can only observe their input and output but not their inside mechanism. Black-box models often have high prediction accuracy at the cost of high computation complexity, while at the same time are hard to interpret and be explained in human terms. While their pros make them irreplaceable, their cons let them get more and more concerns overtime. It is considered risky to apply a model over critical issues such as medicine and law without fully understand its manner. Therefore, as a response to this concern, researchers proposed the concept of explainable artificial intellgence (XAI). According to [1], the definition of explainability within the context can be:

> *Given a certain audience, explainability refers to the details and reasons a model gives to make its functioning clear or easy to understand.*

Then, XAI could be defined as follows:

> *Given an audience, an explainable artificial intelligence is one that produces details or reasons to make its functioning clear or easy to understand.*

From this definition, it is worth noticing that XAI takes the audience into account. Given the audience, an XAI model is required to provide explanation that suits the audience's purpose of use and level of knowledge.

Logic always have a tight relationship with computer science and its subfield in artificial intelligence. Knowledge representation and reasoning is a method that help people solve problems and make decisions by first represent knowledge using various of language modeled after certain types of logic in a form that is computer readable then solve tasks through reasoning applying predetermined inference rules. Because the inference rules are explicit, we can present the reasoning procedure that is understandable for the expert or even non-expert users with proper modifications. Such advantage makes Knowledge representation and reasoning method such as ontology a promising one following the requirements of XAI.

With a wish of bridging the gap of polarization, we focus on the 2020 United States presidential debates aiming to reconstruct the debate in this thesis. The first general United States presidential debates was held on 1960. Since then it becomes a convention of the election campaign at which candidates discuss critical issues of the time. The performances of candidates will cast a significant influence on the result of election. Through watching the debates, the audience seek information related to topics they concern about to help them decide which candidate they will vote for. Unfortunately, this year's debates was probably the most chaotic one in the history. The polarized situation actually got worse after the debates for the audience might find themselves drown in quarrels and interruptions.

Here, we would like to propose an application that can help the audiences to have a better understanding of the narrative of both candidates by visualizing how they arrange their own arguments and counter attack each other's. However, it is difficult for traditional ontology approach to represent and reasoning based on the arguments since ontology based system only supports consistent knowledge base and the collection of arguments that come from different speakers are frequently in conflict with each other therefore inconsistent. Therefore, we leverage the abstract argumentation framework (AF) proposed by [2] to deal with the inconsistency by formalizing the debates into arguments and their attack relations. Furthermore, by applying the idea of natural deduction for structured argumentation framework (NDSA), we can not only show arguments and their relations in a graph but also how it looks like inside these arguments. That is, we can show how these arguments are formed and why they holds in term of the speakers' conviction using the natural deduction proof in propositional logic which is simple and human-understandable.

1.2 Background

1.2.1 XAI

The concepts related to explainable AI is getting more and more attention in the past few years. An iconic event of that is the implementation of EU General Data Protection Regulation (GDPR)[1] and Data Potection Act 2018 (DPA)[2] in the UK in 2018. The GDPR requires the data controller to protect the privacy of data subject and to use the data responsibly. Whatever using an automated or semi-automated system to make a decision based on personal information, the data controller must be able to provide meaningful explanation about how the personal data is processed, why it leads to such a decision and what is the consequence of that to the data subject.

With respect to the principle of these regulation, researchers divide now existing models into two categories: models that are interpretable by design and models that needs external methods to explain. With the same spirit, [3] refers the task of XAI to design transparent models and explain black-box models. [1] further details the three level of transparency and different types of post-hoc techniques.

[1] gdpr.eu
[2] www.gov.uk/data-protection

1.2.1.1 Levels of Transparency

Transparent models are called white-box models as we can have a direct understanding on their inner mechanism. In terms of their explainability, we can further put transparent models into three levels:

- **Simulatability**
  Models in this level can be simulated as a whole by human beings. Note that models with great amount of rules do not belongs to this level even if the rules are simple and explicit since it would be difficult to describe these models using text or visual means.

- **Decomposability**
  Models in this level are able to explain every part within separately but not altogether.

- **Algorithmic Transparency**
  Models in this level allows their users to understand how the model would behave under any possible circumstances. Thus, users can predict the outcome easily given the input data.

A simulatable model is always decomposable. In the same vein, a decomposable model is always algorithmic transparent.

1.2.1.2 Post-hoc Explainability Techniques

Black-box models are by design opaque and cannot explain themselves. Therefore, researchers propose various of methods to interpret their behaviour afterwards. These methods are called post-hoc techniques.

- **Text explanations** denotes methods that generates text or corresponding symbols to help users understand the results of the model.

- **Visual explanations** try to visualize the behaviour of the model using chart, graph, etc., for visual information is considered to be apprehended at a glance. This method coincides with the idea of dimension reduction in machine leaning.

- **Local explanations** stands for attempt to explain a segment or subspace of the solution space of the whole model.

- **Explanations by example** aim at finding a comprehensive example that By reading its input and output, one can get a taste of how the model behave and operate.

- **Explanations by simplification** considering building a simplified model based on the original model. To bulid a new model we can (1) focus on explainability using an interpretable model or (2) sustain the performance while reducing the computational complexity. The latter approach would create a distilled version of the original model; thus it is easier for practical use such as implementation on mobile devices.

- **Feature relevance explanations** design a relevance score. The score measures each variables' impact on the output of the model. We can then explain the model by describing features and their relation of the outcome with respect to each contribution score.

1.2.2 Nonmonotonic Reasoning

If we can derive A from a set of formulae Φ and for $\Phi \cup \Psi$, a super set of Φ, we can also derive A, then we say it is a *monotonic* logic. On the contrary, a *nonmonotonic* logic may produce a different outcome meaning A may not be derivable in the superset. To achieve non-monotonic reasoning in a system, researchers have proposed several methods such as Believe Revision, Defeasible Logic as well as Argumentation system.

Classical logic is known to be monotonic. Therefore, expert systems that employ the classical logic will require the knowledge base to be consistent. We give an informal yet intuitive example in propositional logic knowledge base if the knowledge base becomes inconsistent as follows.

Suppose a knowledge base Δ is represented by propositional logic:

- At timestamp t_0, we have $\Delta_0 = \Delta$.
- At timestamp t_1, we add a new proposition p to the knowledge base (assume that $\neg p \notin \Delta$). Now, we have $\Delta_1 = \Delta \cup \{p\}$ as well as $\Delta_1 \vdash p$.
- At timestamp t_2, we add another proposition $\neg p$. Now we have $\Delta_2 = \Delta \cup \{p, \neg p\}$. Because of the principle of explosion (i.e. the *ex falso quodlibet*), we can conclude any possible propositions from Δ_2. That is, $\Delta_2 \vdash p \wedge \neg p$, $p \wedge \neg p \vdash \bot$, $\bot \vdash q$ (q is an arbitrary proposition). Hence, any inconsistent knowledge base is untrustworthy and should not be used (Note that inconsistency, untrustworthiness

and non-monotonicness are different topics but related).

Following the formalization of NDSA (detail will be discussed in Chapter 3), we can perceive Δ_2 as a set of arguments $\{A_1, A_2, ...\}$ and their attack relation $\{(A_1, A_2), (A_2, A_1), ...\}$. Moreover, we have argument $A1$ that has claim p supported by p and $A2$ that has claim $\neg p$ supported by $\neg p$ (denoted $A_1 : \{p\} \vdash p, A_2 : \{\neg p\} \vdash \neg p$ in NDSA). The definition of NDSA is in chapter 3.
In this way, we avoid conflicts within the inconsistent knowledge base. Therefore, NDSA supports nonmonotonic features in a inconsistent propositional logic knowledge base.

1.3 Objectives

We aim to propose an argumentation based framework that supports practical reasoning. In a sense that our framework can model real life use cases with prepossessing and at the same time provide human-friendly explanations. For our system, we want to achieve at least algorithmic transparency according to section 1.2.1.1. For extra explanations, we want to apply text and visual explanation according to section 1.2.1.2.

The objectives are as follows.

1. We implement natural deduction proof for propositional logic.

2. We visualize the argumentation graph given an inconsistent propositional logic knowledge base.

3. We develop dispute trees for each argument in terms of the semantics of abstract argumentation.

4. We build a web application interface that allows users to interact with it searching for certain arguments and explanations.

5. We apply our model to a practical user case. In particular, we model a fragment of a real debate. We show that our system can reason of the debate and provide human-friendly explanations.

1.4 Contributions

We summarize the contributions of our work into the following aspects.

1. We show that we can decide whether an argument in an inconsistent knowledge base is acceptable computationally with respect to abstract argumentation semantics.

2. We show that the natural deduction proof of an argument can be re-interpreted into human-understandable explanations.

3. We show that dispute trees of an argument can be re-interpreted into human-understandable explanations.

4. We apply our system to a real life debate case demonstrating the usefulness of our system in terms of reasoning and decision making.

1.5 Thesis Structure

The reminder of this thesis is structured as follows: Chapter 2 provides the general idea of argumentation theory, its definition and semantics. We also include definitions of propositional logic as they may vary with respect to different research backgrounds and the brief introduction of corresponding reasoning methods natural deduction and tableaux. Chapter 3 introduces the formalization of NDSA framework and its idea of human-friendly explanation. Chapter 4 presents the design and structure of our proposed framework. Chapter 5 explores our use case including each component of the modeled knowledge base and its prepossessing details. We also evaluate the components of our framework with a set of unit test cases. Evaluation setting and its results will be listed in this Chapter. Chapter 6 makes a summary of our work and discusses the potential future directions.

Chapter 2

Preliminaries

2.1 Overview

Argumentation plays an essential role in the human society. [4] gives a definition of argumentation in the context of original usage and the argumentation theory:

> *Argumentation is a communicative and interactional act complex aimed at resolving a difference of opinion with the addressee by putting forward a constellation of propositions the arguer can be held accountable for to make the claim at issue accountable to a rational judge that judges reasonably.*

Indeed, we can find many materials mentioning argumentation throughout the history. In The Republic (Plato, 348b), Plato takes arguments as the key component of dialectic:

> *The dialectical method is discourse between two or more people holding different points of view about a subject, who wish to establish the truth of the matter guided by reasoned arguments.*

Leibniz also has a famous saying that possibly herald the formal and computational turn in the research field of argumentation. In The art of Discovery (Leibniz and Wilhelm, 1685), he says:

> *The only way to rectify our reasonings is make them as tangible as those of the Mathematicians, so that we can find our error at a glance, and when there are disputes among persons, we can simply say: Let us calculate, without further ado, to see who is right.*

This passage is known as Leibniz' Dream, for it conceives a reasoning method through which we can resolve dispute by computation.

Today's study of argumentation in formal and computational perspective stretches from nonmonotonic reasoning and logic programming. In 1987, 'Defeasible reasoning' [5] was proposed as an adaptation of nonmonotonic reasoning in the argumentation theory. The author Pollock's theory of defeasible reasoning based on arguments and five forms of argument defeat. How arguments defeat each other can be categorized into followings:

1. **Undermine** stands for defeat that achieved by attacks on the supports of an argument.

2. **Undercut** is a form of defeat in which the connection between supports and claim of an argument is attacked.

3. **Rebut** is a form of defeat in which an argument is defeated (rebutted) by another argument with opposite claim.

4. **Sequential weakening** is a form of defeat that can be described using sorites paradox which is a famous paradox formalized by Eubulides of Miletus in 4th century BC. It indicates a situation in which despite that the single support (premise) backs up the claim in an argument, repeatedly applying that support will eventually deny the claim, therefore, defeat the argument.

5. **Parallel strengthening** stands for a form of defeat related to the idea of accrual of reasons. The indicating situation is that an argument containing two supports defeats another argument while each support alone is not able to form an effective attack.

Note that the notation and definition of arguments in Pollock's theory is different from those in structured argumentation theory, as we will discuss later in this section.

In 1995, [2] refreshed the study of argumentation theory. Dung gives a strong yet simple mathematical based formalization of argumentation and focuses on the attack relations between arguments. In this fashion, arguments and relation can be seen as a directed graph where nodes are arguments and directed edges are attack relations. Such a graph is called an argumentation framework (see section 2.2 for detail).

In Dung's theory, each property of arguments is completely decided by their attack relations; that is, the inner structure of an argument including supports and claim as well as their interaction is out of consideration. For this reason, Dung's theory is usually called abstract argumentation theory as it abstracts from the structure of arguments.

In Dung's notation, an extension can be regarded as subsets of the set of all arguments that satisfies certain conditions. Since different extensions can be identified as different interpretation of the argumentation framework, they are often referred to as semantics. In addition to the semantics Dung proposed in his paper, researchers also proposed many new extensions that express desired properties. We will list the details in the next section.

Although it is excluded from abstract argument theory, argument with structure is always a key component of argumentation theory. We focus on the attempts that explore the relation between argumentation and classical logic. [6] proposes a formalization for arguments in terms of classical logic. They define an argument as a pair (Φ, α). Φ is a set of supports for an argument and α is the claim. The defeat condition (which is called 'attack' in Dung's theory of argumentation) is as follows:

1. **Undercut**
   An argument (Φ, α) is undercutted by another argument (Φ, β) if we can combine some formulae in Φ making them equal to the negation of β.

2. **Rebuttal**
   An argument (Φ, α) is rebutted by another argument (Φ, β) if β is equal to the negation of α.

We adapts the notation of undercut in our work as the structure of attack.

For a better understanding of the argumentation theory, we enumerate several important concepts here before we finish this section.

- **Burden of proof** is a well-known concept especially in the law context. To make it simple, we only bring up the basic into two aspects: 'Who bears the burden' and 'to what extend'. Suppose we have two parties of people conducting a discussion, one party makes a claim and another denies it. Here, the first aspect corresponds to which party is obligated to justify its claim. The second aspect corresponds to how many evidence or levels of verification do the party must provide or what constrains the party must meet in order to justify its claim.

In our work, we only refers to the first aspect.

- **Unexpressed premises**
There are many other terms such as an implicit or missing premise that describes this phenomenon depending on the background of the researchers. In real life discourse, it is frequently the case that the speaker omits or implicitly mention premises that considered as convention, fact, commonsense or something aforementioned within the context. It is called *enthymeme* that can be regarded as a rhetorical counterpart or incomplete version of syllogism.

  In our work, we find that in our use case almost all arguments the speakers make are enthymematic arguments. Therefore, we add premises accordingly in order to form logically strict syllogistic arguments.

- **Argument schemes** stands for ways that collect from the daily discourse to conduct an argument from supports to claim. These schemes can be seen as defeasible inference rules in logic as they are only conditional true. Same schemes might be studied extensively in the study of rhetoric along the history. Researchers now explore it in terms of formalization and computational application in the study of argumentation. Although their study is out of the scope of our work, we find it is worth mentioning as there is a potential reunion of dialectic and rhetoric to be found as they parted ways since ancient Greek.

2.2 Abstract Argumentation

2.2.1 Basic Concepts

In the formalism of an abstract argumentation framework (AAF or AF), we have a set of arguments and their attack relations that can be presented in a directed graph [2]. Based on the interpretation of extensions or semantics, we can evaluate the acceptance of the arguments in the frameworks whether they are accepted or not in terms of that extension or semantics using tools of topology of the graph.

Intuitively, an argument is accepted if it can defend itself to a certain degree according to the definition of the extension with or without the help of some other arguments within the set of arguments. Therefore, we can identify accepted sets of arguments where all arguments in one of these sets together defend themselves against attacks concerning the extension.

Although in [2], the argument semantics is defined using mathematical sets, called extension-based approach, researchers later proposes an alternative approach called labelling-based approach.

