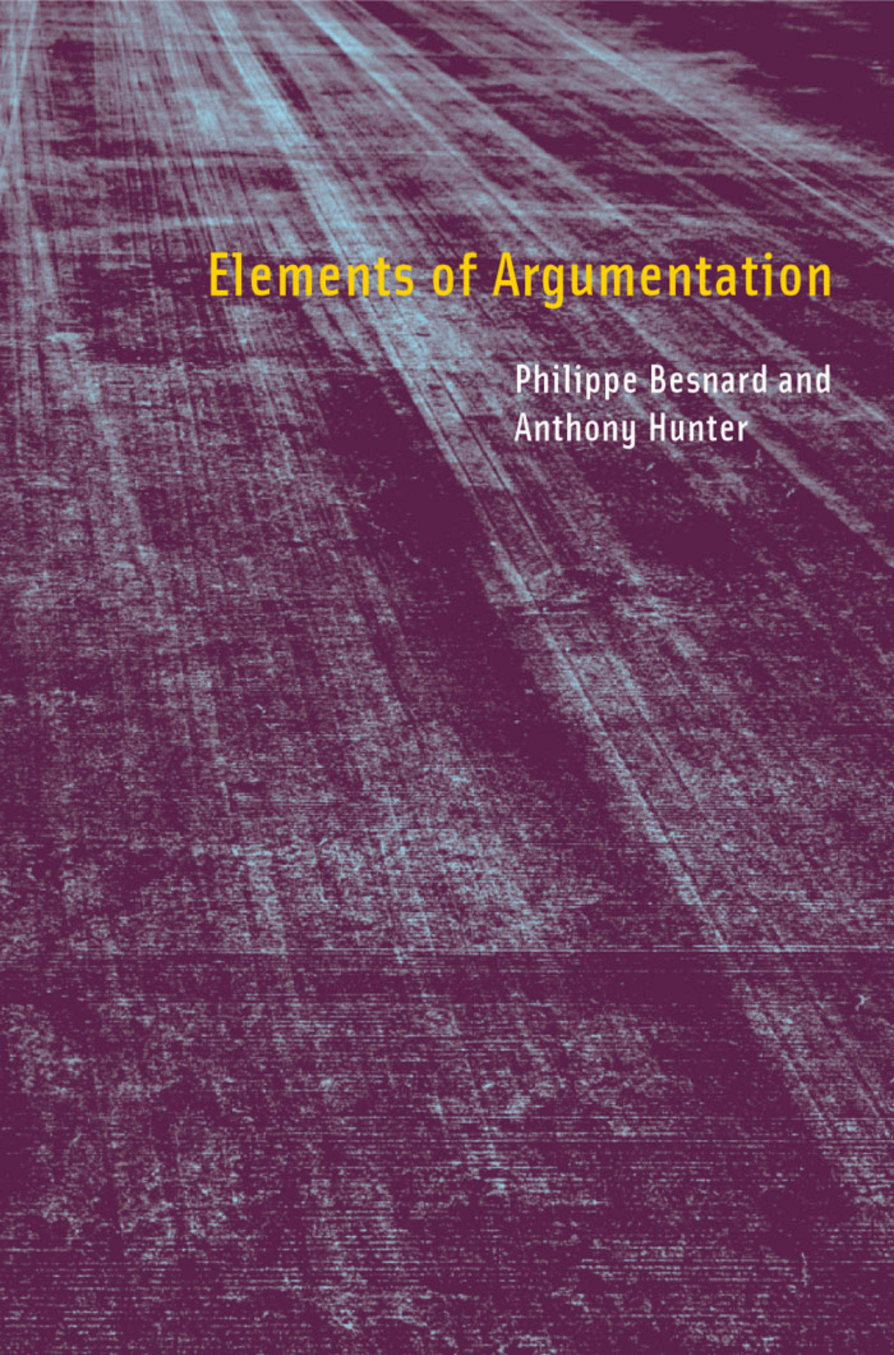


Elements of Argumentation

Philippe Besnard and
Anthony Hunter

Elements of Argumentation

Elements of Argumentation

Philippe Besnard and Anthony Hunter

The MIT Press
Cambridge, Massachusetts
London, England

For information about special quantity discounts, please email special_sales@mitpress.mit.edu

This book was set in Times New Roman and Syntax on 3B2 by Asco Typesetters, Hong Kong.
Printed and bound in the United States of America.

Library of Congress Cataloging-in-Publication Data

Besnard, Philippe, 1958–
Elements of argumentation / Philippe Besnard and Anthony Hunter.
p. cm.
Includes bibliographical references and index.
ISBN 978-0-262-02643-7 (hardcover : alk. paper) 1. Artificial intelligence. 2. Reasoning. 3. Logic. 4. Information technology. I. Hunter, Anthony, 1962– II. Title.
Q335.B489 2008
006.3—dc22 2007032257

10 9 8 7 6 5 4 3 2 1

Preface

Logic-based formalizations of argumentation that take pros and cons for some conclusion into account have been extensively studied over a number of years, and some basic principles have now been clearly established. These formalizations assume a set of formulae and then lay out arguments and counterarguments that can be obtained from these assumed formulae.

More recently, attempts have been made to refine these formalizations in order to more closely capture practical argumentation as used in the real world. This has led to techniques for selecting and reforming the more appropriate arguments and counterarguments for use in problem analysis and decision making. These techniques identify the better arguments based on (1) taking into account intrinsic aspects of the arguments such as their relative consistency, the exhaustiveness of the consideration of counterarguments, and the relative similarities between arguments and (2) taking into account extrinsic factors such as the impact on the audience and the beliefs of the audience.

The aim of this book is to introduce the background and techniques for formalizing argumentation in artificial intelligence and, in particular, to cover the emerging formalizations for practical argumentation. This coverage includes considering how arguments can be constructed, how counterarguments can be identified, how key intrinsic and extrinsic factors can be analyzed, and how these analyses can be harnessed for formalizing practical argumentation. As a part of this coverage, we aim to elucidate and formalize some of the key elements of argumentation.

This book focuses on argumentation by an agent presenting a case for some claim. This involves providing an initiating argument for the claim; then providing undercuts to this argument; and then, by recursion, providing undercuts to undercuts. Each undercut is a counterargument that

Contents

contradicts the premises of an argument. This approach to argumentation can be described as monological as opposed to dialogical.

In monological argumentation, there is a set of possibly conflicting pieces of information (each piece of information is represented by a formula) that has been collated by some agent, or pooled by a set of agents, and the role of argumentation is to construct a constellation of arguments and counterarguments pertaining to some particular claim of interest (i.e., the subject of the argumentation). The presentation of arguments by an agent may then be for the agent to analyze some pieces of information for his or her own purposes (auto-argumentation) or to present them to some audience (e.g., a politician giving a speech to members of an electorate, a journalist writing an editorial in a newspaper, a clinician explaining a treatment plan to a patient, etc.).

In contrast, dialogical argumentation is conducted between two or more agents and involves more complex issues of multi-agent interaction, dialogue protocols, and strategies. This book does not cover dialogical argumentation. However, it is clear that progress in formalizing monological argumentation is necessary for developing better dialogical argumentation systems.

The intended audience for this book consists of researchers interested in the knowledge representation and reasoning issues surrounding argumentation, either to study existing formalisms or to apply and adapt techniques for real problems. The intended readers are therefore in artificial intelligence and computer science. In addition, it is hoped that the book may be relevant to a wider range of readers in logic, philosophy, linguistics, and cognitive science and that it could be used as a primary or secondary text for advanced undergraduate and postgraduate courses in logic and argumentation. Readers will be expected to have had some exposure to classical propositional and predicate logic, though appendix C is a review to remind the reader of the basic concepts and notation. No other prerequisites are assumed.

Acknowledgments

The research undertaken by us that forms the basis of this book has been supported, in part, by funding from EU IST, CNRS, EPSRC, and The Royal Society.

We are very grateful for a number of colleagues who have provided feedback on draft versions of this book, including Leila Amgoud, Robert Bowerman, Claudette Cayrol, Sylvie Doutre, Vasiliki Efstathiou, Jeremy Forth, Paul Krause, Marie-Christine Lagasquie-Schiex, Sanjay Modgil, Yves Moinard, Nick Mann, Juan Gabriel Olarte Garcia, Simon Parsons, Henry Prakken, Guillermo Simari, and Francesca Toni. Of course, any errors and inadequacies in the book are entirely our responsibility.

Finally, we would like to express gratitude to Philippe Smets, who, as the organizer of a series of European projects, provided a wonderful pluralistic environment for proponents of diverse approaches to knowledge representation and reasoning formalisms for inconsistency and uncertainty to come together and fruitfully collaborate. We were both very fortunate to work with him on some of these projects.

Elements of Argumentation

1 Nature of Argumentation

Argumentation normally involves identifying relevant assumptions and conclusions for a given problem being analyzed. Furthermore, it often involves identifying conflicts, resulting in the need to look for pros and cons for particular conclusions.

Argumentation is a vital form of human cognition. Constantly in our daily lives, we are confronted with information that conflicts, and we are forced to deal with the resulting inconsistencies. Often we do this subconsciously: As a mental reflex, we weigh conflicting information and select some items of information in preference to others. Some of the time, we deal with conflicting information in a more conscious way. For example, if we are making a big decision, we may have in mind some of the key arguments and counterarguments. Consider a decision on where to go for a long holiday or a decision on a house to buy. Here, there is a list of options with pros and cons for each option. And when we are not sure about inconsistencies in our information, we may try to seek better information, or to seek advice, in order to resolve the inconsistency.