The labelling-based approach gives each argument one of the three label: *in*, *out*, and *undec*. The *in* label means the argument is accepted in terms of, for example, (admissible) labelling. The *out* label means the argument is rejected in terms of that labelling. The *undec* label means we remain the label of the argument as undecided because we cannot be sure whether the argument is accepted or rejected in terms of that labelling.

As it is already proven by many researchers, every labelling corresponds to an extension under the same semantics (for example, admissible labelling corresponds to admissible extension). One of them can transfer to another through a simple mapping though it is not always bijective (for example, different admissible labelling may resolve to the same extension set). Therefore, in the reminder of this thesis, we will only present the basic definitions of labelling but not definitions with reference to different semantics.

Now we introduce the first and foremost definition of the abstract argumentation framework. Here, we only consider the finite setting.

Definition 2.1: An *argumentation framework* is a pair

$AF = \langle AR, attacks \rangle$

where AR stand for a finite set of arguments, and $attacks$ stands a binary relation, $attacks \subseteq AR \times AR$.

For two arguments $A, B \in AR$, A attacks B iff $(A, B) \in attacks$.

Although not in the original definition, here we would like to add two supplemental definitions for notations that used in the following definitions.

Definition 2.2: For each $a \in AR$, A^- stands for $\{B \mid (B, A) \in attacks\}$, and a^+ stands for $\{B \mid (A, B) \in attacks\}$. Same spirit, for $\mathcal{A}rgs \subseteq AR$, $\mathcal{A}rgs^-$ stands for $\{B \mid \exists A \in \mathcal{A}rgs : (B, A) \in attacks\}$, and $\mathcal{A}rgs^+$ stands for $\{B \mid \exists A \in \mathcal{A}rgs : (A, B) \in attacks\}$.

Definition 2.3: Given an argumentation framework $AF = \langle AR, attacks \rangle$, and a set of arguments $\mathcal{A}rgs \subseteq AR$, $\mathcal{A}rgs$ defends $a \in AR$ iff $\forall B \in A^-, \exists C \in \mathcal{A}rgs : (C, B) \in attacks$. The function $F_{AF} : 2^{AR} \rightarrow 2^{AR}$ is a characteristic function of AF if $F_{AF}(\mathcal{A}rgs) = \{A \mid \mathcal{A}rgs \text{ defends } A\}$.

In fact, this idea is mentioned in definition 16 [2], while Dung uses the words 'acceptable with respect to' instead of 'defend'. We follow [4] using 'defend' as it is more intuitive from our perspective.

Let us move on to the semantics of an argumentation framework. We have definitions in both labelling-based approach and extension-based approach.

For the labelling-based approach:

Definition 2.4: Given an argumentation framework $AF = \langle AR, attacks \rangle$, a set of labels Λ, A Λ-*labelling* is a total function $\mathcal{L}ab : AR \rightarrow \Lambda$. $\mathfrak{L}(\Lambda, AF)$ stands for the set of all Λ-labellings of AF.

A label set Λ is a predefined set. In our setting, $\Lambda = \{in, out, undec\}$.

Definition 2.5: Given an argumentation framework $AF = \langle AR, attacks \rangle$, a set of labels Λ, $\mathcal{L}_\sigma(AF)$ is a *labelling-based semantics* σ with respect to AF is a subset of $\mathfrak{L}(\Lambda, AF)$.

For the extension-based approach:

Definition 2.6: $\mathcal{E}_\sigma(AF)$is an *extension-based semantics* σ with respect to $AF = \langle AR, attacks \rangle$ is a subset of 2^{AR}.

The relation between the labelling-based approach and extension-based approach is given in the following definitions:

Definition 2.7: Given an argumentation framework $AF = \langle AR, attacks \rangle$, a set of labels Λ, $\Lambda = \{in, out, undec\}$ and a Λ-labelling $\mathcal{L}ab$, the corresponding extension (set of arguments) is $Lab2Ext(\mathcal{L}ab)$, where $Lab2Ext(\mathcal{L}ab) = in(\mathcal{L}ab)$, $in(\mathcal{L}ab) = \{A \mid \mathcal{L}ab(A) = in\}$.

Definition 2.8: Given an argumentation framework $AF = \langle AR, attacks \rangle$, a set of labels Λ, $\Lambda = \{in, out, undec\}$ and a Λ-labelling $\mathcal{L}ab$, a sementics σ based on both labelling and extension, the set of extensions corresponding to $\mathcal{L}_\sigma(AF)$ is $\mathcal{E}_\sigma(AF)$, where $\mathcal{E}_\sigma(AF) = \{Lab2Ext(\mathcal{L}ab) \mid \mathcal{L}ab \in \mathcal{L}_\sigma(AF)\}$.

One last thing that is worth remarking is that if we set the set of labels Λ to $\Lambda = \{\in, \notin\}$, then the extension-based approach can be seen as a special case of the labelling-based approach. This indicates that the labelling notation is, in fact, more general.

2.2.2 Semantics

In [2], Dung first introduced four types of semantics: complete, preferred, stable and grounded. Admissibility, despite the fact that it is initially introduced as a semantic property, researchers often regard it as a type of semantics namely admissible semantics. We follow this manner in our work.

Later Dung introduces more semantics, ideal [7], along with other researchers, semi-stable [8, 9] and eager [10]. The semantics listed above are all based on admissibility, there are other semantics that not preserve this property, we leave them out for they are out of the scope of our work. However, we do include naive semantics as a supplement notation of conflict-freeness.

Example 2.1: Here we consider an argumentation framework $AF = \langle AR, attacks \rangle$ where $AR = \{A,\ B,\ C,\ D\}$, $attacks = \{(A,\ B),\ (C,\ B),\ (C,\ D), (D,\ C), (D,\ D)\}$ that is illustrated in figure 2.1. This example will serve as a common example for all following semantics.

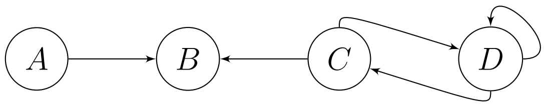


Figure 2.1: An argumentation framework

- **Conflict-freeness**

  Conflict-free is the minimal demand that any other semantics must fulfill. Intuitively, it means for arguments to coexist within a set, they should not attack each other.

  Definition 2.9: Given an argumentation framework $AF = \langle AR, attacks \rangle$, and a set of arguments $\mathcal{A}rgs \subseteq AR$, $\mathcal{A}rgs$ is *conflict-free* iff $\nexists A, B \in \mathcal{A}rgs : (A, B) \in attacks$.

  Note that this definition rules out all self-attacking arguments (consider $A = B$).

  In example Example 2.1, the conflict-free extensions are: $\emptyset$, $\{A\}$, $\{B\}$, $\{C\}$, $\{A,\ C\}$.

- **Naive semantics**

  The naive semantics ($\mathcal{NA}$) coincides with greedy algorithm as its aim is to select as many arguments as possible as long as they do not attack each other.

  Definition 2.10: Given an argumentation framework $AF = \langle AR, attacks \rangle$, and a set of arguments $\mathcal{A}rgs \subseteq AR$, $\mathcal{A}rgs$ is a *naive extension* iff $\mathcal{A}rgs$ is conflict-free and for every conflict-free set $\mathcal{B} \subseteq AR$, $\mathcal{A}rgs \not\subset \mathcal{B}$.

  Note that the naive extension is not necessarily unique (consider conflict-free sets are $\emptyset, \{A\}, \{B\}, \{C\}, \{A, B\}, \{B, C\}$, then the naive extensions are $\{A, B\}, \{B, C\}$).

  In example Example 2.1, the naive extension is: $\{A,\ C\}$.

- **Admissible semantics and admissibility**

  Admissible semantics ($\mathcal{AS}$) or admissibility describes a set of arguments that arguments within not only coexist in peace but also protect each other from attacks coming from outside.

  Definition 2.11: Given an argumentation framework $AF = \langle AR, attacks \rangle$, and a set of arguments $\mathcal{A}rgs \subseteq AR$, $\mathcal{A}rg$ is said to be an *admissible* set iff $\mathcal{A}rg$ is conflict-free and $\mathcal{A}rgs \subseteq F_{AF}(\mathcal{A}rgs)$

  We can see from the definition that the empty set is always admissible for any argumentation framework.
  In example Example 2.1, the admissible extensions are: $\emptyset$, $\{A\}$, $\{C\}$, $\{A,\ C\}$.

- **Complete semantics**

  Complete semantics ($\mathcal{CO}$) can be regarded as a strict version of admissible semantics, as a complete extension demands all arguments it protects belongs to it.

  Definition 2.12: Given an argumentation framework $AF = \langle AR, attacks \rangle$, and a set of arguments $\mathcal{A}rgs \subseteq AR$, $\mathcal{A}rgs$ is a *complete extension* iff $\mathcal{A}rgs$ is conflict-free and $\mathcal{A}rgs = F_{AF}(\mathcal{A}rgs)$.

  As evident from the definition, a complete extension is always admissible.

  In example Example 2.1, the complete extensions are: $\{A\}$, $\{A,\ C\}$.

- **Grounded semantics**

  Grounded semantics ($\mathcal{GR}$) is referred to as the most skeptical semantics. It is similar to the situation where a person wants to take a reasonable position while at the same time trying to commit as little as possible at the presence of a large amount of information containing conflicts.

  Definition 2.13: $AF = \langle AR, attacks \rangle$. A *grounded extension* is the intersection of all complete extensions in AF.

The grounded extension is unique and the inclusion-wise minimal set of the set of all complete extensions. An inclusion-wise minimal set in terms of a set of sets is a set that is not a superset of any other set within the set of sets.

In example Example 2.1, the grounded extension is: $\{A\}$.

- **Preferred semantics**

  Unlike grounded semantics, In the viewpoint of preferred semantics ($\mathcal{PR}$) we try to accept as many arguments as possible as long as they altogether are still reasonable.

  Definition 2.14: $AF = \langle AR, attacks \rangle$. A *preferred extension* is admissible and an inclusion-wise maximal set of all complete extensions in AF.

  An inclusion-wise maximal set (also called maximal set with respect to set-inclusion) in a set of sets is a set that is not a subset of any other set within the set of sets. In opposition to the grounded extension, the preferred extension is not necessarily unique.

  In example Example 2.1, the preferred extension is: $\{A,\ C\}$.

- **Ideal semantics**

  The ideal semantics can be simulated as a person tries to stay neutral yet reasonable among a group of people that follows preferred semantics. This person only accepts arguments that the group of all people agrees about together with the constraint that these arguments stay reasonable jointly. When these arguments conflicts, this person will reject one of them repeatedly until the rest are reasonable altogether. If otherwise, this person will finally choose to believe nothing.

  Definition 2.15: Given an argumentation framework $AF = \langle AR, attacks \rangle$, and a set of arguments $\mathcal{A}rgs \subseteq AR$, $\mathcal{A}rgs$ is an *ideal extension* iff $\mathcal{A}rgs$ is admissible and a subset of the intersection of all preferred extensions.

The ideal extension is also a complete extension and a superset of the grounded extension. Bear in mind that ideal extension is also unique.

In example Example 2.1, the ideal extension is: $\{A,\ C\}$.

- **Stable semantics**

  The stable semantics ($\mathcal{ST}$) stands for an autocratic idea saying "you are either with us or against us", as a set of arguments of stable semantics attacks all arguments outside.

  Definition 2.16: Given an argumentation framework $AF = \langle AR, attacks \rangle$, and a set of arguments $\mathcal{A}rgs \subseteq AR$, $\mathcal{A}rgs$ is a stable extension iff $\mathcal{A}rgs$ is conflict-free and $\forall B \in AR \backslash \mathcal{A}rgs, \exists C \in \mathcal{A}rgs : (C, B) \in attacks$.
  In example Example 2.1, the stable extension is: $\{A,\ C\}$.

- **Semi-stable semantics**

  The semi-stable semantics ($\mathcal{SST}$) can be seen as a weak version of stable semantics. A set of arguments that follows semi-stable semantics includes all arguments it protects and attacks outside arguments as much as possible.

  Definition 2.17: Given an argumentation framework $AF = \langle AR, attacks \rangle$, and a set of arguments $\mathcal{A}rgs \subseteq AR$, $\mathcal{A}rgs$ is a *semi-stable extension* iff $\mathcal{A}rgs$ is a complete extension and $\mathcal{A}rgs \cup \mathcal{A}rgs^+$ is maximal with respect to set inclusion among all complete extensions.

  An interesting connection between stable and semi-stable extension is that every stable extension is a semi-stable extension and the latter coincides with the former if the former exists.
  In example Example 2.1, the semi-stable extension is: $\{A,\ C\}$.

- **Eager semantics**

  Finally we have the most credulous unique status (Eemeren and Verheij, 2018) semantics ($\mathcal{EAG}$) in this section.

Definition 2.18: Given an argumentation framework $AF = \langle AR, attacks \rangle$, and a set of arguments $\mathcal{A}rgs \subseteq AR$, $\mathcal{A}rgs$ is the *eager extension* iff $\mathcal{A}rgs$ is admissible and a inclusion-wise maximal set of the set of all semi-stable extensions.

The eager extension is a superset of the ideal extension and it is unique if the frameworks is finite.

In example Example 2.1, the eager extension is: $\{A,\ C\}$.

In the end we illustrate the relations between semantics in figure 2.2. Note that these relations are implied by their definitions given above and guaranteed for finite frameworks.

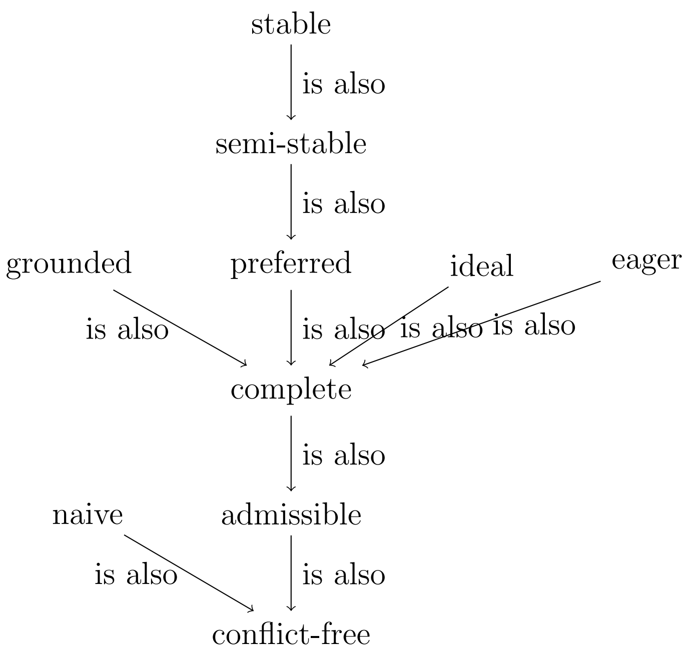


Figure 2.2: Relation among extensions

2.3 Assumption-based Argumentation

The assumption-based argumentation (ABA) [11–14] is a form of structured argumentation [15]. Like most of forms of structured argumentation, its notions of argument and attack relation are different from those in abstract argumentation (AA). Nevertheless, because of ABA's flexibility, allowing other forms of nonmonontonic reasoning to use its tools without any additional machinery.

In this section, we will not go into detail about ABA for it is a sophisticated formalism that beyond our scope. In particular, we will present only the basic concepts but not the semantics to show that (flat) ABA can in fact translate to AA and vice versa.