Professionals routinely undertake argumentation as an integral part of their work. Consider diverse types of professional, such as clinicians, scientists, lawyers, journalists, and managers, who have to identify pros and cons for analyzing situations prior to presenting some information to an audience and/or prior to making some decision. Here, many conflicts are consciously identified in the available information, and then, depending on the task being undertaken, appropriate arguments and counterarguments are constructed.

Argumentation may also involve chains of reasoning, where conclusions are used in the assumptions for deriving further conclusions. Furthermore, the task of finding pros and cons may be decomposed recursively. Thus, counterarguments may be identified that conflict with the assumptions of an argument.

In this chapter, we provide an informal coverage of the nature of argumentation. For this, we consider some definitions for basic concepts for argumentation, for the kinds of information used in argumentation, and for the kinds of agent involved in argumentation. During the course of the chapter, we will provide motivation of some of the key elements of argumentation that we plan to formalize in the rest of this book.

1.1 Basic Concepts for Argumentation

We start by providing some simple informal definitions for argumentation. We will expand on these definitions in this chapter, and then, in subsequent chapters, we will explore formal definitions for these concepts.

Argument An argument is a set of assumptions (i.e., information from which conclusions can be drawn), together with a conclusion that can be obtained by one or more reasoning steps (i.e., steps of deduction). The assumptions used are called the **support** (or, equivalently, the **premises**) of the argument, and its conclusion (singled out from many possible ones) is called the **claim** (or, equivalently, the **consequent** or the **conclusion**) of the argument. The support of an argument provides the reason (or, equivalently, **justification**) for the claim of the argument.

Contradiction One formula contradicts another formula if and only if the first negates the second. In other words, two formulae contradict if and only if they are mutually inconsistent. For example, using classical logic, if we have a claim $\alpha \vee \beta$, then the claim $\neg\alpha \wedge \neg\beta$ negates it. Similarly, a formula α contradicts a set of formulae Γ iff $\Gamma \cup \{\alpha\}$ is inconsistent. For example, using classical logic, $\alpha \vee \beta$ contradicts Γ when Γ is $\{\neg\beta, \alpha \rightarrow \beta\}$.

Rebutting argument A rebutting argument is an argument with a claim that is the negation of the claim of another argument. In other words, if an argument states that β holds, a rebutting argument takes the position that the negation of β holds, hence rebutting the argument for β. Thus, an argument A_1 that rebuts another A_2 (so A_1 is a rebutting argument) is such that the claim of A_1 contradicts the claim of A_2. For example, using classical logic, if A_1 has the claim α, and A_2 has the claim $\neg\alpha$, then A_1 and A_2 rebut each other.

Undercutting argument An undercutting argument is an argument with a claim that contradicts some of the assumptions of another argument. Assuming classical logic, suppose an argument has a support that includes the information that β holds, and the information that $\beta \to \alpha$ holds, and the claim that α holds, then an example of an undercutting argument would be an argument with a claim that is the negation of β (i.e., $\neg\beta$) or the negation of $\beta \to \alpha$ (i.e., $\neg(\beta \to \alpha)$).

Counterargument Given an argument A_1, a counterargument is an argument A_2 such that either A_2 is a rebutting argument for A_1 or A_2 is an undercutting argument for A_1.

Argumentation This is the process by which arguments and counterarguments are constructed and handled. Handling arguments may involve comparing arguments, evaluating them in some respects, and judging a constellation of arguments and counterarguments to consider whether any of them are warranted according to some principled criterion.

For argumentation, we may also assume that each argument has a **proponent**, who is the person (or group of people) putting forward the argument, and that each argument has an **audience**, who is the person (or group of people) intended as the recipient(s) of the argument. To illustrate these concepts, consider the following example.

Example 1.1.1 Consider two people, Charlie and James, working in a newspaper office. Charlie makes the following argument to James. So Charlie is the proponent of the argument and James is the audience of the argument.

Claim We can publicize that Simon Jones is having an affair.

Support Simon Jones is a public person, so we can publicize details about his private life.

In response, James makes the following counterargument to Charlie. So James is the proponent of the argument, and Charlie is the audience.

Claim Simon Jones is no longer a public person.

Support Simon Jones just resigned from the House of Commons; hence, he is no longer a public person.

In order to investigate examples such as the above, we can start with a significant development by Stephen Toulmin [Tou58]. For this, Toulmin identifies the importance of a layout for an argument. He shows that to analyze an argument, it is necessary to identify the key components of the information in terms of the roles played within the argument. These components are summarized as follows:

Facts The term "fact" is used by different authors in different ways. Here we assume a fact is an item of information that is specific to a given context. For example, consider a doctor advising a patient. Facts are information on a given patient, such as *name*, *age*, and *blood pressure*. This information is only applicable to that patient. This contrasts with knowledge, in the form of perhaps defeasible rules, that can be used on all patients, such as *If a patient has high blood pressure and is middle-aged, then prescribe a low sodium diet*.

Warrant This is the part of the argument that relates facts to qualified claims. A warrant captures a form of defeasible rule (a rule that is normally valid, when the required facts hold, but in exceptional circumstances, it may fail to hold): Essentially, it says that if the required conditions (represented by the facts) hold, then there is a reason to accept the qualified claim. For this setup, we can regard the facts plus the warrant as the support for an argument.

Backing A backing is some kind of justification for a warrant. It provides an explanation for why the warrant is a reason to accept the qualified claim. Justifications may be based on diverse criteria such as belief, law, authority, ethics, morals, or aesthetics.

Rebuttal A rebuttal captures the circumstances that would be regarded as exceptions for a warrant. In other words, it captures the reasons that would render the warrant as not holding. Thus, if the facts for the rebuttal hold, then we have a rebutting argument, and hence a counterargument, for the argument based on the warrant.

Qualified claim A qualified claim is a conclusion that can be drawn if the warrant holds and the rebuttal does not hold. In a sense, the facts plus the warrant imply the claim.

An example of an argument, conforming to Toulmin's layout, is given in figure 1.1. Here, we see that from the fact that *Harry was born in Bermuda*, we have the qualified claim that *Harry is a British subject*, unless *both his parents were aliens* or *he has become a naturalized American* or etc.

In a sense, Toulmin's approach to argumentation in terms of the layout of arguments is analogous to using a classification system or decision tree: Given some facts, we can decide whether the qualified claim holds by checking whether the facts and warrant hold and the rebuttal does not hold. The approach is structural and, in a sense, logical. However, it lacks mechanisms for constructing and manipulating the graphs. It is,

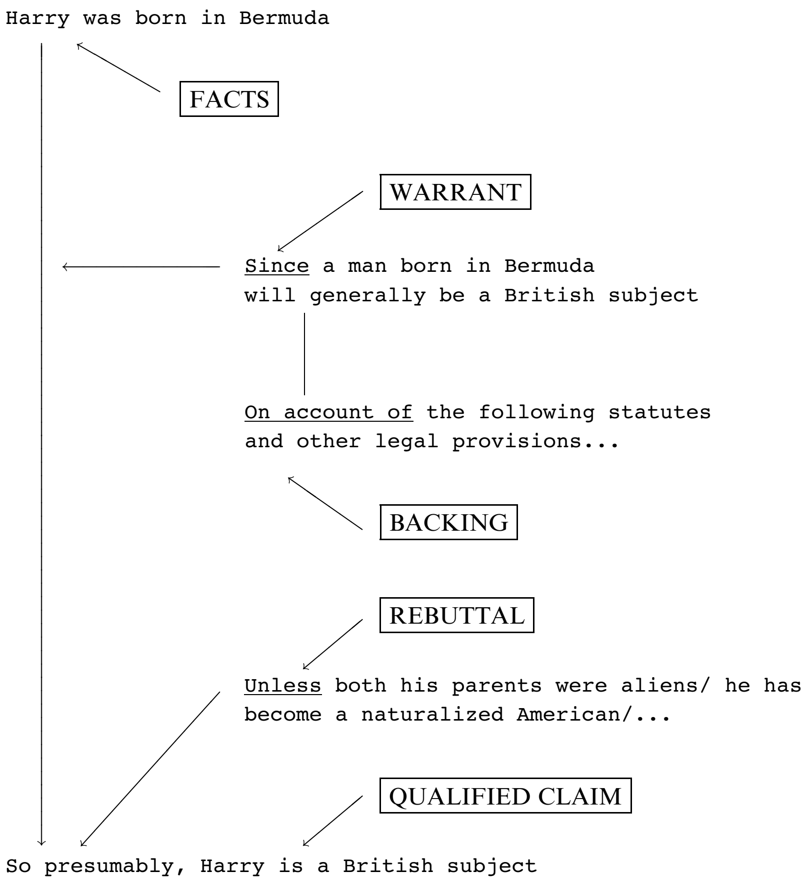


Figure 1.1
An example of argumentation taken from [Tou58].

in a sense, static or fixed. Furthermore, it is text-based, and so it requires some interpretation to use it. This makes it difficult to automate reasoning with it.

To illustrate shortcomings of Toulmin's proposal if we want to use automated reasoning, suppose we have some individual *Mr. Jones*, and we want to determine whether *Mr. Jones is a British subject*; then we need to determine whether *Mr. Jones was born in Bermuda* and whether *both his parents were aliens* or *he has become a naturalized American* or etc. This is not undertaken in any formal language; it is done in natural language and therefore is subject to the problems of ambiguity that arise with the usage of natural language in computing.