Definition 2.19: An *ABA framework* is a tuple $\langle \mathcal{L}, \mathcal{R}, \mathcal{A}, \overline{} \rangle$ where

- $\mathcal{L}$ is a set of sentences called a language.
- $\mathcal{R}$ is a set of inference rules. Each rule has a head and a body. A head is a sentence in $\mathcal{L}$. A body consists $m \geq 0$ sentences in $\mathcal{L}$.
- $\langle \mathcal{L}, \mathcal{R} \rangle$ is a deductive system.
- $\mathcal{A} \subseteq \mathcal{L}$ is a (non-empty) set.
- A element $a \in \mathcal{A}$ is called a assumption.
- $\overline{}$ is a total mapping from $\mathcal{A}$ into $\mathcal{L}$. $\overline{a}$ is referred to as the contrary of $a \in \mathcal{A}$.

A rule in $\mathcal{R}$ with head σ_0 and body $\sigma_1, ..., \sigma_m$ can be written as

$$\sigma_0 \leftarrow \sigma_1, ..., \sigma_m$$

Definition 2.20: $S \subseteq \mathcal{L}$ is a set of sentences. Its closure is:

$$Cl(S) = \{\sigma \in \mathcal{A} \mid \exists S' \vdash^R \sigma, S' \subseteq S, R \subseteq \mathcal{R}\},$$

where $S' \vdash^R \sigma$ is a deduction for $\sigma \in \mathcal{A}$, supported by $S' \subseteq S$ and $R \subseteq \mathcal{R}$.

$A \subseteq \mathcal{A}$ denores a set of assumptions. It is *closed* iff $A = Cl(A)$.

An ABA framework is referred to as *flat* if its set of assumptions is guaranteed to be closed.

Definition 2.21: An ABA framework $\langle \mathcal{L}, \mathcal{R}, \mathcal{A}, \overline{} \rangle$ is *flat* iff $\forall A \subseteq \mathcal{A}$: A is closed.

As observed from [7,16], flat ABA frameworks are instances of AA frameworks where arguments are deductions supported by sets of assumptions and attacks are defined by leveraging the transform from attack between sets of assumptions to attack between arguments properly.

Definition 2.22: $\mathcal{ABA} = \langle \mathcal{L}, \mathcal{R}, \mathcal{A}, \overline{} \rangle$ is a flat ABA framework.

- $A \vdash_{arg} \sigma$ is an argument for $\sigma \in \mathcal{L}$ supported by $A \subseteq \mathcal{A}$ and $R \subseteq \mathcal{R}$. $A \vdash^{R} \sigma$ is a corresponding deduction.
- An argument $A \vdash_{arg} \sigma$ attacks another argument $B \vdash_{arg} \tau$ iff $\exists b \in B$: $\sigma = \overline{b}$.

Then we have $\mathcal{AA}(\mathcal{ABA}) = (AR, attacks)$ is the corresponding AA framework for $\mathcal{AA}$ where AR is the set of all arguments and *attacks* is the set of all pairs (A, B) for $A, B \in AR : A$ *attacks* B.

Theorem 2.1. $\mathcal{ABA} = \langle \mathcal{L}, \mathcal{R}, \mathcal{A}, \overline{} \rangle$ is a flat ABA framework and $\mathcal{AA}(\mathcal{ABA})$ is its corresponding AA framework.

1. If a set of assumptions $A \subseteq \mathcal{A}$ in $\mathcal{ABA}$ is $\spadesuit$, then the set of arguments supported by any $A' \subseteq A$ in $\mathcal{AA}(\mathcal{ABA})$ is $\spadesuit$.
2. The set of all sets of assumptions in $\mathcal{ABA}$ is $\spadesuit$ if it supports a set of arguments in $\mathcal{AA}(\mathcal{ABA})$ that is $\spadesuit$.

$\spadesuit \in \{admissible, grounded, ideal, complete, preferred, stable\}$.

Proof. The proof for admissible, grounded, ideal extensions can be found in the proof of Theorem 2.2 in [7]. The proof for complete, preferred extensions can be found in the proof of Theorem 6.1 and 6.3 respectively in [17]. The proof for stable extensions can be found in the proof of Theorem 1 in [16]. □

This theorem implies that the semantics of flat ABA frameworks can be defined in terms of extensions as sets of arguments. Thus, it is possible to compute extensions of flat ABA frameworks using the existing machinery for computing extensions of AA frameworks.

Conversely, as shown in [16], AA frameworks are instances of flat ABA frameworks.

Definition 2.23: $\mathcal{AA} = (AR, attacks)$. The corresponding ABA framework of $\mathcal{AA}$ is $\mathcal{ABA}(\mathcal{AA}) = \langle \mathcal{L}, \mathcal{R}, \mathcal{A}, \overline{} \rangle$ with

- $\mathcal{A} = AR$.
- $\forall A \in \mathcal{A} : \overline{A} = A^c$.
- $\mathcal{L} = A \cup \{A^c \mid A \in \mathcal{A}\}$.
- $\mathcal{R} = \{B^c \leftarrow A \mid (A, B) \in attacks\}$

We can observe that there is a clear one-to-one correspondence between all semantics of AA and ABA frameworks since the set of all arguments in AA frameworks coincides the set of all assumption in ABA frameworks.

Theorem 2.2. $\mathcal{AA} = (AR, attacks)$. $\mathcal{ABA}(\mathcal{AA})$ is the corresponding ABA framework of $\mathcal{AA}$.

1. If $A \subseteq AR$ is $\spadesuit$ in AA, then A is $\spadesuit$ in $\mathcal{ABA}(\mathcal{AA})$.
2. If $A \subseteq \mathcal{A}$ is $\spadesuit$ in $ABA(AA)$, then A is $\spadesuit$ in $\mathcal{AA}$.

$\spadesuit \in \{admissible, grounded, ideal, complete, preferred, stable\}$.

Proof. The proof can be found in Theorem 2 [16]. □

This theorem suggests that existing machinery in flat ABA frameworks can be used in AA frameworks as well. In deed, dispute trees that we use in our work is a tool adapted from flat ABA frameworks. The feature of dispute trees will be demonstrated in the next section.

2.4 Dispute Trees

Dispute trees [7, 18] provides a way to determine whether an argument in AA frameworks belongs to certain extensions. Although dispute trees is originally designed for ABA frameworks, as shown in Theorem 2.2, it can be utilized for AA frameworks.

For any abstract argumentation framework, dispute trees can be defined abstractly as follows:

Definition 2.24: $(AR, attacks)$ is an arbitrary abstract argumentation framework. A *dispute tree* for $A \in AR$ is a tree $\mathcal{T}$ that constructed following the instruction below:

1. Every node of $\mathcal{T}$ is of the form $[L : X]$, where $L \in \{P, O\}, X \in AR$. P stands for proponent and O for opponent. A node in $\mathcal{T}$ is either a proponent (P) or a opponent (O).

2. The root node of $\mathcal{T}$ is always labelled as $[P : A]$.

3. For every node labelled by $[P : B]$ with $B \in AR$, and for every $C \in AR$ with $(C, B) \in attacks$, there exists a child labelled by $[O : C]$ for node $[P : B]$.

4. For every node labelled by $[O : B]$ with $B \in AR$, there exist one child labelled by $[P : C]$ with $C \in AR : (C, B) \in attacks$.

5. There are no other nodes in $\mathcal{T}$ except those given in the instruction above.

All arguments $D \in AR$ that have nodes labeled by$[P : D]$ together form the defence set of a dispute tree $\mathcal{T}$, denoted by $\mathcal{D}(\mathcal{T})$.

If an argument that belongs to any abstract argumentation framework has a dispute tree that fulfills specific requirements, it is a member of certain extensions of that framework with respect to the requirements. The requirements are as follows:

Definition 2.25: $(AR, attacks)$ is an arbitrary abstract argumentation framework. A dispute tree $\mathcal{T}$ for $A \in AR$ is:

- **admissible** iff $\nexists B \in AR : [P : B], [O : B] \in \mathcal{T}$.

- **grounded** iff it is finite.

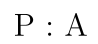
- **ideal** iff $\nexists B \in AR, [O : B] \in \mathcal{T}$: there exists an admissible dispute tree for B.

Here we reuse example Example 2.1 to build dispute trees accordingly in figure 2.3. Note that the dispute tree for B in figure 2.3b is in fact not a dispute tree since $O : A$ is not answered by a P.

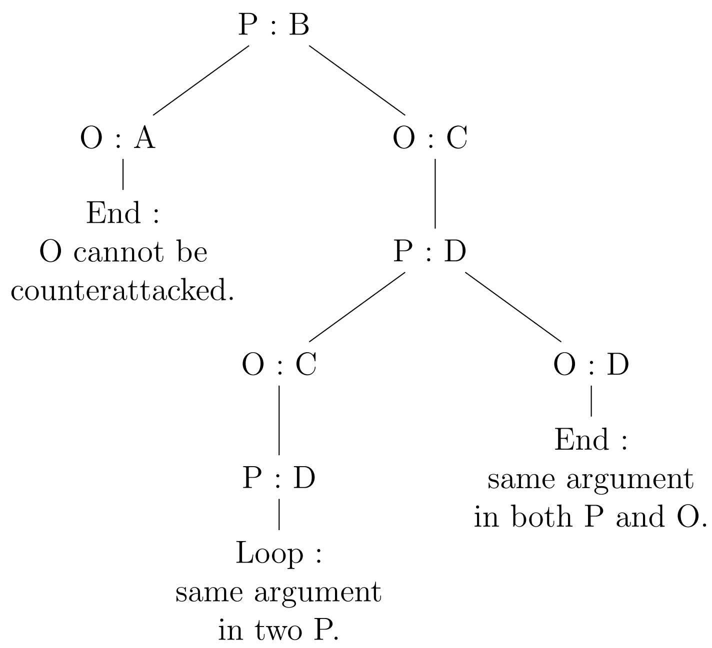
A dispute tree can be regarded as an imitation of a debate between a proponent tries to defend its argument and an opponent tries to defeat it. The debate begins with an argument proposed by the proponent. The opponent attempts to attack from every possible angle by offering all arguments that counterattack the proponent's argument. The proponent therefore must defend its argument by attack all arguments the opponent brings up in return.

What is remarkable is that dispute trees presumes that the proponent will be able to defend itself eventually. Even if a dispute tree goes infinite, it is to the advantage of the proponent. As long as the opponent cannot defeat the proponent with a knock out, meaning a draw is not enough, the result of the debate is always in the proponent's favor. In other words, it is the opponent that bears the burden of proof. If the opponent does defeat the proponent by rising an argument that the proponent cannot counterattack, the dispute tree will not be established, as shown in the example above.

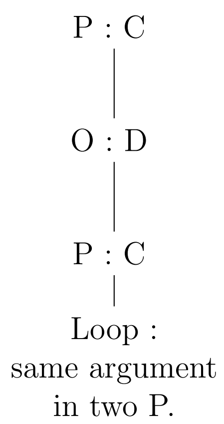
We find dispute trees can be good explanations for real debate as a dispute tree visualizes a tree that starts from an argument given by the proponent at the top (the root) following the opponent and the proponent counterattack each other taking turns to the bottom (leaves) is close to the flow of a debate, as noted in [19]. This idea will be further explored in the next chapter.



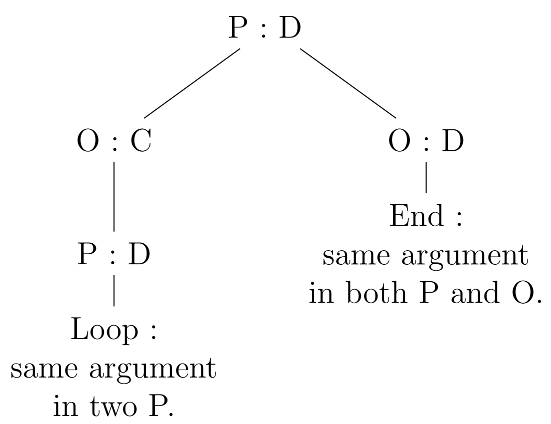
(a) **Grounded and ideal dispute tree for A**



(b) **No dispute tree for B**



(c) **Ideal dispute tree for C**



(d) **Not admissible dispute tree for D**

Figure 2.3: Dispute trees for example Example 2.1

2.5 Natural Deduction For Propositional Logic

2.5.1 Basic Concepts

The history of propositional logic can be traced back to antiquity. Its idea has been suggested by early philosophers and met its very first formalization as Stoic logic by Chrysippus in 3rd century BC which is later extended by successor Stoics. Nevertheless, propositional logic was in fact reinvented by Peter Abelard in the 12th century for at that time most of the original writings of Stoic developed propositional logic were lost and people were not able to interpret the rest. Later propositional logic was refined by symbolic logic. Finally, with the invention of truth table along with natural deduction and truth trees, it became almost like the one we are familiar with today.

On the other hand, natural deduction [20] was invented by Gentzen in 1935, with the idea to propose a logical reasoning method that expressed by inference rules that close to 'natural' human reasoning in mind. Because natural deduction is considered more human-friendly in terms of intelligibility, it is not only a calculus for logical reasoning but also widely used for educational purpose helping students learning logic.

In this section we will introduce the definition of propositional logic with respect to natural deduction as follows:

Definition 2.26: A *propositional logic with respect to natural deduction* is a formal system $\mathcal{L} = \mathcal{L}(A, \Omega, Z, I)$, where A, Ω, Z, I are defined as follows:

- A is a countably infinite set of element called *proposition symbols.*
  $A = \{p_0, p_1, p_2, ...\}$.

- Ω is a finite set of elements called *connectives.*
  $\Omega = \{\wedge, \vee, \rightarrow, \neg, \leftrightarrow, \bot\}$.

- Z is a finite set of transformation rules called *inference rules.*

- I is a countable set of initial points that called *axioms.*
  $I = \emptyset$.

The connectives carry traditional names:

$$\begin{array}{ll} \wedge - and & conjunction \\ \vee - or & disjunction \\ \rightarrow - if..., then... & implication \\ \neg - not & negation \end{array}$$

$$\begin{array}{ll} \leftrightarrow - iff & equivalence, bi\text{-}implication \\ \bot - falsity & falsum, absurdum \end{array}$$

In contrast to Hilbert-style systems who try to use axiom as much as possible, natural deduction system do not use axiom at all.

Although $\{\rightarrow, \bot\}$ or $\{\neg, \wedge\}$ is computationally complete set already, we include more connectives for two reasons: (1) We can extend Ω by more inference rules. (2) We can therefore have more derived rules as well to simplify the natural deduction proof in our implementation.

We have 11 inference rules [21] that classify into three categories: introduction rules, elimination rules and rules for $\bot$. An inference rules can be written as:

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

where the propositions above the line are *premises* and the one below the line is the *conclusion*.

Introduction rules are rules that introduce a connective in the conclusion:

$$\frac{\varphi \quad \psi}{\varphi \wedge \psi} (\wedge I) \qquad \frac{\begin{matrix}[\varphi] \\ \vdots \\ \psi\end{matrix}}{\varphi \rightarrow \psi} (\rightarrow I)$$

$$\frac{\varphi}{\varphi \vee \psi} (\vee I) \quad \frac{\psi}{\varphi \vee \psi} (\vee I)$$

And *elimination rules* are rules that eliminate a connective in the premises:

$$\frac{\varphi \quad \neg\varphi}{\bot}\,(\neg E) \qquad \frac{\varphi \wedge \psi}{\varphi}\,(\wedge E) \qquad \frac{\varphi \wedge \psi}{\psi}\,(\wedge E)$$

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

Finally we have rules for $\bot$ that eliminate $\bot$ and introduce a formula:

$$\frac{\bot}{\varphi}\,(\bot) \qquad \frac{\begin{array}{c}[\neg\varphi]\\ \vdots \\ \bot\end{array}}{\varphi}\,(RAA)$$

By applying these rules in sequence appropriately, we will get a derivation. We write $\Gamma \vdash \varphi$ if there is a derivation from Γ to φ. Such a derivation is regarded as a proof.