For example, suppose we have the fact that *Mr. Jones was born in France*; in order to automate the reasoning that would prohibit the usage of this particular warrant for *Mr. Jones*, we would need to state explicitly that *France is not part of Bermuda*. Of course, this is trivial for a human, but for computing it may require much commonsense knowledge being explicitly specified, and we would need some knowledge representation and reasoning formalism for capturing and automating the use of this commonsense knowledge. As another example, now suppose instead we have the information that *Mr. Jones was born in Hamilton*; again, we need commonsense knowledge and automated reasoning that would allow the usage of the warrant for the individual *Mr. Jones*.

Thus, Toulmin's layout of arguments gives us some important concepts, including warrant, backing, rebuttal, and qualified claim, for describing arguments. We can see these concepts featuring in numerous examples. However, the approach does not just provide a comprehensive account of the logic of argumentation, and furthermore, the approach does not address many important questions on how to automate the construction or use of layouts of arguments. Nonetheless, it would be reasonable to suggest that Toulmin's layout of arguments is an antecedent to many formal approaches to argumentation in artificial intelligence, though many of the formal proposals deviate significantly from this starting point.

1.2 Information Involved in Argumentation

At the core of argumentation is the need for information. If we have no information, we have no arguments, except perhaps tautologies.

Potentially, argumentation can be based on any kind of information. In the following, we consider some informal delineations of types of information. Our aim in presenting these classifications of information is solely to support our presentation of argumentation. We do not wish to suggest that these classifications are contributions in their own right.

Information can be described as being either certain or uncertain as delineated in the following descriptions:

Certain (or categorical) information This is information that is treated as absolutely correct. It is straightforward to treat as certain a mathematical definition or commonly-known knowledge like *the capital of France is Paris*. However, a broader range of examples includes information where the possibility of doubt is so small that the information can be regarded as certain, like *tomorrow the sun will rise*.

Uncertain information This is information that is not certain. Most information is uncertain to some degree. Deciding on whether information is certain or uncertain often depends on the circumstances in which the information is used and evaluated—for example, *the length of the banana is 15 cm*, *it will rain in New York tomorrow*, and *Mr. Jones has had a mild infarction*.

Deciding whether information is certain or uncertain can depend on the application. For example, it is reasonable to assume that *every day the sun will rise* is certain, but there is a minuscule chance of a cosmic catastrophe that would result in the sun's failing to rise tomorrow. Thus, for example, if we consider going for a walk in town tomorrow, it is reasonable to assume that *the sun will rise tomorrow* (though it may be obscured by clouds), but if we consider the future of the universe, we may consider that it is not certain that the sun will rise on any particular day in the future.

We do not need to get sidetracked here into the nature of uncertainty. Rather, we suggest that the reader may wish to adapt the definitions for these terms according to their needs. For more information on the nature of uncertainty see [KC93, PH98].

Information (both certain and uncertain) can also be described as being one of objective, subjective, or hypothetical, as follows:

Objective information This is information that comes from a "reliable source" or can be observed, measured, or verified by everyone involved in the argumentation. For example, consider *a clinical trial for a new drug treatment where 90% of the group of patients with the new treatment survive after five years and 20% of the group of patients with the control treatment survive after five years*. This is objective information. However, just because information is objective, this does not mean that it is necessarily correct or consistent. Errors, and so forth, can occur in obtaining objective information. A possibility for a major error could be that the way the patients were selected for the trial might unknowingly allow for a selection that would respond very well to the treatment, whereas in a wider population of patients with the disease, the success rate may be substantially lower. Thus, if the above value of 90% was used for the population in general, it would be erroneous. In general, inconsistency in information has a diverse etiology, but in objective information it often arises from errors. Consider, for example, two honest witnesses to a bank robbery: Overall, they may give useful information on the event, but they may give quite conflicting descriptions of the getaway car.

Subjective information This is information that comes as beliefs or opinions from some of those involved in the argumentation. This is not necessarily consistent information. An example of subjective information may arise when an oncologist is advising a patient on options for treatment plans. Here, the oncologist may present arguments for and against some options involving combinations of radiotherapy, chemotherapy, and surgery. Much of the information used by the oncologist in the supports for the arguments will be objective, for example, scientific knowledge about the relative efficacies of treatments, but it may also involve preferences expressed by the patient about different drugs or about the weight the patient would put on quality of life over overall survival chances.

Hypothetical information This is information that is assumed for the sake of constructing arguments of interest. It is not necessary for hypothetical information to be true. It may even be such that it is unlikely to be true now or in the future. It may still be useful to consider it as part of the assumptions for argumentation if one wants to explore possibilities. Thus, some hypothetical information may be described as speculative information. For example, it is unlikely that the sea level will rise by 50 cm in the next twenty years, but it would be useful for a government to consider the possibility in their coastal areas in order to consider whether or not they are adequately prepared for flooding. This could be done by assuming the hypothetical information that *the sea level will rise by 50 cm in 20 years*, and then arguments could be constructed for and against the conclusion that they are adequately prepared. As another example, the government may wish to extend this civil emergency planning to consider the hypothetical information that *a UFO will land in the country next week*. Hypothetical information may even include information that the proponent regards as false (i.e., the proponent regards it as fallacious information). Consider, for example, how a sophist may construct an argument for a claim of interest.

Deciding whether information is objective, subjective, or hypothetical can also depend on the application. Again, we do not need to get sidetracked here into more precise definitions for these categories. We present these categories only to indicate the range of situations for which we may wish to formalize argumentation. There are also other dimensions that we could consider for describing information (including epistemics, deontics, vagueness, and particular kinds of uncertainty such as probability and possibility) based on developments in the knowledge representation and reasoning literature.

1.3 Agents Involved in Argumentation

Central to conceptualizing argumentation is that argumentation involves agents and groups of agents. This is for considering both the proponent and the audience of each argument.

First, we will delineate a notion of agent and entity, and then we will define the notions of monological argumentation and dialogical argumentation in terms of agents and entities involved. To support this goal, we will adopt the following informal definitions (for a comprehensive and formal conceptualization of agents and associated notions, see [Woo01]):

Agent An agent is an autonomous, proactive, and intelligent system that has some role. Examples of kinds of agent include lawyers, clinicians, and journalists. Further examples of types of agent include voters in an election, readers of a newspaper, jury members, and patients in a hospital. We may wish to also think of some software systems as agents if they display sufficiently significant intelligence, autonomy, and proactiveness.

Entity An entity is composed of a set of agents that in concert have some role. A simple example of an entity is a board of directors for a company, where each agent in the entity is a director. The agents in an entity may be heterogeneous. In other words, different agents in an entity may have different roles. For example, a court is an entity that is composed of a judge, a prosecution lawyer, a defense lawyer, witnesses, a defendant, and jury members. These are agents with different roles, and in concert they have the role of conducting a trial of the defendant. Another example of an entity is an audience for a political speech. Here the audience may be composed of agents who each have a political standpoint, and so in this case the role of the entity is only to be an audience to the political speech. A third example of an entity is a group of scientists, working on a research project, who publish a scientific paper.

Thinking of argumentation in terms of agents allows us to formalize the different roles that agents can play in different kinds of argumentation. In order to further delineate our concerns in this book, we need to briefly describe the monological and the dialogical views on argumentation:

Monological A single agent or entity has collated the knowledge to construct arguments for and against a particular conclusion. This involves collating both categorical and uncertain information. Furthermore, this may include objective information (e.g., externally measured or verifiable information, information obtained from reliable third-party sources, etc.), subjective information (e.g., beliefs, aesthetics, etc.), and hypothetical information. The knowledge may come from heterogeneous sources. After constructing the arguments, the entity may then draw some conclusion on the basis of the assembled arguments. The emphasis of the monological view is on how to construct the arguments and how to draw conclusions from the assembled arguments. Monological argumentation can be viewed as an internal process for an agent or an entity with perhaps a tangible output (e.g., an article or a speech or a decision). In monological argumentation, there is no representation of the dialogue between the agents or entities involved. However, the knowledge used to construct the support for one or more arguments may have been obtained from a dialogue.

Dialogical A set of entities or agents interact to construct arguments for and against a particular claim. If an agent offers an argument, one or more of the other agents may dispute the argument. Agents may use strategies to persuade the other agents to draw some conclusion on the basis of the assembled arguments. The emphasis of the dialogical view is on the nature of the interactions and on the process of building up the set of arguments until the agents collectively reach a conclusion. Dialogical argumentation can be viewed as incorporating monological argumentation, but in addition, dialogical argumentation involves representing and managing the locutions exchanged between the agents/entities involved in the argumentation.

In a sense, monological argumentation is a static form of argumentation. It captures the net result of collating and analyzing some conflicting information. In contrast, dialogical argumentation is a dynamic form of argumentation that captures the intermediate stages of exchanges in the dialogue(s) between the agents and/or entities involved.

Nonetheless, monological and dialogical argumentation involve a proponent and an audience. Some agent or entity provides each argument, and some agent or entity is the intended audience for that argument—though, of course, the proponent and audience for an argument may be the same agent or entity, particularly in the case of monological argumentation.

To illustrate the difference between monological and dialogical argumentation, we consider some examples. For monological argumentation, we list some situations for static argumentation and the kinds of agent or entity that are responsible for producing that argumentation:

- A newspaper article by a journalist.
- A political speech by a politician.
- A political manifesto by a political party.
- A review article by a scientist.

For dialogical argumentation, we list some situations for dynamic argumentation and the kinds of agent or entity that are responsible for that argumentation:

- Lawyers arguing in a court.
- Traders negotiating in a marketplace.
- Politicians debating about new legislation.
- Governments negotiating a new world trade agreement.
- Family members arguing over who should do the washing up.