It is clear that this natural deduction system is sound and complete, as shown in the following theorem given by [21]:

Theorem 2.3 (Soundness)**.** Given a set of propositions Γ and a proposition φ, $\Gamma \vdash \varphi \Rightarrow \Gamma \models \varphi$.

Proof. The proof can be found in lemma 1.5.1 [21]. □

Theorem 2.4 (Completeness)**.** Given a set of propositions Γ and a proposition φ, $\Gamma \models \varphi \Rightarrow \Gamma \vdash \varphi$.

Proof. The proof can be found in theorem 1.5.13 [21]. □

The presentation we used above to present inference rules is called *tree-like representation*. While it distinguishes premises and conclusions evidently, tree-like representation is hard to illustrate complex natural deduction proofs for with the increase in the number of propositions, the tree extends both vertically and horizontally. Hence we adapt a sequential presentation, Fitch notation, in our implementation.

2.5.2 Fitch Notation

Fitch notation [22] is a notational system for the construction of formal proofs. Fitch-style proofs arrange the sentences of a proof sequences into rows.

In a Fitch-style proof, each row is either: 1. an assumption or subproof assumption; and 2. a sentence and the citation of its corresponding inference rule and the serial number or numbers of the prior line or lines in the proof that license that inference rule.

The degree of indentation of each row conveys which assumptions are active for that row in the proof. Each time a new assumption is introduced, the level of indentation increases, resulting in an additional vertical bar in the subsequent lines until the current assumption is discharged.

Example 2.2: Here we give an example of proof in Fitch notation. Assume we have $\{a \to b\} \models (b \to c) \to (a \to c)$, then the Fitch-style proof of it in our implementation is as figure 2.4

```
1.| Premise a>b
2.| | IfI Assume b>c
3.| | | IfI Assume a
4.| | | b—IfE 3,1
5.| | | c—IfE 4,2
6.| | Therefore a>c —IfI 3,5
7.| Therefore (b>c)>(a>c) —IfI 2,6

a>b |- (b>c)>(a>c)
```

Figure 2.4: Fitch-style proof example

Note that for implementation propose, we use '>' instead of '→' for the implication connective in our application. The detail will be discussed in chapter 4.

2.6 Tableaux

We have shown that the nature deduction system is sound and complete in the former section. In practice, however, its soundness is hard to achieve. To be specific, given a set of propositions Γ and the corresponding conclusion φ that we do not know if it holds in advance, the system will easily fall into a loop without telling whether there exists a derivation from Γ to φ or not. Thus, researchers usually achieve the soundness with the help of truth table. Nevertheless, the truth table itself is not an efficient method to evaluate the truth of a formula, either. Therefore, we seek a feasible way to prove if a formula is true or false, the technique we found is *tableaux*.

Analytic tableaux [23] is a schematic method that replaces logical reasoning with syntactic manipulations. In particular, tableaux replaces a statement '$a\ is\ true$' with a signed formula 'Ta', and a statement '$a\ is\ false$' with a signed formula 'Fa'. Additionally, it replaces the reasoning by inference rules with application of syntactic rules. Each connective has two syntactic rules, one signed 'T' and another signed 'F':

1.
$$\frac{T\neg a}{Fa}\ (\alpha) \qquad \frac{F\neg a}{Ta}\ (\alpha)$$

2.
$$\frac{T(a \wedge b)}{Ta, Tb}\ (\alpha) \qquad \frac{F(a \wedge b)}{Fa \mid Fb}\ (\beta)$$

3.
$$\frac{T(a \vee b)}{Ta \mid Tb}\ (\beta) \qquad \frac{F(a \vee b)}{Fa, Fb}\ (\alpha)$$

4.
$$\frac{T(a \to b)}{Fa \mid Fb}\ (\beta) \qquad \frac{F(a \to b)}{Ta, Fb}\ (\alpha)$$

Type α is called *conjunctive* type, for formulas that result in direct consequences. Type β is called *disjunctive* type, for formulas that result in branching. In a type α formula, we denote the signed formula above the line as 'α', the signed formula(s) below the line as 'α_1 (and α_2)', respectively. Likewise, In a type β formula, we denote the signed formula above the line as 'β', the signed formulas below the line as 'β_1 and β_2', respectively. The meaning of these terms is describe in the following.

When we try to prove a formula, naming determining whether there is a derivation from the premises to the conclusion, instead of proving it to be true, tableaux assumes that formula to be false and attempts to derive a contradiction from this assumption which is similar to indirect proof.

In a tableau tree-like proof:

1. We start from the given formula with the sign 'F' as the root node, meaning the formula is assumed to be false.
2. When we derive signed formula(s) from a signed formula, we write them below it as its only child. The child is called *direct consequences* of its parent node.
3. On the other hand, when we get two derived signed formulas that either one of them may holds, meaning they needs to be investigated separately, we divide them into two branches, in other words, two children.
4. We keep establishing the tree until we found a formula is signed by both 'F' and 'T' in one branch. This signifies a contradiction in that branch. We therefore close that branch by marking a '$\times$' as its leaf.
5. If every leaf in the proof tree is a '$\times$', it means that the assumption that the given formula is false is false definitely, so the given formula must be true.
6. On the contrary, if we find a leaf in the proof tree is not '$\times$' but a signed formula that cannot be decomposed anymore, we have to admit that the assumption is true and the given formula must be false.

An example of a tableau proof is in figure 2.6a.

Now that we demonstrate how a tableau proof looks like, we present a formal definition of tableaux method.

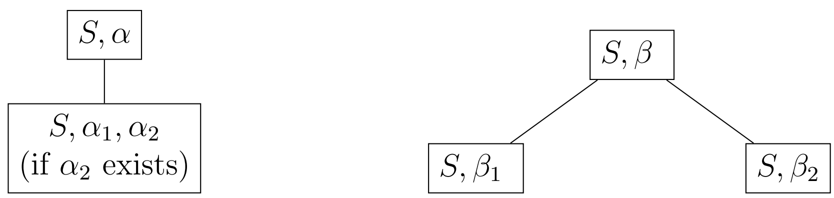
Definition 2.27: An *analytic tableau* $\mathcal{T}$ for an unsigned formula a is a dyadic ordered tree that:

1. The nodes of $\mathcal{T}$ are signed formulas.
2. The root of $\mathcal{T}$ is Fa.
3. If there is some α in the path P_b of a node b, then the node b has one successor α_1 (and α_2).
4. If there is some β in the path P_b of a node b, then the node b has two successor β_1 and β_2.
5. A branch $\mathcal{B}$ of $\mathcal{T}$ is closed if a signed formula and its conjugate (negation) are in it. if all branches of $\mathcal{T}$ are closed, then $\mathcal{T}$ is *closed* .
6. A closed tableau for Fa is a proof for a.

The presentation shown in figure 2.6a has a disadvantage that we have to track along the path of a node to find a decomposable formula. Researchers want to redesign the calculus to support local reasoning that can operate on a set of signed formulas instead of a single one. The new designed calculus is called *block tableaux* that defined as follows:

Definition 2.28: A *block tableau* proof tree satisfies following conditions:

1. The root of a tableau tree is a finite set of formulas S.
2. Operating rules for a particular formula are shown in figure 2.5



(a) **For α-rules** (b) **For β-rules**

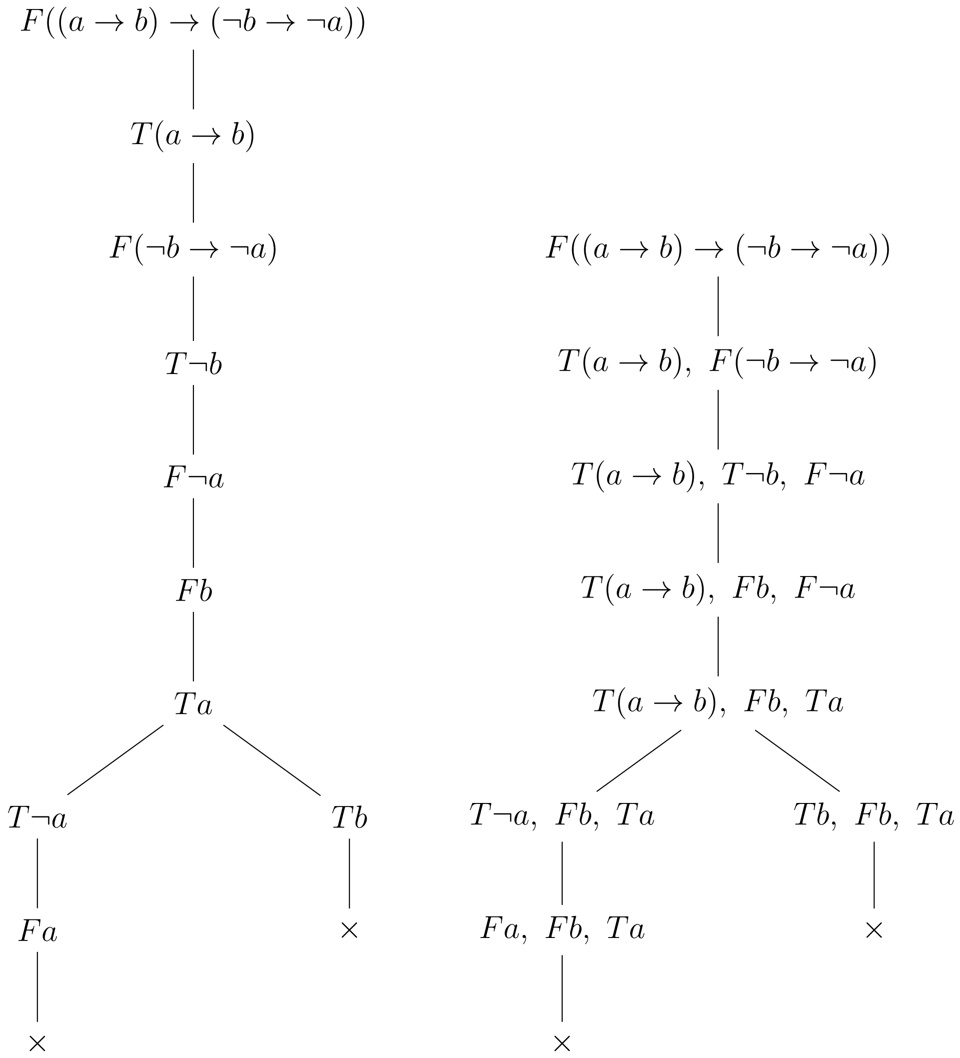
Figure 2.5: Block tableaux formats

Note that comma stands for set union. Therefore, α and β can be element of S.

3. If all branches are closed because a node has a signed formula and its conjugate (negation), then a block tableau is *closed* .

An example of a block tableau proof is in figure 2.6b.

Since every analytic tableau can be simulated by a block tableau for same rules is applied to same formulas, block tableaux is complete as analytic tableaux is complete.



(a) **Analytic tableaux example** (b) **Block tableaux example**

Figure 2.6: Tableaux example

Chapter 3

NDSA Framework

3.1 Overview

In this chapter we introduce natural deduction-based structured argumentation (NDSA) framework [19]. It is a approach that adapt argumentative semantics in abstract argumentation framework and the structure of natural deduction to provide a reasoning method with an propositional logic-based knowledge base. We find it suitable for our implementation because not only it is a system that can reasoning in a inconsistent knowledge base, but also it can offer ways of human-friendly explanations to its users.

We will first give the definitions of NDSA framework while its related semantics can be found in section 2.2.2, then we delineate the idea of different levels of explanation in NDSA.

3.2 Concepts

The author of NDSA observes that a derivation for a formula in natural deduction can somehow be regarded as a explanation of why that formula holds. This argue is acceptable because it is often the case that a teacher teach logic to students for the first time with the help of natural deduction. The author therefore develops the structure of argumentation based on natural deduction.

Definition 3.1: Let Δ be a knowledge base written in propositional logic, Φ be a set of formulae, $\Phi \subseteq \Delta$, α be a propositional formula (atomic or compound).

$\langle \Phi, \alpha \rangle$ is an *ND argument* in which claim α supported by Φ is a natural deduction proof tree such that we can derive α but not $\neg\alpha$ from Φ.

Set Φ is called supports while α is called a claim. Note that the author of NDSA imposes a consistency constraint to avoid the construction of illogical argument, for example, through *ex falso quodlibet*.

If we find two arguments that have the same claim, we can say one is more general than the other if they meet the condition in the following definition:

Definition 3.2: Given two arguments $\langle\Phi, \alpha\rangle$ and $\langle\Psi, \beta\rangle$, $\langle\Phi, \alpha\rangle$ is *comparable to and more concise* than $\langle\Psi, \beta\rangle$ iff $\alpha = \beta$ and $\Phi \subset \Psi$.

Argumentation based system allows the reasoning in a knowledge base that contains propositions and (some of) their complements. Arguments that constructed from such a knowledge base will inevitably conflicts with each other. The author of NDSA follows the notion of classical direct undercut [6] to give a definition of attack relation as follows:

Definition 3.3: Let $A = \langle\Phi, \alpha\rangle$, $B = \langle\Psi, \beta\rangle$ be two arguments. Argument A *attacks* argument B iff $\exists\psi \in \Psi : \alpha = \neg\psi$.

Here we find something worth considering. When an argument is attacked, its less concise versions will be attacked as well. Besides, we may end up making infinite many of arguments by applying or-introduction rule to the claim. In regard of these matters, the author borrows the concept of core from [24].

Definition 3.4: Arguments $A = \langle\Phi, \alpha\rangle$ and $B = \langle\Psi, \beta\rangle$ are *structurally equivalent* iff $\Phi = \Psi$ and $\alpha = \beta$.

Definition 3.5: Let $AF = \langle AR, attacks\rangle$ and $AF' = \langle AR', attacks'\rangle$ be two different AA frameworks. AF' is a *core* of AF iff:

- $AR', attacks'$ are finite.
- $\forall A \in AR\ \exists A' \in AR'$: A and A' are structurally equivalent and A' satisfies following conditions:
  - $\forall B \in AR, (B, A) \in attacks\ \exists B' \in AR' : (B', A') \in attacks'$.
  - $\forall B \in AR, (A, B) \in attacks\ \exists B' \in AR' : (A', B') \in attacks'$.

In practice, we only choose the most concise arguments omitting their less

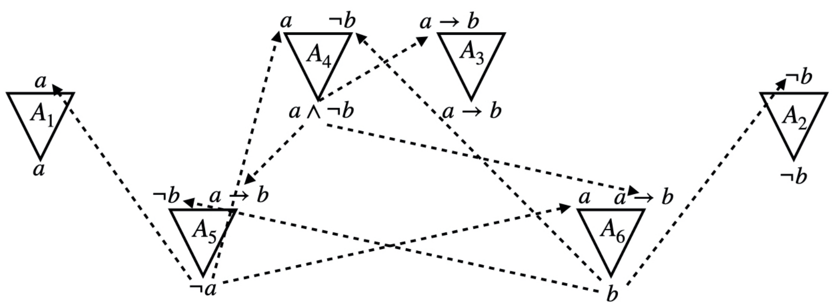


Figure 3.1: A core of an NDSA instantiated abstract argumentation framework.

concise versions and exactly one argument among all structurally equivalent arguments.