Ultimately, both monological and dialogical argumentation aim for some final result from the process, but in monological argumentation, the emphasis is on the final result, whereas in dialogical argumentation, the emphasis is the process as represented in terms of dialogue exchanges. This obviously has important ramifications for formalizing argumentation. To formalize dialogical argumentation, a lot of extra machinery is required to model or automate the role of the agents involved.

It is clear that there are a variety of roles for monological argumentation depending on the kind of information used and on the aim of the presenter of the arguments. The following breakdown is only meant to indicate the diversity of roles for monological argumentation; it is not meant to be a "definitive classification."

Factual argumentation Use just objective information with the aim of informing the audience about some verifiable information—for example, a scientific review. Here, we assume that there is no hidden bias in how the argumentation is undertaken.

Positional argumentation Use objective information, subjective information, and hypothetical information with the aim of informing the audience of the presenter's beliefs—for example, a newspaper opinion article.

Persuasional argumentation Use objective information, subjective information, and hypothetical information (including possibly fallacious information) with the aim of persuading the audience to do something—for example, a political speech, a team pep talk, or a sales pitch.

Provocational argumentation Use objective information, subjective information, and hypothetical information (including possibly fallacious information) with the aim of provoking the audience of some hypothetical situations for entertainment, to invoke further thinking, or to map extremes in a space—for example, a newspaper opinion article, a think-tank pamphlet, or an academic article. Provocational argumentation can also be used as entertainment, such as in satire and in sophism.

Speculational argumentation Use objective information, subjective information, and hypothetical information (including speculative information) with the aim of informing the audience about a possible scenario for explaining some past event or some possible future event—for example, a risk management scenario or an academic article.

Monological argumentation has directionality. In other words, when an agent constructs some arguments and counterarguments, there is normally an intended recipient in some sense. The intended audience can range from one particular agent through to the global audience (i.e., anybody).

We regard all monological argumentation as either argumentation by the proponent for the proponent (auto-argumentation), and so the proponent and the intended audience are the same, or argumentation by the proponent for one or more other agents (one-to-many argumentation):

Auto-argumentation This is argumentation for agents(s) to identify key arguments and counterarguments for their own use, such as for problem analysis prior to making a decision. For example, for most of us, when we buy a house, we have a limited budget, a list of features we would like, and a list of features we would dislike. It is often the case that we can narrow the choice down to a few possible houses that are available, and none of them are perfect. Each may lack some of the features we would like, and each may have some features we would dislike. In other words, each of the houses on the shortlist is inconsistent with our requirements. However, because we have to make a choice, we can consider the pros and cons for each possible house. Auto-argumentation could also be called self-argumentation.

One-to-many argumentation This is argumentation by an agent or entity for distribution to other agents or entities—for example, a newspaper article by a journalist, a lecture by a scientist, or a speech by a politician. Of course, one-to-many argumentation does not have to involve professionals. Consider, for example, a child making a case to his or her parents for a higher allowance. One-to-one argumentation (a proponent presenting an argument to an audience of exactly one agent) is a special case of one-to-many argumentation.

Even though monological argumentation is argumentation that involves just one proponent, which may be an agent or an entity, it may summarize information that has come from debates or discussions, as well as other sources of information.

To illustrate this, consider a situation where a clinician and a patient discuss options for an oncology treatment plan, taking into account relevant medical knowledge together with the patient's preferences and personal circumstances. The meeting involving these two agents would involve dialogical argumentation. However, when they have exchanged the necessary information, and perhaps debated some of the major issues of concern, they could then bring together the major points for the patient, highlighting them in the form of the pros and cons of the key options, prior to the patient's making necessary decisions and/or giving the necessary authorization. This representation of the key information gleaned during the meeting in the form of pros and cons is a form of monological argumentation. Furthermore, since the doctor and the patient come together to act as an entity in collating the arguments and counterarguments, it is a form of auto-argumentation.

We will see further examples, in this book, of how we can represent the key information gleaned during a discussion or debate using monological argumentation. We may choose to think of there being a "third agent" who is collecting information from the dialogical argumentation and using it to undertake monological argumentation, without necessarily attributing the gleaned information to any of the sources and without representing any of the history of the dialogue.

1.4 Requirements for Formalizing Argumentation

The overall aim of this book is to present formalizations of aspects of monological argumentation. In so doing, we will formalize key elements of practical argumentation. By practical argumentation, we mean argumentation that reflects more closely argumentation as practiced by agents in the real world.

In our coverage, we will consider how abstract argumentation and logical argumentation provide important foundations for formalizing monological argumentation. We will also see shortcomings in the basic versions of these proposals for capturing practical argumentation. In order to consider these proposals systematically, we now sketch the requirements we have for formalizing practical monological argumentation in this book. As part of presenting solutions to these requirements, during the course of the book, we will be conceptualizing some of the key elements of argumentation:

Presentation of arguments We want to be able to present an exhaustive display of the constellation of arguments and counterarguments relevant to a particular claim. This should act as an inventory of all the different ways that the conclusion can be inferred from the assumptions and all the different ways that counterarguments can be inferred from the assumptions. Given a particular claim and a "knowledgebase" (from which we find the supports for arguments and counterarguments), we want to be able to automate the construction of each constellation.

Analysis of intrinsic factors Given a constellation of arguments and counterarguments relevant to a particular claim, we want to analyze the nature and type of conflicts that arise in the constellation. We also want to be able to annotate the constellation with information about the results of analyzing intrinsic factors.

Analysis of extrinsic factors For a constellation of arguments and counterarguments relevant to a particular claim, we want to analyze the quality of the arguments and counterarguments from the perspective of a representative (or stereotypical) member of the audience. This includes considering how believable the arguments are for the representative and what the impact is for the representative. We also want to be able to annotate the constellation with information about the results of analyzing extrinsic factors.

Selection of arguments Given the ability to undertake analyses of intrinsic and extrinsic factors, we want principled techniques for selectivity in the choice of arguments and counterarguments used in a constellation. The net result is that, as an alternative to an exhaustive presentation of arguments and counterarguments, we obtain a more focused constellation of arguments and counterarguments tailored for the intended audience. Being selective means that the argumentation can be made more believable and have higher impact for the intended audience.

Judgment of constellations For a constellation of arguments and counterarguments relevant to a particular claim, we want principled criteria for suggesting whether the claim is warranted or unwarranted.

Reformation of constellations Given a constellation of arguments and counterarguments relevant to a particular claim, we want principled means for reforming (i.e., restructuring) the arguments by, for example, merging arguments with logically equivalent supports.

We want to be able to present an exhaustive display of arguments and counterarguments relevant to the conclusion as output because we want the user to decide what to do with the information. We do not want to develop a black box for outputting conclusions; rather, we want to output the key arguments and highlight the key conflicts. If we consider some of the kinds of monological argumentation that we are interested in capturing, such as newspaper articles, political speeches, and scientific research papers, it is clear that the information assumed, and the way it is put together, is as important as, if not more important than, the conclusion obtained.

However, there is also normally the need for arguments to be apposite for the intended audience. Consider an article in a current affairs magazine: Only a small subset of all possible arguments that the journalist could construct from his or her own knowledgebase is used. The journalist regards some arguments as having higher impact or as being more believable for the intended audience or more relevant than others and so makes a selection. This need for appositeness is reflected in law, medicine, science, politics, advertising, management, and just ordinary everyday life.

Thus, taking the audience into account means that there has to be some selectivity of the arguments presented to them. Numerous formal theories of argumentation exercise selectivity on grounds of certainty and preference as viewed from the presenter's perspective, but the audience's perspective is largely ignored. We want to formalize these in terms of knowledge about the audience. We will argue that being selective in argumentation improves the constellation of arguments and counterarguments by making it more interesting and more believable.

We can think of the requirements giving the user a range of options. The user of an argumentation system can choose to have an exhaustive display of a constellation of arguments and counterarguments, or the user can choose to have a selective display of a constellation of arguments and counterarguments, based on a particular audience. In either case, the user can choose to annotate the constellation with information coming from analyzing intrinsic and extrinsic factors arising in the constellation.

1.5 Frameworks for Formalizing Argumentation

If we want to handle arguments systematically, then we need a "formalization" of argumentation. Many professions implicitly or explicitly explore these issues and, indeed, put the systematic use of arguments at the heart of their work. Consider, for example, the legal, medical, and journalistic professions.

However, in this book, we want to go beyond the systematic handling of arguments: We want to handle arguments automatically, and we want the techniques to scale up to handling substantial and complex problems. This calls for more detailed formalizations with algorithms. Furthermore, if we want predictable behavior for our argumentation systems, then we need theoretical properties and empirical results. This, in turn, will call for a sophisticated and precise understanding of the principles of argumentation, which, in turn, calls for richer and deeper theoretical formalisms.

Classical logic is appealing as a starting point for argumentation: The representation is rich and the reasoning powerful. Furthermore, it can be argued that classical reasoning captures some of the important ways that people undertake logical reasoning: For example, modus ponens, modus tollens, and disjunctive syllogism. However, the appeal of classical logic extends beyond the naturalness of representation and reasoning. It has some very important and useful properties that mean that it is well-understood and well-behaved and that it is amenable to automated reasoning.

In classical logic, statements are represented by formulae. Both assumptions and conclusions are represented by formulae, and the language for assumptions and conclusions is the same. Let Δ be a set of formulae, let $\vdash$ be the classical consequence relation, and let α be a formula; then $\Delta \vdash \alpha$ denotes that α is an inference (i.e., a conclusion) from Δ using classical logic. In Appendix C, we provide a review of the syntax and semantics for classical logic.

However, there is a key concern if we are to use classical logic for argumentation. We have already acknowledged that argumentation involves considering conflicting (i.e., inconsistent) information. If the knowledge we have for constructing arguments is consistent, then we will not be able to construct conflicting arguments, and hence we will not have recourse to argumentation. Unfortunately, inconsistency causes problems in reasoning with classical logic.

In classical logic, any inference can follow from an inconsistent set of assumptions. A useful definition of inconsistency for a set of assumptions Δ is that if $\Delta \vdash \alpha$ and $\Delta \vdash \neg\alpha$, then Δ is inconsistent. A property of classical logic is that if Δ is inconsistent, then for any β in the language, $\Delta \vdash \beta$. This property results from the following proof rule, called *ex falso quodlibet*, being a valid proof rule of classical logic.

$$\frac{\alpha \qquad \neg\alpha}{\beta}$$

Thus, inconsistency causes classical logic to collapse. No useful reasoning follows from an inconsistent set of premises. It can be described as exploding, or trivialized, in the sense that all formulae of an the language are consequences of an inconsistent set of assumptions. From a semantic perspective, there are no models of a set of inconsistent formulae.

Partly in response to the issue of inconsistency arising in argumentation, there have been three main approaches to formalizations for argumentation, namely, abstract systems, defeasible systems, and coherence systems. The first two approaches use formalisms that are, in key respects, much less expressive (in terms of the complexity of information that can be represented and in the complexity of the inferences that can be drawn) when compared with classical logic, thereby circumventing the problem of inconsistency as manifested by *ex falso quodlibet*, and the third approach adopts a simple strategy to ameliorate the problem of inconsistency. We delineate these three approaches as follows:

Abstract systems These are based on the seminal proposal by Dung [Dun95] that assumes a constellation of arguments and counterarguments can be captured by a set of arguments and a binary "attacks" relation between pairs of arguments. The attacks relation captures the situation where one argument undermines the credibility of another. This setup can be viewed as a graph, with each node representing an argument and each arc representing an "attacks" relationship. Thus, the constellation, represented by the graph, is the starting point. It is not constructed from a knowledgebase. Reasoning with the graph is based on finding coalitions of arguments such as a coalition of arguments that do not attack each other and that attack any argument that attacks any member of the coalition. We review abstract systems in chapter 2.

Defeasible systems There are a number of proposals for defeasible logics. The common feature for these logics is the incorporation of a defeasible implication into the language. Defeasible logics have their origins in philosophy and were originally developed for reasoning problems similar to those addressed by nonmonotonic logics in artificial intelligence. Arguments can then be defined as chains of reasons leading to a conclusion with consideration of potential counterarguments at each step. With the explicit structure in the chains of reasoning, diverse notions of defeat can be conceptualized. Once nonclassical notions of implication are introduced into the language, giving rise to either subclassical systems (i.e., systems weaker than classical logic) or superclassical systems (i.e., systems stronger than classical logic), an interesting range of issues arise for refining the notion of an argument, a counterargument, an undercut, a rebut, and so on. A number of these defeasible systems construct arguments logically and then evaluate sets of them as an abstract system (each logical argument is a node in the graph, and if an argument rebuts or undercuts another, then this is represented by an arc in the graph). In this way, a defeasible system can "instantiate" an abstract system, or equivalently, the abstract system provides a "semantics" for the defeasible system. We review some of these issues in chapter 8.

Coherence systems One of the most obvious strategies for handling inconsistency in a knowledgebase is to reason with coherent (i.e., consistent) subsets of the knowledgebase. This is closely related to the approach of removing information from the knowledgebase that is causing an inconsistency. In coherence systems, an argument is based on a consistent subset of an inconsistent set of formulae—the inconsistency arises from the conflicting views being represented. Further constraints, such as minimality or skeptical reasoning, can be imposed on the consistent subset for it to be the support for an argument. The most common choice of underlying logic for coherence systems is classical logic, though other logics such as modal, temporal, spatial, or description logics are possible. While coherence systems, based on classical logic, give substantial expressibility in order to capture a wide range of monological argumentation situations, there may be computational and applicational advantages of using argumentation systems based on simpler defeasible logics.

We will cover abstract systems in detail in chapter 2, and we will cover defeasible systems in detail in chapter 8. The main framework that we will present in this book can be regarded as a coherence system. We will introduce this in chapter 3 and develop it during subsequent chapters. In this way, we will use the logic-based approach presented in chapter 3 as the vehicle to isolate key elements of argumentation during the subsequent chapters, and we believe that many of these elements can be seen in all logic-based approaches to argumentation.

1.6 Discussion

In this chapter, we have attempted to delineate some of the basic concepts that are part of argumentation. In addition, we have considered our focus in this book on formalizing monological argumentation. This type of argumentation is central to other forms of argumentation. It is an interesting subject of study in its own right, and it offers much potential as part of technological solutions for decision support, multi-agent systems, and computational linguistics.

For the logic-based frameworks we present in this book, we are assuming that the input for a system based on monological argumentation is a knowledgebase, together with a claim of interest, and the output is a constellation of arguments and counterarguments. This constellation may have been subject to selectivity according to analysis based on intrinsic and/or extrinsic factors, and it may have been annotated with information on some of these analyses. Via the presentation of this framework, we aim to delineate some of the key elements of argumentation.

In section 1.4, we sketched some of the key requirements for formalizing practical argumentation, by which we mean we have delineated some of the key features of argumentation undertaken by real-world agents with the aim of being able to capture these features in a logic-based setting. In order to investigate the issues surrounding this aim, we have decoupled practical argumentation from the wider aims of an agent, such as planning and acting in the real world. Though, as we suggest later, and as other authors have recognized (e.g., [AP05a, AP05b]), we will need to consider these wider issues to more fully understand practical argumentation.

1.7 Bibliographic Notes

Numerous textbooks have explained how individual arguments, as found in the real world, can be represented and analyzed by classical logic. The focus of these books is on what constitutes a valid logical argument. A paragon of such a textbook by Fisher [Fis88] includes many examples of arguments originally presented in philosophy and in politics using free text, together with a comprehensive explanation of how they can be translated into propositional logic. However, these textbooks tend to circumvent the more difficult issues of inconsistency, conflict, and counterarguments.

To address these more difficult issues, an excellent starting point for considering monological argumentation is Toulmin's book [Tou58], with developments of Toulmin's approach having been reviewed by van Eemeren et al. [vGK87]. Toulmin's work is regarded as a precursor to much of the formal developments in argumentation systems found in the artificial intelligence field. For reviews of formalisms for argumentation systems in artificial intelligence, see [PV02, CML00].

While much interesting progress has been made in recent years on formalizing argumentation, it is clear that there are many more difficult features of argumentation that remain to be captured in some way. Within the philosophy community, there has been a line of research called "informal logic" that has been studying the nature of argumentation, and this has resulted in a number of valuable conceptualizations for the development of argumentation in artificial intelligence (see, e.g., [Wal89]).

More generally within philosophy, there has long been an interest in argumentation (for a review from a historical and sociological perspective, see [Bil87]). The initiation of the study of formal deductive reasoning is often attributed to Aristotle. He can also be said to have initiated the study of rhetoric, which in part considers how a proponent should take the audience into account when presenting arguments. While we do not wish to get sidetracked into a review of the study of rhetoric, more recently, Perelman [Per82] reworked the philosophy of rhetoric in a way that offers some valuable insights into the importance and nature of audiences for argumentation. We view Perelman's work as providing valuable background and motivation for some of the formal developments we present in this book for taking the audience into account. Still more recent background for taking the audience into account is presented in Cockcroft and Cockcroft [CC92] and Hollihan and Baaske [HB05].

2 Abstract Argumentation

2.1 Introduction

The simplest way to formalize a collection (i.e., a constellation) of arguments consists of just naming arguments (so, in a sense, not describing them at all) and merely representing the fact that an argument is challenged by another (and so not indicating what the nature of the challenge is). In other words, a collection of arguments can be formalized as a directed binary graph [Dun95]. This provides a very simple and illuminating starting point for considering how to formalize monological argumentation.

Definition 2.1.1 An **argument framework** is a pair $(\mathfrak{A}, \mathscr{R})$ where $\mathfrak{A}$ is a set and $\mathscr{R}$ is a binary relation over $\mathfrak{A}$ (in symbols, $\mathscr{R} \subseteq \mathfrak{A} \times \mathfrak{A}$).

Each element $a \in \mathfrak{A}$ is called an **argument** and $a\mathscr{R}b$ means that a **attacks** b (accordingly, a is said to be an **attacker** of b). Thus, a is a **counterargument** for b when $a\mathscr{R}b$ holds.

Such a formalization is abstract because both the nature of the arguments and the nature of the attack relation are ignored. In particular, the internal (logical) structure of each of the arguments is not made explicit, and accordingly it is not indicated whether an attack is a contradiction between claims, for example, or some other form of conflict. Also, there is no indication whatsoever that the nature of the attack can vary depending on the arguments at hand. In other words, there is no mechanism in this setup to allow arguments to be context sensitive.

An argument framework can simply be identified with a directed graph whose vertices are the arguments listed by $\mathfrak{A}$ and edges correspond to the pairs in $\mathscr{R}$. An example of an argument framework represented as a directed graph is given in figure 2.1.

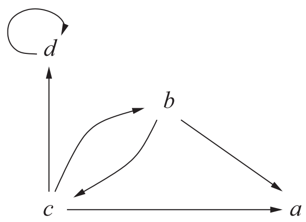


Figure 2.1
A small but complicated illustration of an argument framework that includes an argument attacking itself.

Example 2.1.1 Consider the following two arguments that may arise in the office of a newspaper editor (cf. example 1.1.1):

(a) Simon Jones is a public person, so we can publish an article that includes even details about his private life.