Now that we have the definition for argument and attack relation, let us define NDSA frameworks precisely.

Definition 3.6: (NDSA). A *NDSA framework* is a triple $\langle \mathcal{L}, \Delta, \vdash_{ND} \rangle$ where:

- $\mathcal{L}$ is a language in propositional logic.
- Δ is a knowledge base based on $\mathcal{L}$.
- $\vdash_{ND}$ is a derivation represented by natural deduction proof.

Proposition 3.1: Let $AF = \langle AR, attacks \rangle$ and $AF' = \langle AR', attacks' \rangle$ be two different AA frameworks built according to NDSA. AF' is a core of AF if for each set of structurally equivalent arguments $\mathcal{A} \subseteq AR$, only one of them denoted argument A is non-deterministically included in AR' and all attacks relation containing arguments in $AR \backslash A$ are excluded from *attacks* to yield $attacks'$.

Proof. The proof can be found in proposition 3.1 [19]. □

An example of a NDSA framework is in figure 3.1 as given in [19].

Since NDSA frameworks instantiates AA frameworks from propositional logic based knowledge bases, all semantics in AA also stand in NDSA. Thus, we can present the definition of consequences which is a common approach in logic based argumentation theories to denote the set of claims in accepted arguments of a concerned extension with respect to NDSA as follows:

Definition 3.7: $ext(AF)$ is an AA extension of AF built with respect to $\langle \mathcal{L}, \Delta, \vdash_{ND} \rangle$. $Con_{ext(AF)}$ is a set of consequences that built upon Δ. $Con_{ext(AF)} = \{\alpha \mid \langle \Phi, \alpha \rangle \in ext(AF)\}$.

Proposition 3.2: Let $AF = \langle AR, attacks \rangle$ and $AF' = \langle AR', attacks' \rangle$ be two AA frameworks built according to NDSA. If AF' is a core of AF, the the following holds for any concerned extension:

- $\forall \alpha' \in Con_{ext(AF')}, \; \exists \alpha \in Con_{ext(AF)} : \alpha' = \alpha$.
- $\forall \alpha \in Con_{ext(AF),} \; \exists \alpha' \in Con_{ext(AF')} : \alpha = \alpha'$.

Proof. The proof can be found in proposition 4.1 [19]. □

The author of NDSA further proposes an algorithm to calculate accepted arguments given a extension in a NDSA framework.

Algorithm 3.1 Finding accepted arguments in a NDSA framework $\langle \mathcal{L}, \Delta, \vdash_{ND} \rangle$

```
input: a knowledge base Δ, an AA semantics s
output: the set of sets of acceptable arguments with respect to s
function AcceptedArguments(Δ, s)
    Let G be an empty directed graph.
    Construct an AA framework from Δ based on the definitions of NDSA. Assign it to G.
    Find a core of G based on the definition of core.
    Remove irrelevant arguments (nodes) and attack relations (edges) in G that are not part of the core .
    calculate exts, the set of sets of acceptable arguments in the core of G with respect to the semantics s.
    return exts
end function
```

One thing that is worth mentioning in NDSA is that it provides propositional logic-based knowledge base with nonmonotonic behavior as it separates the reasoning into two level: a level of reasoning in a set of arguments according to AA semantics and a level of reasoning within each argument according to natural deduction in propositional logic. This feature further offers the capability for NDSA frameworks to illustrate two forms of explanations to its user.

3.3 Two-level Interpretation

The author of NDSA refers the two levels to macro-scope and micro-scope, where macro-scope explanations corresponds to dialogical explanation and micro-scope explanation corresponds to logical explanation.

3.3.1 Dialogical Explanations

The dialogical explanation corresponds to macro-scope interpretation for it can be regarded as a simulation of a debate where two agents, proponent and opponent, interchange their opinions through arguments.

This level of explanation is done by reinterpreting a dispute tree of an argument as a debate beginning with that argument. Whether a proponent can win with that argument depends on the acceptance with respect to a concerned extension. Therefore, we can regard the dispute tree as a dialogical simulation of possible outcomes of a real debate where the win condition is depended on correlated semantics.

The definition of dispute trees is already given in section 2.4. We can explain the dispute tree in figure 2.3d as follows:

1. Imagine we have a debate between a proponent and an opponent.
2. The debate starts with the proponent giving an argument D.
3. The opponent then can counterattack proponent's argument with either C or D.
4. Therefore the proponent has to prepare for defending itself from either attack.
5. For the attack with argument C, the proponent can answer it with argument D. The proponent then notice that in this way, the debate

will go infinite as the opponent can always counterattack D with C and the proponent can in turns counterattack C with D. Because it is the opponent that must outperform the proponent, the proponent will be able to win in the end.

6. For the attack with argument D however, the proponent find itself been attacked by its own argument. This will results in an cancellation of argument D that it is no longer admissible.
7. In the end, we get a not admissible dispute tree for argument D implying that the proponent with argument D will lose this debate in terms of a setting in admissible semantics.

3.3.2 Logical Explanations

The logical explanation corresponds to micro-scope interpretation for it illustrates why each argument holds.

This level of explanation is done by reinterpreting the natural deduction proof from the set of premises (supports) to the conclusion (claim) of an argument. When tracing a natural deduction tree from top (premises) to the bottom (conclusion), we find that it is close to the actual reasoning procedure of a human being that tries to make sense of an argument with the help of inference rules.

We can explain the dispute tree in figure 2.4 as follows:

1. We already know a implies b.
2. If we assume b implies c.
3. If we further assume a holds.
4. We find that b holds given a holds and a implies b.
5. We find that c holds given b holds and b implies c.
6. Therefore we have a implies c given that when we assume a we can get c.
7. Therefore we have if b implies c then a implies c given that when we assume b implies c we can get that a implies c.

Chapter 4

System Design

4.1 Overview

We define our work as a system for practical reasoning that is base on natural deduction-base structured argumentation (NDSA) framework.

Our system is written in Python using following packages:

- **dash** is for web interface.
- **plotly** is for interactive graphs.
- **networkx, igraph** are for graphs.
- **ast** is for storing and reading class objects.
- **HtmlTestRunner** is for unit test.
- **numpy, pandas** are for data frames.
- **flask** is for creating a web server.

The structure of our system is:

- argument engine
  - natural deduction prover
  - tableaux prover
  - argumentation reasoner
- interface
- knowledge base

The workflow of our system is demonstrated in the following algorithm 4.1.

Algorithm 4.1 NDSA Framwork Visualization

input: a prepossessed propositional logic-based knowledge base Δ and a concerned claim c, $c \in \Delta$
output: an argumentation relation graph $\mathcal{G}$, a dispute tree $\mathcal{T}$ of a chosen argument A, a natural deduction proof $\mathcal{P}$ in Fitch notation of chosen argument A, a natural language form proof $\mathcal{P}'$ of $\mathcal{P}$
function VISUALIZATION(Δ, c)
    Develop arguments that has claim c as well as their attacker arguments using propositions in Δ. Build arguments that attack at least of the existing arguments until no more argument can be built. This step is done by argumentation reasoner.
    Find sets of accepted arguments according to admissible, grounded and ideal extensions using argumentation reasoner.
    Draw a directed graph $\mathcal{G}$ where nodes are arguments and edges are attack relations that start from the attacker and end at the attacked coloring nodes with respect to the acceptance of each arguments.
    yield $\mathcal{G}$
    if a concerned argument A is chosen **then**
        Draw a dispute tree $\mathcal{T}$ of A using argumentation reasoner.
        Calculate a natural deduction proof $\mathcal{P}$ in Fitch notation where the premises are the supports and conclusion is the claim of A using natural deduction prover.
        Transform $\mathcal{P}$ into $\mathcal{P}'$ written in natural language using a subfunction of the interface.
        return $\mathcal{T}$, $\mathcal{P}$, $\mathcal{P}'$
    end if
    System ends when it is closed by the users.
end function

Each component of our system will be detailed in the following section of this chapter. The prepossessing method of the knowledge base is discussed in section 5.1 in the next chapter as it is related to the use case.

4.2 Natural Deduction Prover

Our natural deduction prover is built according to the definitions in section 2.5. But for implementational purpose, we modify some parts of it:

1. For an easier input from a keyboard as well as an easier representation

on the web pages, we change the notation of propositions and connectives to symbols that are within ASCII code. In particular, we limit the notation for propositions to lowercase roman alphabets from 'a' to 'z'. Moreover, we change '$\wedge, \rightarrow, \vee, \neg, \bot$' to '&, >, |, ~, F', respectively. We do not use '$\leftrightarrow$ since it is not exist in a derivation.

2. For a simple and concise proof representation, we use derived rules along with the original reference rules.

All rules we used in our prover is listed below.

- The notation of inference rules used in our system is slightly different from the original:
  - 'AndI/ IfI/ OrI/ AndE/ IfE/ OrE/ RAA' stands for '$\wedge I, \rightarrow I, \vee I, \wedge E, \rightarrow E, \vee E, RAA$', respectively.
  - For 'IfI' rule, we use it in a way that begins with 'IfI Assume' and ends with 'IfI'.
  - For 'RAA' rule, we use it in a way that begins with 'RAA Assume' and ends with 'ExFalsoQuodlibet'.
  - For '$\neg E$' rule and '$\bot$' rule, we only use them implicitly within 'RAA' rule for the demand of consistency in the premises, as discussed in Definition 3.1.
- Derived rules we used in our system is as follows:
  - De morgan's law is treated as 'DeMorgan' rule. It contains four rules of two types:

```
~(p|q) |- ~p&~q          ~(p&q) |- ~p|~q
~p&~q |- ~(p|q)          ~p|~q |- ~(p&q)
```

  - Double negation law is regarded as 'DoubleNegation' rule:

```
p |- ~~p          ~~p |- p
```

  - Modus tollens is denoted as 'ModusTollens' rule:

```
~q, p > q |- ~p
```

  - Modus tollendo ponens is denoted as 'ModusTollendoPonens' rule:

```
~p, p|q |- q                ~q, p|q |- p
```

In the end we present the algorithm 4.2 of this prover. Note that the input of this prover is a set of premises and a conclusion that are from an argument. That argument is determined to be valid by tableaux prover, that is, there is always a derivation from the premises to the conclusion.

4.3 Tableaux prover

The tableaux prover is used to decide whether a conclusion is supported by a set of premises with the constraint that the set of premises must be consistent, meaning that the set cannot contain a proposition and its complement at the same time.

The notation of propositions and connectives are same as those in the natural deduction prover (Section 4.2). The syntactic rules applied in the prover can be found in Section 2.6. The corresponding algorithm 4.3 is as follows.

4.4 Argumentation Reasoner

In our system, the argumentation reasoner serves three purposes:

1. Calculate attack relations that is defined in Definition 3.1.
2. Find sets of accepted arguments with respect to admissible, grounded and ideal extension that is defined in Section 2.2.2.
3. Create dispute trees of a given argument with concerned extensions. Dispute trees is defined in Section 2.4.

These aims are achieved by Algorithm 4.4, Algorithm 4.5, and Algorithm 4.6, respectively. Note that in practice, unlike algorithm 4.6, because we already know the acceptance of an argument in terms of extensions, we only search for dispute trees for concerned extensions to save the computational cost.

Algorithm 4.2 Natural Deduction Prover

input: a set of premises Pre and a conclusion c
output: a natural deduction proof $\mathcal{P}$
function NATRUALDEDUCTIONPROVER(Pre, c)
    Transform propositions in Pre and C from plain text to nested structure according to their connectives.
    Let $\mathcal{C}$ be a set of clauses that contains all propositions in Pre.
    Set current conclusion c' to c
    while $c' \notin \mathcal{C}$ **do**
        Apply all possible inference rules to propositions in $\mathcal{C}$ according to their connectives.
        Add all propositions that generated by the former step to $\mathcal{C}$.
        if $c' \in \mathcal{C}$ **then**
            if In the assumption mode **then**
                Minus one level. Remove the assumption and all propositions obtained in this level of assumption from $\mathcal{C}$
                Add c' to $\mathcal{C}$.
                Set c' back to what it is before.
            elseBreak
            end if
        else
            Try to use inferences rules wisely with respect to c.
            Add all propositions that generated by the former step to $\mathcal{C}$.
            if $c' \in \mathcal{C}$ **then**
                if In the assumption mode **then**
                    Minus one level. Remove the assumption and all propositions obtained in this level of assumption from $\mathcal{C}$
                    Add c' to $\mathcal{C}$.
                    Set c' back to what it is before.
                elseBreak
                end if
            else
                Try to make an assumption a with respect to c.
                Add a to $\mathcal{C}$
                Set c' to the expected conclusion e that discharges the assumption.
                Enter the level one assumption mode. If already in the assumption mode, add one level.
            end if
        end if
    end while
    Record the whole reasoning procedure in Fitch notation as proof $\mathcal{P}$
    return $\mathcal{P}$
end function

Algorithm 4.3 Tableaux Prover

```
input: a set of premises Pre and a conclusion c
output: a boolean result B,B ∈ {True, False}
function TableauxProver(Pre, c)
    Transform propositions in Pre and C from plain text to nested structure according to their connectives.
    Let p be the conjunction of all propositions in Pre, then nests p and c in an implication i.
    Assume i to be false by sign it to be Fi.
    Set the current set of formulas {Fi}
    while True do
        Apply syntactic rules to signed formulas in current set of formulas according to their connectives.
        if The current set of formulas cannot be decomposed then
            return False
        else if The current set of formulas contain a signed formula and its conjugate then
            Mark current branch as closed.
            Move to another branch (another current set of formulas).
        else if New generated signed formulas coexists then
            Set current set of formulas to the set of new generated formulas.
        else if New generated signed formulas separate into two groups then
            Create two branches with each branch contains a maximal current sets of formulas that is different from the other.
            Continue with one of the branches and stock the other with all other branches.
        end if
        if all branches is closed then
            return True
        end if
    end while
end function
```

Algorithm 4.4 Find Arguments

input: a prepossessed propositional logic-based knowledge base Δ and a concerned claim c, $c \in \Delta$
output: a set of arguments $\mathcal{A}rgs$ and their attack relations *attacks*
function VISUALIZATION(Δ, c)
    Let $\mathcal{A}rgs$ be an empty set of arguments
    Develop arguments that has claim c using propositions in Δ.
    Add developed arguments to $\mathcal{A}rgs$.
    while there are new arguments added to $\mathcal{A}rgs$ **do**
        Develop arguments that attack at least one of the arguments in $\mathcal{A}rgs$ using propositions in Δ.
        Record all the attacks as binary relations onto *attacks*.
        Add developed arguments to $\mathcal{A}rgs$.
    end while
    return $\mathcal{A}rgs$, *attacks*
end function

Algorithm 4.5 Find Accepted Arguments of concerned extensions

input: a set of arguments $\mathcal{A}rgs$ and their attack relations *attacks*
output: the admissible extensions $\mathcal{E}_{\mathcal{AS}(\mathcal{A}rgs)}$, the grounded extension $\mathcal{E}_{\mathcal{GR}(\mathcal{A}rgs)}$ and the ideal extension $\mathcal{E}_{\mathcal{ID}(\mathcal{A}rgs)}$
function EXTENSION($\mathcal{A}rgs$), *attacks*)
    Find $\mathcal{E}_{\mathcal{AS}(\mathcal{A}rgs)}$, sets of admissible arguments, in all the combinations of $\mathcal{A}rgs$) referring to *attacks*.
    Determine $\mathcal{E}_{\mathcal{GR}(\mathcal{A}rgs)}$, a set of grounded arguments, from all sets in $\mathcal{E}_{\mathcal{AS}(\mathcal{A}rgs)}$ referring to *attacks*.
    Determine $\mathcal{E}_{\mathcal{ID}(\mathcal{A}rgs)}$, a set of ideal arguments, from all sets in $\mathcal{E}_{\mathcal{AS}(\mathcal{A}rgs)}$ referring to *attacks*.
    return $\mathcal{E}_{\mathcal{AS}(\mathcal{A}rgs)}$, $\mathcal{E}_{\mathcal{GR}(\mathcal{A}rgs)}$, $\mathcal{E}_{\mathcal{ID}(\mathcal{A}rgs)}$
end function

Algorithm 4.6 Create dispute trees

```
input: a set of arguments Args, their attack relations attacks and a concerned argument A, A ∈ Args
output: a set of dispute trees TREE for A that contains exactly one tree of each extension that accepts A
function DisputeTrees(A, Args, attacks)
    Let TREE contains only the first dispute tree T.
    Set the first proponent, the root node of T, as A.
    while True do
        Find all possible opponents that attacks each former proponent according to Args and attacks.
        if end or loop condition is met then
            Check what extension accepts this dispute tree.
            Add this dispute tree to TREE.
            if no more tree to be examined then
                break
            else
                Switch to next dispute tree
                continue
            end if
        else
            Add these opponents as children to each former proponents.
            Find all possible proponents that attacks each former opponent according to Args and attacks.
            if end or loop condition is met then
                Check what extension accepts this dispute tree.
                Add this dispute tree to TREE.
                if no more tree to be examined then
                    break
                else
                    Switch to next dispute tree
                    continue
                end if
            else
                Create as many dispute tree as the possible combinations. In particular, for each opponent, pick exactly one proponent in the set of proponents so that each opponent only has one child.
                Continue with one of the dispute trees and stock the others.
            end if
        end if
    end while
    Filter TREE so that for each extension there is only one corresponding tree in TREE.
    return TREE
end function
```

4.5 Natural language proof transformer

In order to give a text explanation that can be understood more easily for non expert users, we present a simple rewriting of the original natural deduction proof. In particular, for each line in the proof, we will translate the mentioned propositions and used reference rules to corresponding natural language texts. This feature is realized by regular expression tools as demonstrated in algorithm 4.7.

Algorithm 4.7 Natural language proof transformer

input: a prepossessed propositional logic-based knowledge base Δ and a natural deduction proof $\mathcal{P}$ for an argument in Δ
output: a natural language proof $\mathcal{P}'$
function ProofTransformer($\Delta, \mathcal{P}$)
    Let $\mathcal{P}'$ be an empty text string.
    for each line in $\mathcal{P}$ **do**
        Find the corresponding text descriptions in Δ for each proposition mentioned.
        Arrange these descriptions properly into a sentence.
        Append the sentence to $\mathcal{P}'$.
    end for
    return $\mathcal{P}'$
end function

4.6 Web interface

With the help of all the tools mentioned above, we can finally create a web application to visualize the explanations of a given knowledge base.

The layout of the web interface is designed into 4 rows and 3 columns as shown in figure 4.1 and table 4.1.

Some components that is worth remarking is listed as follows:

- The argument relation graph is a graph that given a concerned claim, the graph illustrates all related arguments as nodes and their attack relations as edges. The color of each node is correspond to its acceptance with respect to extensions.

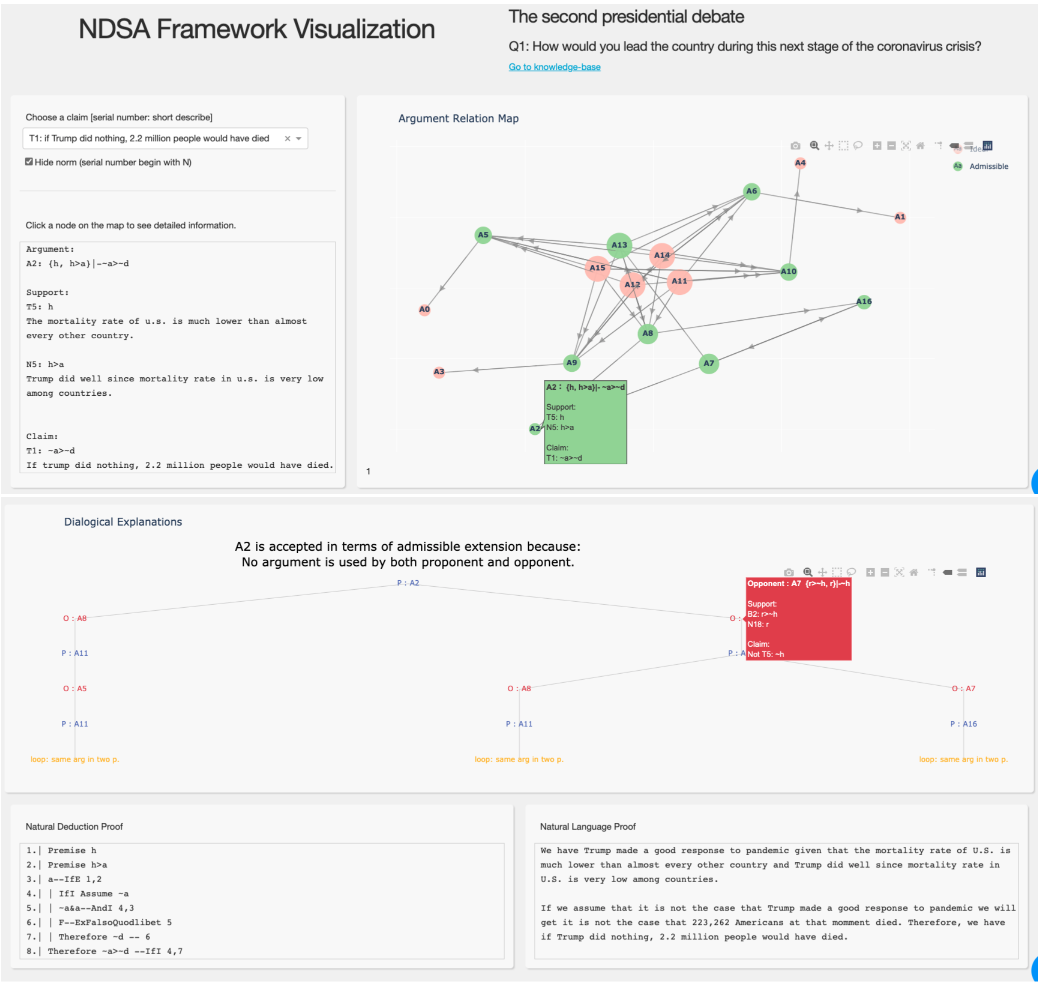


Figure 4.1: layout of NDSA Framework Visualization

Table 4.1: Layout for NDSA Visualization

<table>
<tr><td>Tile of the App</td><td>Tile of the use case
and a link to
the knowledge base</td></tr>
<tr><td>Dropdown for claims
and a text box for
detailed information</td><td>Argument relation graph</td></tr>
<tr><td colspan="2">Dispute tree(s)</td></tr>
<tr><td>Natural deduction proof</td><td>Natural language proof</td></tr>
</table>

- The dispute tree presents one tree for each extension that accepts the concerned arguments. Proponents is colored blue as well as opponents colored red. The end or loop condition is colored yellow.

The detail of knowledge base will be introduced in section 5.1.

Chapter 5

Evaluation

5.1 Use case

The use case in our work is a fragment of 2020 presidential debate in the United States. In particular, the debate is between two president candidates Donald Trump and Joe Biden that held in October 22, 2020. We use the transcription of the debate [1] as the raw material and modeled after two candidates' responses to the very first question in the debate. More detail about the transcript for our use case can be found in appendix A

For this question, we want to remark, Donald Trump goes first. As Dung hints in [2], in an argument-based discourse:

> *The one who has the last word laughs best.*

Meaning in a debate, a person who gives the last speech will have the advantages as what he or she says within this speech will not be attacked. Therefore, in our use case, Joe Biden will have the advantages since we only model the first question and within which he goes second.

For the speech given by each candidate, we divide it into statements not sentence-wisely but 'semantic-wisely'. To better demonstrate our idea, let us see the following example:

In his speech, Trump says:

> *We closed up the greatest economy in the world in order to fight this horrible disease that came from China.*

Although it is a whole sentence, we find it in fact have two aspects. The first aspect is about what he has done. He closed the United States as a response to the pandemic. The second aspect is about where the COVID

[1] www.speakwrite.com/transcripts/presidential-debate-2020-02/

came from. He mentioned it intentionally implying the cause of current situation is more objective than personal. Therefore, we divide this sentence into two statements as they stands for two different meanings.

Following the manner mentioned above, we finally get 10 statements for Trump and 11 statements for Biden. To make it comprehensible explicitly for both users and reasoning methods; however, we need to add the unexpressed premises. As mentioned in 2.1, we usually omit them in a daily discourse because we assume they are already known by our audience. But for audience not familiar with the context and for the reasoning system in the computer, we have to make them explicit. In our setting, we borrow the idea in [25] calling them *norms*, either *strict* or *defeasible*. Strict means norms are treated as fact while defeasible means norms are treated as tentaive information as long as no one disproves them.

In this way, we get 10 norms from Trump, 9 norms from Biden. Most of the norms we add to the knowledge base is related to an argument in the speech, but there are also some norms that are of a more general idea. For these norms, we do not assign them to a certain argument.

We also notice that it is often the case that the final sentence of a speech could be regarded as a conclusion of the whole speech. Nevertheless, we observe that all ideas of a speech may not be fully presented by a single conclusion. Thus, among the conclusion that is the final sentence, we also add two hidden conclusions so that we can summarize the idea of a speech more precisely. We also categories the statements into three groups where each group corresponds to a conclusion. Note that some statements may belongs to more than one group.

5.1.1 Annotation guideline

Now that we have taken a grasp on the setting, we elaborate a yet informal guideline of our annotation and provide examples along the way.

Although for the annotation, it is usually the case that , as in [26], researchers annotate their databases based on the consensus of several experts, we do it ourselves. This might result in a lack of confidence in saying that the labeling is 'correct'. But if we regard the annotation as a whole mapping from a transcript to a set of statements assigned to symbols which indicates their relations, then the difference in various of annotations will not undermine the capability of our system.

We first split the transcript TR into statements $\{STA_1,\ STA_2,\ \ldots,\ STA_n\}$ with respect to their meaning, as we discussed before. In the transcript, Trump goes first, then goes Biden. Therefore, we can say $\{STA_1,\ \ldots,\ STA_i\}$ is for Trump and $\{STA_{i+1},\ \ldots,\ STA_n\}$ is for Biden.

We then set up the conclusions. (1) We reassign STA_i as conclusion CON_1 for Trump, STA_n as conclusion CON_2 for Biden. (2) We then add hidden conclusions for each candidate as complement, which is discussed above. (3) At last, we will have a set of conclusions as $\{CON_1,\ \ldots,\ CON_m\}$.

To assign propositions to passages mentioned above, we consider syllogism. A syllogism is a logical deduction which, as defined by Aristotle, is from a general statement known as the major premise and a specific statement known as the minor premise to a conclusion. On the other hand, in our research, we regard a enthymeme as a truncated syllogism where one of the premises is unstated. A classic example of a syllogism is as follows:

1. All men are mortal. – major promise
2. Socrates is a man. – minor promise
3. Therefore, Socrates is mortal. – conclusion

If we omit the major promise, we will get a enthymeme as:

Socrates is mortal since Socrates is a man.

If we instead omit the minor promise, we will get another enthymeme as:

Socrates is mortal since all men are mortal.

We want to reconstruct the transcript into a combination of syllogisms. However, in practice, enthymemes dominates the debate. Therefore, we try to build enthymemes with statements and conclusions we have, then transform them to syllogisms by adding norms as unstated premises.

When we find a statement $STA_j,\ j \in \{1,\ \ldots,\ i-1,\ i+1,\ \ldots,\ n-1\}$ combined with a conclusion $CON_k,\ k \in \{1,\ \ldots,\ m\}$ can form a enthymeme, we add a norm NOR_α to reform it as a syllogism.

If the statement STA_j acts as the major premise, we entail a proposition to each passage as follows. Note that the assigned propositions are arbitrary as long as they are not used. p stands for literal.

1. STA_j – major premise – $p_a \to p_b$
2. NOR_α – minor premise – p_a
3. CON_k – conclusion – p_b

If the statement STA_j acts as the minor premise, we entail a proposition to each passage as follows. Note that the assigned propositions are arbitrary as long as they are not used. p stands for literal.

1. NOR_α – major premise – $p_c \to p_d$
2. STA_j – minor premise – p_c
3. CON_k – conclusion – p_d

The norm as major premise $p_c \to p_d$ corresponds to the statement p_c, while the norm as minor premise p_a does not correspond to any certain statement. A norm can be strict or defeasible. We sign it as strict if we believe it is a truth that can be proved by evidence in real life. We sign it as defeasible if we can not tell if it is true or not.

If we found two literal p_β, p_θ represent two passages that have opposite meanings, we replace the p_θ with $\neg p_\beta$.

In this way, we will finally get a knowledge base containing passages that typed as statement, conclusion and norm.

In [26], the author defines the data model with three concepts, topic, context dependent claim (CDC) and context dependent evidence (CDE). A topic is a subject of interest. A CDC supports or opposes the given topic, while a CDE directly supports a CDC within the context of the topic.

Informally, topic, CDC and CDE are similar to question, conclusion and statement in our setting, respectively. Note that however, we also add norm as a supplement, which might close to the concept of CDE.

Also, our model is close to Toulmin model [27], where conclusion, statement, norm corresponds to claim, grounds, warrant in Toulmin model,

Table 5.1: Format of the knowledge base and a sample proposition

question title						
serial number	original passage	speaker	passage type	summarized version used in the proof	corresp-onding propo-sition	the group where the passage belongs to
T1	2.2 million people modeled out were expected to die.	DONALD TRUMP	statement	if Trump did nothing, 2.2 million people would have died	∼a>∼d	a
...						

respectively.

5.1.2 Knowledge base description

The proposed knowledge base is stored as a csv file. The file contains 44 rows and 7 columns. The format is depict in table 5.1. Each column is designed as follows:

1. The serial number of each proposition begins with a capital letter following the number. The letter is defined as follows:
    - T stands for Trump's speech.
    - B stands for Biden's speech.
    - N stands for norm.
    - C stands for conclusion.
2. The original passage is a fragment extracted from transcript. Norms and hidden conclusions do not have corresponding original passage.
3. The speaker is either Trump or Biden.
4. Passage type is defined as follows:
    - statement: a passage extracted from the original speech.
    - norm()[]: norms are also called unexpressed premises, which are added accordingly in order to transform enthymematic proof into

syllogistic proof. The sub-type within the '()' can be either strict or defeasible. Strict means that this norm are treated as truth therefore cannot be attacked. Defeasible means this norm can be attacked as its correctness is unknown. The '[]' contains the proposition that the norm corresponds to.

- conclusion: statement that can be treated as a conclusion of the whole speech. Conclusions that are not in the original speech but given accordingly has (hidden) mark.

5. Summarized version of the original passage will be used and shown in the proof.

6. Corresponding proposition acts as a code of the passage to go through the proof machinery. Note that for atomic formula in our system, we only use alphabet from a to z.