(b) But Simon Jones just resigned from the House of Commons; hence, he no longer is a public person.

These arguments can be represented, in a sense, in the argument framework represented by the following graph.

$b \longrightarrow a$

This graph provides a concise and clear summary of the information we have about this discussion: It says that argument a is attacked by argument b.

No formal property is required for the attack relation. Specifically, it need not be irreflexive, antisymmetric, or antitransitive. This liberal approach raises issues about the meaning of the knowledge representation (e.g., what is an argument that attacks itself?) as is illustrated by argument d in figure 2.1. To consider this question, we start by recalling the properties of a binary relation R being irreflexive, antisymmetric, and antitransitive, as follows:

- A relation R is irreflexive iff $\forall x.\neg R(x,x)$.
- A relation R is antisymmetric iff $\forall x \forall y.R(x,y) \wedge R(y,x) \rightarrow x = y$.
- A relation R is antitransitive iff $\forall x \forall y \forall z.(\neg R(x,y) \vee \neg R(y,z) \vee \neg R(x,z))$.

It then follows that we can make the following observations on the attacks relation $\mathscr{R}$:

- $\mathscr{R}$ fails to be irreflexive iff $\exists x.\mathscr{R}(x,x)$.
- $\mathscr{R}$ fails to be antisymmetric iff $\exists x\exists y.\mathscr{R}(x,y) \wedge \mathscr{R}(y,x) \wedge x \neq y$.
- $\mathscr{R}$ fails to be antitransitive iff $\exists x\exists y\exists z.\mathscr{R}(x,y) \wedge \mathscr{R}(y,z) \wedge \mathscr{R}(x,z)$.

When the attack relation fails to be antitransitive, there exists an argument such as c in figure 2.1 where c attacks both a and b while b attacks a. An argument such as c is called a (directly) **controversial** argument. It is not trivial to demonstrate that controversial arguments make sense. However, they do occur, as illustrated by the following example.

Example 2.1.2 Consider the following conversation between three teenagers.

(a) *Allyson:* She locked the door on purpose because she hates me.

(b) *Beverly:* She likes you; she volunteered to help you move in.

(c) *Christin:* She didn't volunteer; she had to because of Dad, who also told her to lock the door.

Here, c is controversial: b attacks a but c attacks both a and b.

Importantly, a controversial argument need *not* determine which *claim is right* (as the example shows: Christin does not say whether the other girl likes Allyson, ...).

More generally, if c attacks both b and a while b attacks a, perhaps a rational analysis may show that both a and b can be objected to. This is why controversial arguments are meaningful.

When the attack relation fails to be antisymmetric, there exist two arguments, such as b and c in figure 2.1, that attack one another. Actually, the case of two arguments attacking each other (via opposite claims) makes sense. It reflects the notion of rebuttal that we introduced in chapter 1.

Example 2.1.3 Richard is an American; he happens to be a Quaker and a Republican. Then, it is possible to find two arguments that attack each other:

1. Richard is a Quaker and Quakers are pacifist, so he is a pacifist.
2. Richard is a Republican and Republicans are not pacifist, so he is not a pacifist.

When the attack relation fails to be antireflexive, at least one argument attacks itself: It is a **self-attacking** argument. The matter of self-attacking arguments is not easily settled. Authors in the literature attempt to adapt the liar paradox, but the result fails to make the point. Here is another example exhibiting a genuine self-attacking argument:

Protagoras, an ancient Greek master teaching law, meets with Euathlus, who cannot afford the tuition fees. Protagoras agrees that Euathlus does not pay until he is finished with his course. Then, should Euathlus win his first trial, he pays Protagoras. Should Euathlus lose his first trial, he need not pay Protagoras. Later, Euathlus is finished with learning law from Protagoras. Time passes by and Euathlus is not taking any case. Protagoras gets impatient and takes Euathlus to court, asking him to pay. Euathlus argues as follows:

"Even if I lose the trial, I need not pay (according to our initial agreement)."

There seems to be some self-deception here, as the reason in the argument, loss of the trial, comes with the fact that Euathlus has to pay (it's what the trial is about).

2.2 Acceptability

Abstract argumentation is concerned with ascribing arguments a status, which amounts to determining whether some arguments of interest can be accepted (in other words, whether some arguments of interest can be evaluated as undefeated in some sense) or, on the contrary, refuted (in other words, defeated). This is, of course, relative to a given argument framework, which is implicitly assumed in all definitions throughout the chapter.

The first step to be taken is to extend the attack relation to sets of arguments in the obvious way.

Definition 2.2.1 A set $S \subseteq \mathfrak{A}$ of arguments **attacks** an argument $a \in \mathfrak{A}$ if some argument in S attacks a.

Definition 2.2.2 An argument $a \in \mathfrak{A}$ is **acceptable** with respect to a set $S \subseteq \mathfrak{A}$ of arguments iff for each argument $b \in \mathfrak{A}$, if b attacks a then S attacks b.

Alternatively, it is also said that S **defends** a when applying definition 2.2.2.

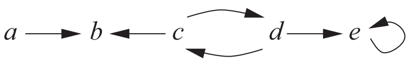
Definition 2.2.3 A set $S \subseteq \mathfrak{A}$ of arguments is **conflict free** iff there are no a and b in S such that a attacks b.

The simplest case of arguments that can be accepted is as follows.

Definition 2.2.4 A set $S \subseteq \mathfrak{A}$ of arguments is **admissible** iff S is conflict free and defends all its elements.

The intuition here is that for a set of arguments to be accepted, we require that, if any one of them is challenged by a counterargument, then they offer grounds to challenge, in turn, the counterargument.

Example 2.2.1 Consider the following argument framework $(\mathfrak{A}, R)$



The admissible sets of $(\mathfrak{A}, R)$ are the following:

$\{\emptyset, \{a\}, \{c\}, \{d\}, \{a, c\}, \{a, d\}\}$

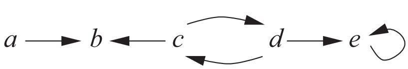
The result below asserts that admissible sets always exist. It means that, given an argument framework, an admissible set exists: As already indicated, and similarly for all formal statements throughout the chapter, the property is not absolute but relative to a given argument framework.

Proposition 2.1.1 ([Dun95]) There always exists at least one admissible set: The empty set is always admissible.

Clearly, the notion of admissible sets of arguments is the minimum requirement for a set of arguments to be accepted. For instance, a is not attacked at all in example 2.2.1, and it may therefore seem odd that some admissible sets such as $\{c\}$ need not contain a. Hence, we require the more demanding notion of an extension as defined next.

Definition 2.2.5 A set $S \subseteq \mathfrak{A}$ of arguments is a **complete extension** iff S is an admissible set such that each argument that S defends is in S.

Example 2.2.2 Let $(\mathfrak{A}, \mathscr{R})$ be as in example 2.2.1:



$$(\mathfrak{A}, \mathscr{R}) \text{ has three complete extensions: } \begin{cases} \{a\} \\ \{a, c\} \\ \{a, d\} \end{cases}$$

Thus, a complete extension is a set of arguments such that (1) it is conflict free, and (2) it consists of *all and only* the arguments that it defends.

Proposition 2.2.2 ([Dun95]) Every complete extension is an admissible set.

Example 2.2.1 shows that the converse is not true.

It may now seem odd that both c and d can be omitted in a set of accepted arguments, namely, the complete extension $\{a\}$ of example 2.2.2 —so, what about defining extensions that are maximal?

Definition 2.2.6 A **preferred extension** is a $\subseteq$-maximal admissible subset of $\mathfrak{A}$.

Example 2.2.3 Again, $(\mathfrak{A}, \mathscr{R})$ is as in example 2.2.1:

$$\begin{tikzcd} a \arrow[r] & b & c \arrow[l] \arrow[r, bend left] & d \arrow[l, bend left] \arrow[r] & e \arrow[loop right] \end{tikzcd}$$

$(\mathfrak{A}, \mathscr{R})$ has two preferred extensions: $\begin{cases} \{a, c\} \\ \{a, d\} \end{cases}$

Thus, a preferred extension is a set of arguments such that (1) it is conflict free, (2) it defends all its elements, and (3) no more arguments can be conjoined to it because *either* the extended set would contain a conflict *or* one of the extra arguments would not be defended by the extended set. Note, though, that it may be possible to extend the set by *adjoining several arguments at once!* (An illustration is the case where the argument framework in example 2.2.3 also includes f, g, h, and i such that $f\mathscr{R}g$, $g\mathscr{R}h$, $h\mathscr{R}i$, and $i\mathscr{R}f$; then, none of f, g, h, and i can be *individually* conjoined to $\{a, c\}$, for example, although $\{f, h\}$ or $\{g, i\}$ can.)

Theorem 2.2.1 ([Dun95]) There always exists a preferred extension.

Once more, the statement here is relative to a given argument framework. That is, theorem 2.2.1 means that an argument framework has at least one preferred extension.

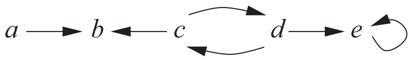
Theorem 2.2.2 ([Dun95]) Every preferred extension is a complete extension.

Example 2.2.2 shows that the converse is untrue.

Regarding a preferred extension as a set of accepted arguments, it may dually be expected that the arguments not in the extension are not accepted, meaning that they all are attacked by arguments in the extension. Then, a notion of an ideally well-behaved extension emerges.

Definition 2.2.7 A set $S \subseteq \mathfrak{A}$ of arguments is a **stable extension** iff S is conflict free and attacks each argument that is not in S.

Example 2.2.4 Consider once more the argument framework $(\mathfrak{A}, \mathscr{R})$ introduced in example 2.2.1:



$(\mathfrak{A}, \mathscr{R})$ has a single stable extension: $\{a, d\}$

Theorem 2.2.3 ([Dun95]) Each stable extension is a preferred extension.

Example 2.2.3 shows that the converse is untrue.

Proposition 2.2.3 ([Dun95]) No stable extension is empty, unless $\mathfrak{A} = \emptyset$.

Theorem 2.2.4 ([Dun95]) A stable extension is a set of arguments $S \subseteq \mathfrak{A}$ such that $S = \{a \in \mathfrak{A} \mid S \text{ does not attack } a\}$.

The drawback with the notion of stable extensions is that not every argument framework has stable extensions.

Example 2.2.5 The argument framework $(\mathfrak{A}, \mathscr{R})$ below has no stable extension:

Notice that $(\mathfrak{A}, \mathscr{R})$ is a single argument framework; it simply has unconnected parts.

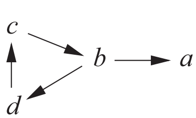
2.3 Cycles

Although argument frameworks range over all binary directed graphs, the possibilities for different kinds of extensions are limited when we consider appropriate properties that we would require of them for them to reflect argumentation intuitively. The main limits lie with cycles.

Definition 2.3.1 A **cycle (of length n)** in $(\mathfrak{A}, \mathscr{R})$ is a finite sequence $e_1, e_2, \ldots, e_n$ of arcs from $\mathscr{R}$ such that

1. The endpoint of e_n is the origin of e_1.
2. For $i = 1 \ldots n-1$, the endpoint of e_i is the origin of e_{i+1}.

Example 2.3.1 Here we consider a dispute concerning the ownership of a valuable painting. The original version of the example is given in [JV00] as a collection of arguments formalized by the following argument framework that contains a cycle:



Ann just bought an old house. She hires Bill to do some repairs. While doing them, Bill finds paintings of great value. Ann and Bill are now at odds about who gets the paintings:

(a) *Ann:* These paintings have been found in my house, so they are mine.

(b) *Bill:* It only is because I tore down the wall that these paintings were found. It is the result of my actions; I found them; this clearly is what counts; who's the landlord is irrelevant.

(c) *Ann:* You found them while working for me as my employee, and so laws on labor must apply to settle the case.

(d) *Bill:* Laws on labor are beside the point; the previous landlord sold you the house but no content such as furniture, rugs, or paintings.

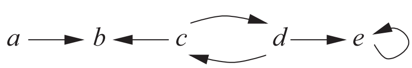
As with many argument frameworks involving cycles, no nonempty extension exists. In this example, it seems intuitive that there is no nonempty extension since it indicates there is "total conflict" between the viewpoints emanating from the two protagonists.

Other illustrations of argument frameworks with odd and even cycles are given in figure 2.2.

The first property relating cycles to the number of (preferred, here) extensions is as follows.

Theorem 2.3.1 ([Dun95]) If $(\mathfrak{A}, \mathscr{R})$ has no cycle of even length, then $(\mathfrak{A}, \mathscr{R})$ has a single preferred extension.

Example 2.3.2 Consider the following argument framework $(\mathfrak{A}, \mathscr{R})$:



$(\mathfrak{A}, \mathscr{R})$ has two preferred extensions: $\begin{cases} \{a, c\} \\ \{a, d\} \end{cases}$

Figure 2.2
An argument framework with an even cycle (left) and an argument framework with an odd cycle (right).

Example 2.3.3 Consider the following argument framework $(\mathfrak{A}, \mathscr{R})$ given in example 2.2.5:

$(\mathfrak{A}, \mathscr{R})$ has a single preferred extension: $\{a\}$.

Corollary 2.3.1 ([Dun95]) Let $(\mathfrak{A}, \mathscr{R})$ be an argument framework with no cycle of even length. If, additionally, each vertex in $\mathfrak{A}$ is the endpoint of an arc in $\mathscr{R}$, then the only preferred extension of $(\mathfrak{A}, \mathscr{R})$ is the empty set.

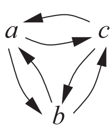
See figure 2.3 for illustrations of argument frameworks with a single preferred extension which is the empty set.

Cycles can also indicate cases where preferred and stable extensions coincide.

Theorem 2.3.2 ([Dun95]) If $(\mathfrak{A}, \mathscr{R})$ has no cycle of odd length, then each preferred extension of $(\mathfrak{A}, \mathscr{R})$ is a stable extension of $(\mathfrak{A}, \mathscr{R})$.

The converse does not hold as illustrated by the following example.

Example 2.3.4 Consider the following argument framework $(\mathfrak{A}, \mathscr{R})$:



The preferred extensions for $(\mathfrak{A}, \mathscr{R})$ are $\{a\}$, $\{b\}$, $\{c\}$, and they are all stable.

In the finite case, cycles are the only cause for an argument framework to have more than one extension of any kind and for the diverse notions of extensions to actually differ from one another.

Theorem 2.3.3 ([Dun95]) If $(\mathfrak{A}, \mathscr{R})$ has no cycle and $\mathfrak{A}$ is finite, then $(\mathfrak{A}, \mathscr{R})$ has a single extension. It is stable, preferred, complete, and grounded (to be introduced in definition 2.4.2 below).

Figure 2.3
Each of the illustrated argument frameworks has exactly one preferred extension, and each of these is the empty set.

Example 2.3.5 Consider the following argument framework $(\mathfrak{A}, \mathscr{R})$:

$$d \longrightarrow c \longrightarrow b \longrightarrow a$$

Obviously, $(\mathfrak{A}, \mathscr{R})$ has no cycle. $\{b, d\}$ is the only extension of $(\mathfrak{A}, \mathscr{R})$. It is stable, preferred, complete, and grounded.

In contrast, here is a counterexample.

Example 2.3.6 In the argument framework $(\mathfrak{A}, \mathscr{R})$ below, each preferred extension is stable but is not unique:

$$\begin{tikzcd} a \arrow[r] & b & c \arrow[l] \arrow[r, bend left] & d \arrow[l, bend left] \arrow[r] & e \end{tikzcd}$$

Indeed, $(\mathfrak{A}, \mathscr{R})$ has more than one preferred extension; it has two: $\{a, c, e\}$ and $\{a, d\}$. Both are stable. Moreover, $\{a\}$ is also a complete extension of $(\mathfrak{A}, \mathscr{R})$, but it is neither preferred nor stable.

The treatment of cycles in an argument framework has been subject to some discussion, and there are some alternative proposals for extensions in the presence of cycles (see, e.g., [KT99, KMD94, BG02]).

2.4 Fixpoints

While the basic definitions for abstract argumentation look simple, there are some significant and complex ramifications of the definitions that have to be considered if we are to have well-behaved and well-understood argumentation systems. Some of these issues can be usefully considered via fixpoint analyses.

2.4.1 Characterizing Extensions

Admissible sets and extensions are conflict-free sets that satisfy some condition(s) concerning the defense of arguments. These two notions can be turned into operations that can serve to define fixpoint equations characterizing admissible sets and extensions. Here is a first technique [Dun95].

Definition 2.4.1 The **characteristic function** of an argument framework $(\mathfrak{A}, \mathscr{R})$ is defined as follows:

$$\mathfrak{F} : 2^{\mathfrak{A}} \to 2^{\mathfrak{A}}$$
$$\mathfrak{F}(S) \stackrel{\text{def}}{=} \{a \in \mathfrak{A} \mid a \text{ is acceptable wrt } S\} \quad \text{for all } S \subseteq \mathfrak{A}.$$

Stated otherwise, $\mathfrak{F}(S)$ is the set of all arguments that S defends.

Proposition 2.4.1 ([Dun95]) A conflict-free $S \subseteq \mathfrak{A}$ is an admissible set iff $S \subseteq \mathfrak{F}(S)$.

Proposition 2.4.2 ([Dun95]) A conflict-free $S \subseteq \mathfrak{A}$ is a complete extension iff S is a fixpoint of $\mathfrak{F}$.

The least fixpoint of $\mathfrak{F}$ is distinguished [Dun95] as another notion of an extension.

Definition 2.4.2 If it exists, the least (with respect to set inclusion) fixpoint of $\mathfrak{F}$ is called the **grounded extension**.

Example 2.4.1 Consider the following argument framework $(\mathfrak{A}, \mathscr{R})$:

$$d \longrightarrow c \longrightarrow b \longrightarrow a$$

$\mathfrak{F}(\emptyset) = \{d\}$
$\mathfrak{F}(\{d\}) = \{b, d\}$
$\mathfrak{F}(\{b, d\}) = \{b, d\}$

The grounded extension is $\{b, d\}$ (it is the least $S \subseteq \mathfrak{A}$ such that $\mathfrak{F}(S) = S$).

The example also illustrates the fact that, if $\mathfrak{A}$ is finite, then the least fixpoint can be reached by iterating the application of $\mathfrak{F}$ from the empty set.

Proposition 2.4.3 ([Dun95]) The grounded extension is the least (with respect to set inclusion) complete extension.

Theorem 2.4.1 ([Dun95]) The grounded extension is the set-theoretic intersection of all complete extensions.

Example 2.4.2 Consider the following argument framework $(\mathfrak{A}, \mathscr{R})$:

$$\begin{tikzcd} a \arrow[r] & b & c \arrow[l] \arrow[r, bend left] & d \arrow[l, bend left] \arrow[r] & e \arrow[loop right] \end{tikzcd}$$

The grounded extension of $(\mathfrak{A}, \mathscr{R})$ is: $\{a\}$

Theorem 2.4.2 ([Dun95]) The grounded extension always exists.