7. The group where a passage belongs to is determined by its connection with a conclusion. Each group is named after the proposition of one of the conclusions. Each passage can belongs to more than one group. In this knowledge base, $conclusion = \{c, p, a, b, \sim b, \sim c\}$ and $group = \{c, p, a, b\}$.Note that b and $\sim b$ as well as c and $\sim c$ are in the same group for they attack each other.

   As an example, see the sample passage in table 5.1.It is in group a, which means it is related to the passage (not listed here) that is a conclusion and has a corresponding proposition a.

5.2 Unit Test

To examine whether the key components of our system is working as intended, we use unit test to evaluate the two provers.

5.2.1 Natural Deduction Prover

For natural deduction prover, the test cases fall into four categories:

- **Rules of implication test**
  - Test hypothetical syllogism

  - Test constructive dilemma

- **Rules of replacement test**
  - Test associativity (both sides)
  - Test commutativity (both sides)
  - Test distribution (two conditions, both sides)
  - Test exportation (both sides)
  - Test material implication (both sides)
  - Test tautology (two conditions, both sides)
  - Test transpositon

- **Conditional proof test**
  - 8 complex cases that involve the use of conditional proof.

- **Indirect proof test**
  - 6 complex cases that involve the use of indirect proof.

The test cases in the first two categories evaluate if our prover can stand for well-known properties of propositional logic that cannot be derived directly from inference rules as well as derived rules we added. The test cases in the last two categories evaluate if our prover can work properly on cases that needs extensive computation.

5.2.2 Tableaux Prover

For tableaux prover, we test the syntactic rules for each connective. These rules can be found in section 2.6.

- Test syntactic rules (α) for $\neg$.
- Test syntactic rules (α, β) for $\wedge$.
- Test syntactic rules (β, α) for $\vee$.
- Test syntactic rules (β, α) for $\rightarrow$.

For tableaux prover, we need it to determine whether a formula holds. Therefore, we add false test cases as well which is different from natural deduction prover that we use it to generate a proof for a formula proved to be true. Besides, we also use test with premise whether our prover can combine the promises with the conclusion together into a formula correctly.

5.3 Further evaluation proposal

The unit test mentioned in section 5.2 evaluates the correctness of our system. We can further evaluate the usefulness of our system according to the System Usability Scale (SUS) [2].

The SUS method provides a simple yet reliable way to measure the usability of a system. Since SUS is based on questionnaire, this measurement can apply to both expert and non-expert users without obstacles in terms of operation. the questionnaire of SUS generally contains 10 questions. Each question has 5 response options from strongly disagree to strongly agree. The corresponding score of each option is from 0 to 4. The way to interpret the score can be complex. Researchers usually multiply 2.5 to covert the original total score that is 0-40 to 0-100. Based on [28], a SUS score above 68 is considered above average.

A sample questionnaire is in table 5.2.

If we leverage this questionnaire to test the usefulness of system among non expert users, the step can be as follows.

1. Provide users the background of our use case.
2. Explain the basic idea of argument and show the users how to interpret texts and graphs explanations we implemented in our system.
3. Teach users how to interact with our system.
4. Let users explore our system.
5. Ask users to complete the SUS questionnaire.
6. Collect the questionnaire and interpret the result.

We also modified our application to create a 'non-expert' version for users without a background in logic or argumentation theory. In particular, we replace the usage of jargon, logic formula in propositional logic, Fitch-style proof and graphics with plain text explanations. Further discussion can be found in section 6.2.

[2] www.usability.gov/how-to-and-tools/methods/system-usability-scale.html

Table 5.2: System Usability Scale Questionnaire

#	Question	Scale						
1	I think that I would like to use this system frequently.	Strongly disagree	1	2	3	4	5	Strongly agree
2	I found the system unnecessarily complex.	Strongly disagree	1	2	3	4	5	Strongly agree
3	I thought the system was easy to use.	Strongly disagree	1	2	3	4	5	Strongly agree
4	I think that I would need the support of a technical person to be able to use this system.	Strongly disagree	1	2	3	4	5	Strongly agree
5	I found the various functions in this system were well integrated.	Strongly disagree	1	2	3	4	5	Strongly agree
6	I thought there was too much inconsistency in this system.	Strongly disagree	1	2	3	4	5	Strongly agree
7	I would imagine that most people would learn to use this system very quickly.	Strongly disagree	1	2	3	4	5	Strongly agree
8	I found the system very cumbersome to use.	Strongly disagree	1	2	3	4	5	Strongly agree
9	I felt very confident using the system.	Strongly disagree	1	2	3	4	5	Strongly agree
10	I needed to learn a lot of things before I could get going with this system.	Strongly disagree	1	2	3	4	5	Strongly agree

The non-expert version is deployed using Heroku [3] on `https://ndsa-debate-visualization.herokuapp.com/` as shown in figure 5.1 and table 5.3.

[3] www.heroku.com/

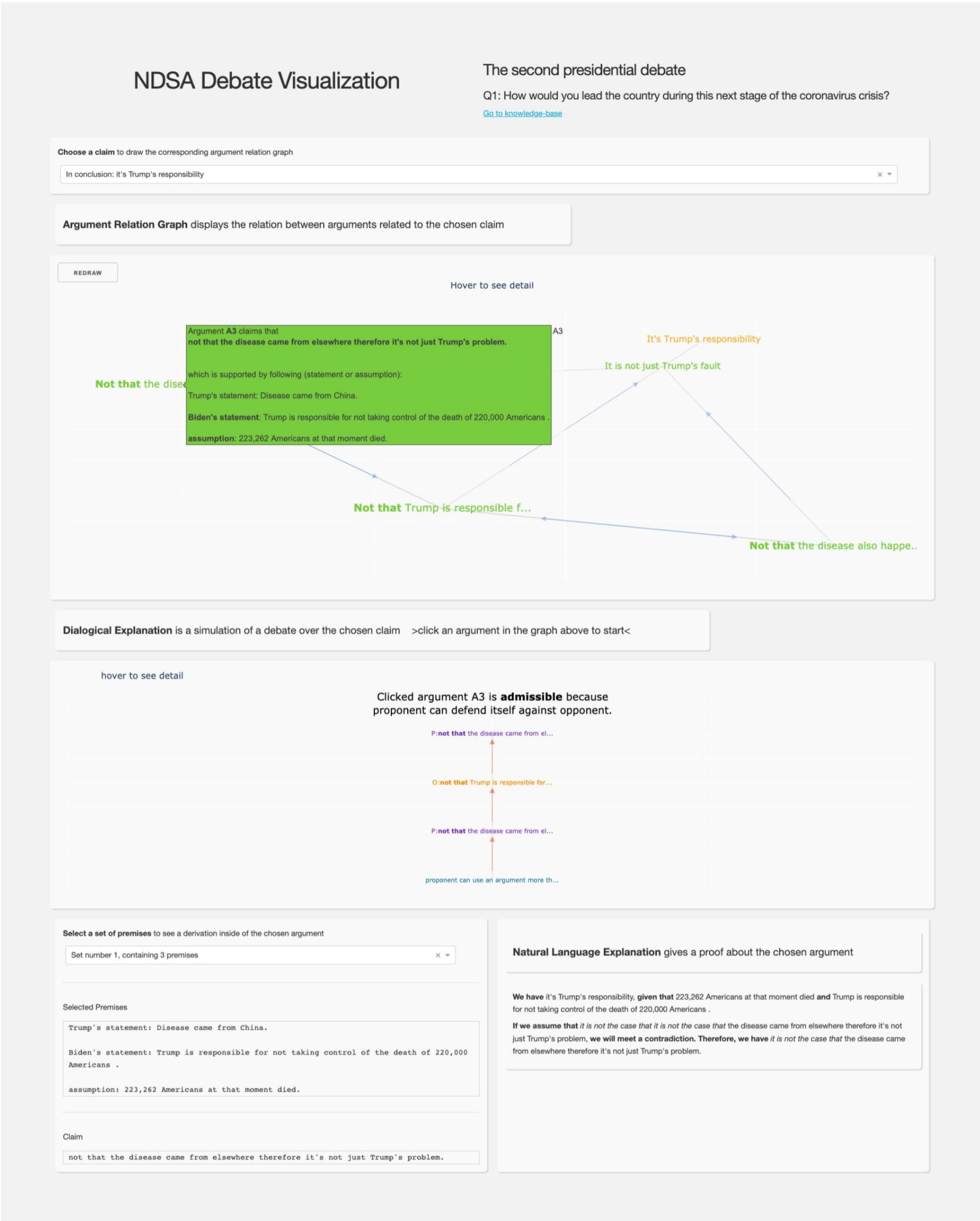


Figure 5.1: Layout for non-expert version

<table>
<tr><td>Tile of the App</td><td>Tile of the use case
and a link to
the knowledge base</td></tr>
<tr><td colspan="2">Dropdown for choosing a claim</td></tr>
<tr><td colspan="2">Argument relation graph</td></tr>
<tr><td colspan="2">Dispute tree(s)</td></tr>
<tr><td>Dropdown for choosing
a set of premise and
a text box for detail</td><td>Natural language proof</td></tr>
</table>

Table 5.3: Layout for non-expert version

Chapter 6

Conclusion

6.1 Conclusion

We present a NDSA framework based visualization system that can reason in a inconsistent propositional logic-based knowledge base and provide human-friendly explanations to expert and non expert users.

Given a knowledge base:

1. We model it as an abstract argumentation framework so that the knowledge base can be regarded as a set of arguments and their attack relations. We then visualize it as a direct graph.
2. We calculate the acceptance of each argument in terms of extensions and illustrate this status using dispute trees as an explanation to it.
3. We go into each argument and develop a natural deduction proof of it. We further show the proof in Fitch-style as an explanation of the validity of the argument.
4. In order to make the natural deduction proof even more intelligible, we translate the proof to natural language as an text explanation.

We demonstrate the capability of our system in practical reasoning using an concrete use case. Through the visualization, we not only model a real debate but also provide text and visual explanation to it.

Furthermore, we examine the correctness of our system using unit test and propose an plan for a more profound SUS usefulness test.

With all aspects mentioned above, we argue that our system can be regarded as a helpful tool for practical reasoning as our system models and explains it.

6.2 Future Direction

The future direction of our system has four aspects: theoretical aspect, computational aspect, visual aspect and collaborative aspect.

- **Theoretical aspect**
  In our use case, almost all arguments are admissible. To distinguish the acceptance of each argument more precisely, here is some options.
  - We can introduce value-based argumentation framework that take the reliability of each argument into account.
  - We can introduce more semantics to calculate more levels of acceptance.

- **Computational aspect**
  Our system reads and models a given knowledge base in real time because we believe real time calculation supports many valuable features even though we have not implemented them yet. We will discuss them later in this section. However, sometimes our system becomes extremely slow for: (1) The computational cost for our system is relatively high especially for the algorithm 4.5. As indicated by Line 4 in algorithm 4.5, when the number of arguments grows, the number of combinations we need to examine grows exponentially. (2) Our use case is very complex for it is a full modeling though only a fragment of a real debate.

  To boost the real time performance of our system, we consider following modification in our future work:

  - **Change the programming language**
    In our implementation, we use Python to build the whole system as it is a popular tool for visualization. Nevertheless, the fact that Python is an interpreted language limits its performance. By rewriting our system in a compiled language such as C++ and JAVA, we can save a lot of running time.
  - **Change the algorithm of some components**
    Answer set programming might be another possible option since it is able to achieve the goal of algorithm 4.5 in an efficient way. Some algorithms in answer set programming even support real time updating of the knowledge base without calculating it as a

whole. Therefore, by using answer set programming to cope with algorithms involving massive set calculation, we may be able to save a lot of running time in our system as well.

- **Visual aspect**
We represent arguments in our visualization following the convention in the study of formal argumentation. That is, the code of an argument (in NDSA setting, the code can be the premises and the conclusion written as $\{a_1, ...\} \vdash c$) seat in a circle of a rectangle. While an argument in this form is no stranger to expert, we find it might be hard for non expert users to understand what these codes exactly means in a glance. Although we already include detailed information of an argument when the user hover their mouse over that argument in our system, we can in addition replace the graphics and codes form of an argument with natural language representation. Highlighting keywords in text explanations can be helpful as well.

- **Collaborative aspect**
In our user case, we split and annotate the transcript by hand. With the help of machine learning approaches such as argument mining, we might be able to automate the prepossessing step of our system in the future. On the other hand, natural language processing approaches can help us improve the quality of generated natural language explanations.

We hope in the near future, with the realization of all aspect mentioned above, we can build a system that for an ongoing real debate, whenever the stenographer inputs a sentence, the system can automatically annotate it and update the knowledge base. Users that watch the debate on the internet can have the live stream window to the right and a dynamic interactive application to the right. The application updates simultaneously with the live stream and provides all kinds of easy-understandable explanations to its users.

Appendix A

Use Case

We use the transcript for the first question

> *How would you lead the country during this next stage of the coronavirus crisis?*

in 2020 Election Second Presidential Debate between Joe Biden and Donald Trump as our use case.

This part is in appendix B. Note that we omit the speech of the moderator Kristen Welker as well as some of the fillers in the speeches of both candidates in our use case.

The full transcript can be found on https://speakwrite.com/transcripts/presidential-debate-2.

Our knowledge base is in appendix C. The notations used in the knowledge base in given in 5.1.

The knowledge base as well as our code can be found on https://github.com/xlives/master-thesis-gu-chengwei.

Online deployment of expert version (original version discussed in section 4.6) is on https://ndsa-viz-expert.herokuapp.com/.

The non-expert version (modified version for practical use discussed in section 5.3) is on https://ndsa-debate-visualization.herokuapp.com/.

Appendix B

Transcript

KRISTEN WELKER: Good evening from Belmont University in Nashville, Tennessee. I'm Kristen Welker of NBC News, and I welcome you to the final 2020 Presidential Debate between President Donald J. Trump and former Vice President Joe Biden. Tonight's debate is sponsored by the Commission on Presidential Debates. It is conducted under health and safety protocols designed by the Commission's health security advisor. The audience here in the hall has promised to remain silent. No cheers, boos, or other interruptions except right now, as we welcome to the stage former Vice President Joe Biden and President Donald J. Trump. And I do want to say a very good evening to both of you. This debate will cover six major topics. At the beginning of each section, each candidate will have 2 minutes uninterrupted to answer my first question. The debate commission will then turn on their microphone only when it is their turn to answer, and the commission will turn it off exactly when the 2 minutes have expired. After that, both microphones will remain on, but on behalf of the voters, I'm going to ask you to please speak one at a time. The goal is for you to hear each other and for the American people to hear every word of what you both have to say. And so, with that, if you're ready, let's start, and we will begin with the fight against the coronavirus. President Trump, the first question is for you. The country is heading into a dangerous new phase. More than 40,000 Americans are in the hospital tonight with COVID, including record numbers here in Tennessee, and since the two of you last shared a stage, 16,000 Americans have died from COVID. So please be specific. **How would you lead the country during this next stage of the coronavirus crisis?** Two minutes uninterrupted.

DONALD TRUMP: So, as you know, 2.2 million people modeled out were expected to die. We closed up the greatest economy in the world in order to fight this horrible disease that came from China. It's a worldwide pandemic. It's all over the world. You see the spikes in Europe and many other places right now. Uh, if you notice, the mortality rate is down 85 percent. Uh, the excess mortality rate is way down and much lower than almost every other country, and we're fighting it, and we're fighting it hard. There is a spike – there was a spike in Florida, and it's now gone. There was a very big spike in Texas; it's now gone. There was a very big spike in Arizona; it's now gone, and there are some spikes and surges in other places. They will soon be gone. We have a vaccine that's coming. It's ready. It's going to be announced within weeks, and it's going to be delivered. We have, uh, Operation Warp Speed, which is – the military is going to distribute the vaccine. I can tell you from personal experience that, uh, I was in the hospital. I had it and I got better, and I will tell you that, uh, I had something that they gave me, a therapeutic, I guess they would call it. Some people could say it was a cure, but, uh, I was in for a short period of time and I got better very fast, or I wouldn't be here tonight, and now they say I'm immune. Whether it's 4 months or a lifetime, nobody's been able to say that, but I'm immune. Uh, more and more people are getting better. We have, uh, a problem that's a worldwide problem. This is a worldwide problem, but I've been congratulated by the heads of many countries on what we've been able to do, uh, with the – if you, if you take a look at what we've done in terms of goggles and masks and gowns and everything else, and, in particular, ventilators. We're now making ventilators all over the world, thousands and thousands a month, distributing them all over the world. It will go away, and as I say, we're rounding the turn, we're rounding the corner. It's going away.

KRISTEN WELKER: Okay, former Vice President Biden, to you, **how would you lead the country out of this crisis?** You have 2 minutes uninterrupted.

JOE BIDEN: 220,000 Americans dead. If you hear nothing else I say tonight, hear this. Anyone who's responsible for not taking control, in fact, not saying I'm – I take no responsibility initially, anyone who's responsible for that many deaths should not remain as President of the United States of America. We're in a situation where there are a thousand deaths a day now. A thousand deaths a day, and there are over 70,000 new cases per day. Compared to what's going on in Europe, as the New England Medical Journal said, they're starting from a very low rate, we're starting from a very high rate. The expectation is we'll have another 200,000 Americans dead between now and the end of the year. If we just wore these masks, the President's own advisors had told him, we could save 100,000 lives. And we're in a circumstance where the President, thus far, still has no plan, no comprehensive plan. What I would do is make sure we have everyone encouraged to wear a mask all the time. I would make sure we move in the direction of rapid testing, investing in rapid testing. I would make sure that we set up national standards as to how to open up schools and open up businesses so they can be safe and give them the wherewithal, the financial resources to be able to do that. We're in a situation now where the New England Medical Journal, one of the serious, most-serious journals in the, in the whole world, said for the first time ever that this, the way this President has responded to this crisis has been absolutely tragic. And so folks, I will take care of this. I will end this. I will make sure we have a plan.

Appendix C

Knowledge Base

Table C.1: Full knowledge base

number	origin	speaker	type	proof	proposition	group
T1	2.2 million people modeled out were expected to die.	DONALD TRUMP	statement	if Trump did nothing, 2.2 million people would have died	∼a>∼d	a
T2	We closed up the greatest economy in the world in order to fight this horrible disease	DONALD TRUMP	statement	Trump closed up the country	e	a
T3	that came from China.	DONALD TRUMP	statement	disease came from China	f	b
T4	It's a worldwide pandemic. It's all over the world. You see the spikes in Europe and many other places right now.	DONALD TRUMP	statement	it's a worldwide pandemic.There are spikes in Europe and many other places right now	g	b
T5	Uh, if you notice, the mortality rate is down 85 percent. Uh, the excess mortality rate is way down and much lower than almost every other country,	DONALD TRUMP	statement	the mortality rate of U.S. is much lower than almost every other country	h	a
T6	and we're fighting it, and we're fighting it hard. There is a spike – there was a spike in Florida, and it's now gone. There was a very big spike in Texas; it's now gone. There was a very big spike in Arizona; it's now gone, and there are some spikes and surges in other places. They will soon be gone.	DONALD TRUMP	statement	Trump are fighting it hard since there were some spikes and surges and they are all gone now	i	ac
T7	We have a vaccine that's coming. It's ready. It's going to be announced within weeks, and it's going to be delivered. We have, uh, Operation Warp Speed, which is – the military is going to distribute the vaccine.	DONALD TRUMP	statement	U.S. will have a vaccine within weeks	j	c
T8	I can tell you from personal experience that, uh, I was in the hospital. I had it and I got better, and I will tell you that, uh, I had something that they gave me, a therapeutic, I guess they would call it. Some people could say it was a cure, but, uh, I was in for a short period of time and I got better very fast, or I wouldn't be here tonight, and now they say I'm immune. Whether it's 4 months or a lifetime, nobody's been able to say that, but I'm immune. Uh, more and more people are getting better.	DONALD TRUMP	statement	Trump had it and got better very fast as well as More and more people getting better	k	c

Table C.1 continued from previous page

number	origin	speaker	type	proof	proposition	group
T9	We have, uh, a problem that's a worldwide problem. This is a worldwide problem, but I've been congratulated by the heads of many countries on what we've been able to do,uh, with the – if you, if you take a look at what we've done in terms of goggles and masks and gowns and everything else, and, in particular, ventilators. We're now making ventilators all over the world, thousands and thousands a month, distributing them all over the world.	DONALD TRUMP	statement	Trump is helping the world against the pandamic	l	a
T10	It will go away, and as I say, we're rounding the turn, we're rounding the corner. It's going away.	DONALD TRUMP	conclusion	it's going away	c	c
B1	220,000 Americans dead. If you hear nothing else I say tonight, hear this. Anyone who's responsible for not taking control, in fact, not saying I'm – I take no responsibility initially, anyone who's responsible for that many deaths should not remain as President of the United States of America.	JOE BIDEN	statement	Trump is responsible for not taking control of the death of 220,000 Americans	d>∼b	b
B2	We're in a situation where there are a thousand deaths a day now. A thousand deaths a day, and there are over 70,000 new cases per day. Compared to what's going on in Europe, as the New England Medical Journal said, they're starting from a very low rate, we're starting from a very high rate.	JOE BIDEN	statement	we are in a bad situation, as the New England Medical Journal said, U.S. are starting from a very high rate compare to Europe	r>∼h	a
B3	The expectation is we'll have another 200,000 Americans dead between now and the end of the year.	JOE BIDEN	statement	the expectation is another 200,000 Americans will die between now and the end of the year	s	c
B4	If we just wore these masks, the President's own advisors had told him, we could save 100,000 lives.	JOE BIDEN	statement	if Trump encouraged people to wear masks, 100,000 lives could be saved	o>∼d	b
B5	And we're in a circumstance where the President, thus far, still has no plan, no comprehensive plan.	JOE BIDEN	statement	Trump has no plan	v	c
B6	What I would do is make sure we have everyone encouraged to wear a mask all the time.	JOE BIDEN	statement	Biden will have everyone encouraged to wear a mask all the time	t	p
B7	I would make sure we move in the direction of rapid testing, investing in rapid testing.	JOE BIDEN	statement	Biden will investing in rapid testing	w	p
B8	I would make sure that we set up national standards as to how to open up schools and open up businesses so they can be safe	JOE BIDEN	statement	Biden will set up national standards as to how to open up safely	x	p
B9	give them the wherewithal, the financial resources to be able to do that.	JOE BIDEN	statement	Biden will supports people with financial resources	y	p

Table C.1 continued from previous page

number	origin	speaker	type	proof	proposition	group
B10	We're in a situation now where the New England Medical Journal, one of the serious, most-serious journals in the, in the whole world, said for the first time ever that this, the way this President has responded to this crisis has been absolutely tragic.	JOE BIDEN	statement	the New England Medical Journal claimed that the way Trump has responded to this crisis has been absolutely tragic	r>∼a	a
B11	And so folks, I will take care of this. I will end this. I will make sure we have a plan.	JOE BIDEN	conclusion	Biden will end this with a plan	p	p
N1	/	DONALD TRUMP / JOE BIDEN	norm(strict)[]	223,262 Americans at that momment died	d	ab
N2	/	DONALD TRUMP	norm(defeasible)[e]	closing the boundary is a response	e>a	a
N3	/	DONALD TRUMP	norm(defeasible)[f]	the disease came from elsewhere therefore it's not just Trump's problem	f>b	b
N4	/	DONALD TRUMP	norm(defeasible)[g]	the disease also happens in elsewhere therefore it's not just Trump's problem	g>b	b
N5	/	DONALD TRUMP	norm(strict)[h]	Trump did well since mortality rate in U.S. is very low among countries	h>a	a
N6	/	DONALD TRUMP	norm(defeasible)[i]	Trump did well since Trump settled the spikes	i>a	a
N7	/	DONALD TRUMP	norm(defeasible)[i]	Trump will win against spikes in the future as Trump did before	i>c	c
N8	/	DONALD TRUMP	norm(defeasible)[j]	once vaccine come the disease gone	j>c	c
N9	/	DONALD TRUMP	norm(defeasible)[k]	people don't have to worry since the disease can be cured	k>c	c
N10	/	DONALD TRUMP	norm(defeasible)[l]	Trump made a good response since he was helping the world	l>a	a
N11	/	JOE BIDEN	norm(strict)[s]	another 200,000 Americans might die till the end of year is a bad situation	s>∼c	c
N12	/	JOE BIDEN	norm(strict)[t]	encouraging people to wear mask is part of the plan	t>p	p
N13	/	JOE BIDEN	norm(strict)[w]	investing rapid testing is part of the plan	w>p	p
N14	/	JOE BIDEN	norm(strict)[x]	standard is essential for opening up safely	x>z	p
N15	/	JOE BIDEN	norm(strict)[y]	financial resources is essential for opening up safely	y>z	p
N16	/	JOE BIDEN	norm(strict)[]	a safe open up is necessary for American people	z	p
N17	/	JOE BIDEN	norm(strict)[z]	opening up safely is part of the plan	z>p	p
N18	/	JOE BIDEN	norm(strict)[]	New England Medical Journal is one of the most serious journals in the whole world	r	a
N19	/	JOE BIDEN	norm(defeasible)[v]	without a plan pandemic won't end	v>∼c	c
C1	/	DONALD TRUMP	conclusion(hidden)	Trump made a good response to pandemic	a	a
C2	/	DONALD TRUMP	conclusion(hidden)	it is not just Trump's fault	b	b
C3	/	JOE BIDEN	conclusion(hidden)	it's Trump's responsibility	∼b	b
C4	/	JOE BIDEN	conclusion(hidden)	the situation is bad	∼c	c

References

[1] A. B. Arrieta, N. Díaz-Rodríguez, J. D. Ser, A. Bennetot, S. Tabik, A. Barbado, S. García, S. Gil-López, D. Molina, R. Benjamins, R. Chatila, and F. Herrera, "Explainable artificial intelligence (xai): Concepts, taxonomies, opportunities and challenges toward responsible ai," 2019.

[2] P. M. Dung, "On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games," *Artificial intelligence*, vol. 77, no. 2, pp. 321–357, 1995.

[3] R. Guidotti, A. Monreale, S. Ruggieri, F. Turini, F. Giannotti, and D. Pedreschi, "A survey of methods for explaining black box models," *ACM computing surveys (CSUR)*, vol. 51, no. 5, pp. 1–42, 2018.

[4] F. van Eemeren and B. Verheij, "Argumentation theory in formal and computational perspective," in *Handbook of Formal Argumentation*, G. M. . v. d. T. L. Baroni P., Gabbay D., Ed. London: College Publications, 2018, pp. 3–73.

[5] J. L. Pollock, "Defeasible reasoning," *Cognitive science*, vol. 11, no. 4, pp. 481–518, 1987.

[6] P. Besnard and A. Hunter, *Elements of argumentation*. MIT press Cambridge, 2008, vol. 47.

[7] P. M. Dung, P. Mancarella, and F. Toni, "Computing ideal sceptical argumentation," *Artificial Intelligence*, vol. 171, no. 10-15, pp. 642–674, 2007.

[8] B. Verheij, "Two approaches to dialectical argumentation: admissible sets and argumentation stages," *Proc. NAIC*, vol. 96, pp. 357–368, 1996.

[9] M. Caminada, "Semi-stable semantics," *COMMA*, vol. 144, pp. 121–130, 2006.

[10] ——, "Comparing two unique extension semantics for formal argumentation: ideal and eager," in *Proceedings of the 19th Belgian-Dutch conference on artificial intelligence (BNAIC 2007)*. Utrecht University Press, 2007, pp. 81–87.

[11] A. Bondarenko, F. Toni, and R. A. Kowalski, "An assumption-based framework for non-monotonic reasoning." in *LPNMR*, vol. 93, 1993, pp. 171–189.

[12] A. Bondarenko, P. M. Dung, R. A. Kowalski, and F. Toni, "An abstract, argumentation-theoretic approach to default reasoning," *Artificial intelligence*, vol. 93, no. 1-2, pp. 63–101, 1997.

[13] P. M. Dung, R. A. Kowalski, and F. Toni, "Assumption-based argumentation," in *Argumentation in Artificial Intelligence*, G. R. Simari and I. Rahwan, Eds. Springer, 2009, pp. 199–218.

[14] F. Toni, "A tutorial on assumption-based argumentation," *Argument & Computation*, vol. 5, no. 1, pp. 89–117, 2014.

[15] P. Besnard, A. Garcia, A. Hunter, S. Modgil, H. Prakken, G. Simari, and F. Toni, "Introduction to structured argumentation," *Argument & Computation*, vol. 5, no. 1, pp. 1–4, 2014.

[16] F. Toni, "Reasoning on the web with assumption-based argumentation," in *Reasoning Web International Summer School*. Springer, 2012, pp. 370–386.

[17] M. W. A. Caminada, S. Sá, J. Alcântara, and W. Dvořák, "On the difference between assumption-based argumentation and abstract argumentation," *IfCoLog Journal of Logics and their Applications*, 2015.

[18] P. M. Dung, R. A. Kowalski, and F. Toni, "Dialectic proof procedures for assumption-based, admissible argumentation," *Artificial Intelligence*, vol. 170, no. 2, pp. 114–159, 2006.

[19] T. Racharak and S. Tojo, "On explanation of propositional logic-based argumentation system," in *In Proceedings of 13th International Conference on Agents and Artificial Intelligence*. ICAART, 2021, p. Online Streaming.

[20] G. Gentzen, "Untersuchungen über das logische schließen. i," *Mathematische zeitschrift*, vol. 39, no. 1, pp. 176–210, 1935.

[21] D. Van Dalen, *Logic and structure*. Springer, 2004.

[22] F. B. Fitch, "Symbolic logic," 1952.

[23] R. M. Smullyan, "Analytic cut," *The Journal of Symbolic Logic*, vol. 33, no. 4, pp. 560–564, 1968.

[24] L. Amgoud, P. Besnard, and S. Vesic, “Identifying the core of logic-based argumentation systems,” in *2011 IEEE 23rd International Conference on Tools with Artificial Intelligence*. IEEE, 2011, pp. 633–636.

[25] T. Racharak, S. Tojo, N. D. Hung, and P. Boonkwan, “Argument-based logic programming for analogical reasoning,” in *JSAI International Symposium on Artificial Intelligence*. Springer, 2016, pp. 253–269.

[26] E. Aharoni, A. Polnarov, T. Lavee, D. Hershcovich, R. Levy, R. Rinott, D. Gutfreund, and N. Slonim, “A benchmark dataset for automatic detection of claims and evidence in the context of controversial topics,” in *Proceedings of the first workshop on argumentation mining*, 2014, pp. 64–68.

[27] D. Hitchcock and B. Verheij, *Arguing on the Toulmin model*. Springer, vol. 10.

[28] A. Bangor, P. Kortum, and J. Miller, “Determining what individual sus scores mean: Adding an adjective rating scale,” *Journal of usability studies*, vol. 4, no. 3, pp. 114–123, 2009.