The grounded extension need not be a subset of all admissible sets.

Apart from introducing the notion of the grounded extension, the above fixpoint approach only captures complete extensions. In order to provide a fixpoint approach that additionally captures the other notions of an extension, we need some more notation.

To begin with, the set-theoretic complement of a set of arguments $S \subseteq \mathfrak{A}$ is denoted with a bar:

$$\overline{S} \stackrel{\text{def}}{=} \mathfrak{A} \backslash S$$

Next is the notation about predecessor and successor, borrowed from graph theory, that have been harnessed for a more comprehensive framework for fixpoint analyses of argument frameworks [BD04].

Definition 2.4.3

$$R^+(S) \stackrel{\text{def}}{=} \{a \in \mathfrak{A} \mid \text{some argument in } S \text{ attacks } a\}.$$
$$R^-(S) \stackrel{\text{def}}{=} \{a \in \mathfrak{A} \mid a \text{ attacks some argument in } S\}.$$

Clearly, $R^+(S)$ consists of all arguments that S attacks and $R^-(S)$ consists of all arguments that attack S.

Although $\mathfrak{F}(S) = R^+(\overline{R^+(S)})$ [AC98], $\mathfrak{F}$ is further used in the presence of $R^+(S)$ and $R^-(S)$ as a practical abbreviation.

Now, admissible sets can be characterized in the following way. Recall that an admissible set is a conflict-free set that defends all its elements. Clearly, a set S defends all its elements if and only if $S \subseteq \mathfrak{F}(S)$. Also, S is conflict free if and only if $S \subseteq \overline{R^+(S)}$ (or, equivalently, $S \subseteq \overline{R^-(S)}$). Therefore, S is admissible if and only if $S \subseteq \mathfrak{F}(S) \cap \overline{R^+(S)}$, which is equivalent with $S \subseteq \mathfrak{F}(S) \cap \overline{R^-(S)}$ as well as with two other conditions as summarized in the next theorem.

Theorem 2.4.3 ([BD04]) $S \subseteq \mathfrak{A}$ is an admissible set iff any of the equivalent conditions below hold:

- $S \subseteq \mathfrak{F}(S) \cap \overline{R^+(S)}$
- $S \subseteq \mathfrak{F}(S \cap \overline{R^-(S)})$
- $S \subseteq \mathfrak{F}(S) \cap \overline{R^-(S)}$
- $S \subseteq \mathfrak{F}(S \cap \overline{R^+(S)})$

Theorem 2.4.4 ([BD04]) $S \subseteq \mathfrak{A}$ is a complete extension iff any of the equivalent conditions below hold:

- $S = \mathfrak{F}(S) \cap \overline{R^+(S)}$
- $S = \mathfrak{F}(S \cap \overline{R^-(S)})$
- $S = \mathfrak{F}(S) \cap \overline{R^-(S)}$
- $S = \mathfrak{F}(S \cap \overline{R^+(S)})$

- $S = \mathfrak{F}(S) \cap \overline{R^{+}(S)} \cap \overline{R^{-}(S)}$
- $S \subseteq \mathfrak{F}(S \cap \overline{R^{+}(S)} \cap \overline{R^{-}(S)})$

In [BD04], there are numerous similar but more general results in the following form: $S \subseteq \mathfrak{A}$ is an extension in class $\mathscr{E}$ iff $S = \mathfrak{F}(S \cup X) \cap Y$ iff $S = \mathfrak{F}((S \cup X) \cap Y)$ where the range of X and Y depends on what notion of intermediate extension $\mathscr{E}$ stands for:

$$\left.\begin{array}{l}\mathscr{C}\text{: The set of all complete extensions of } (\mathfrak{A}, \mathscr{R}) \\ \mathscr{S}\text{: The set of all stable extensions of } (\mathfrak{A}, \mathscr{R})\end{array}\right\} \mathscr{S} \subseteq \mathscr{E} \subseteq \mathscr{C}$$

An example of $\mathscr{E}$ is the set of all complete extensions having a maximum of defenders, and so X and Y can be defined for the above fixpoint equation to hold.

2.5 Discussion

Graphs are an appealing approach to formalizing argumentation. An advantage is that any argument can be captured, just because argument frameworks do not take into account the internal structure of an argument:

Pourtant, il convint que toute l'Europe avait les yeux fixés sur l'Empereur. Une parole prononcée aux Tuileries ébranlait les trônes voisins. "C'est un prince qui sait se taire," ajouta-t-il, [...]* (E. Zola, "Son Excellence Eugène Rougon," chapitre 7)

Not distinguishing variations in the way arguments can challenge one another facilitates the task of formalizing a given case of argumentation. One does not have to worry about differences between "defeating" or "rebutting" or "undercutting," etc. Overall, any argumentative situation can be formalized as an argument framework. It is a very general approach—the most general, actually. In addition, the graph-theoretical basis induces many results in the form of various properties about existence and number of extensions.

However, a major drawback of abstract argumentation is that there is a considerable burden on the user since the approach provides no help in finding arguments nor in checking that all arguments relevant to

*He admitted, however, that the eyes of Europe were fixed upon the Emperor. A word spoken at the Tuileries made neighboring thrones tremble. "He is a prince who knows how to hold his tongue," he added [...] (E. Zola, "His Excellency," chapter 7) a particular issue have been identified. Similarly, there is a lack of means to check that all instances of arguments challenging others have been considered.

There are also other disadvantages. One is that having just a single way to formalize the fact that an argument challenges another may result in two cases being given the same formalization despite a contrary intuition:

(a_1) There are 23 people in the room, so there must be at least two who have the same birthday.

(a_2) There are 367 people in the room, so there must be at least two who have the same birthday.

Each of these two arguments attacks this fairly hasty argument:

(a_3) There are no twins among the 23 people in the room, so there are no two people in the room who have the same birthday.

However, it seems rather problematic to avoid the fact that the second argument (a_2) is purely logical and cannot fail, whereas the first may well turn out to support a wrong claim. In fact, probability theory tells us that the first argument (a_1) is roughly as speculative as the third argument (a_3).

We can see a similar problem arising in the following two argumentation frameworks. They have the same structure, but the way they should be assessed and compared should be predominantly based on the content, particularly of the attacking arguments. The attacking argument for the first graph raises issues of personal risk and of believability in the existence of that risk, whereas the attacking argument for the second graph raises issues of flying causing environmental damage and of believability in the existence of that causality. Reducing evaluation and comparison of such examples to the analysis of extensions loses too much of the relevant information associated with the arguments.

Go to Bermuda because there are nice beaches.

$\uparrow$

Don't go to Bermuda because it is dangerous flying through the triangle.

Go to Bermuda because there are nice beaches.

$\uparrow$

Don't go to Bermuda because flying there causes too much environmental damage.

Overall, the main defect of abstract argumentation is that some situations are given the same formalization even though they convey argumentative content that are intuitively dissimilar.

Another disadvantage of abstract argumentation can be put in the following way: Even though an abstract view of the notion of attacks between arguments may intuitively impose it as a nonsymmetric relation, such an intuition is wrong, as some natural symmetric attack relations exist that make the formalization by argument frameworks collapse. Here is an example:

That an argument a attacks an argument a' means that the premises of a contradict (in the sense of classical logic) some subset of the premises of a'.

That is, the premises of an argument a are represented by a set R_a of formulae of classical logic as follows:

$a\mathcal{R}b$ iff $R_a \cup R' \vdash \perp \quad$ for some $R' \subseteq R_b \quad (C)$

Although such an approach makes sense, the unfortunate consequence is the following proposition.

Proposition 2.5.1 In any instance of an argument framework where (C) holds, every argument is acceptable with respect to itself, and therefore every conflict-free set of arguments is admissible.

As a proof, it is enough to show that a is acceptable wrt $S \cup \{a\}$. So, consider $b \in \mathfrak{A}$ such that $b\mathcal{R}a$. That is, $R_b \cup R'' \vdash \perp$ for some $R'' \subseteq R_a$. Then, $R_b \cup R_a \vdash \perp$ holds. Therefore, $R_a \cup R' \vdash \perp$ for $R' = R_b$. Hence, $a\mathcal{R}b$ holds (meaning that a defends itself against b), and the proof is over.

Of course, one could imagine a variant of definition 2.2.2 as follows: $a \in \mathfrak{A}$ is acceptable with respect to S iff for each $b \in \mathfrak{A}$, if $b\mathcal{R}a$ then $c\mathcal{R}b$ for some $c \in S$ where $c \neq a$. But then an argument could never defend itself. Thus, no singleton set of arguments would be admissible.

All this shows that the abstract argumentation approach demands that the attack relation be asymmetric, which is too restrictive a condition as is illustrated by (C), which yields a quite respectable notion of argumentation.

Another concern is that care needs to be taken when formalizing *instantiations* of abstract argumentation. In [CA05], a seemingly reasonable notion of attack is defined that is based on strict and defeasible rules relating formulae of classical logic. However, directly applying acceptability to characterize admissible sets, and the various kinds of extensions, leads to defective, possibly incoherent, conclusions. To address this in [CA05], the authors have proposed some coherence conditions that need to be used with a defeasible logic for the application of abstract argumentation to be safe.

Finally, it can also be argued that challenge between arguments need not always be one-on-one. An illustration is with the lottery paradox. Consider a winner-takes-it-all lottery with 1,000 tickets. There is a strong argument that ticket 1 is not the winning ticket. As well, there is a strong argument that ticket 2 is not the winning ticket. And so on, up to ticket 1,000. Evidently, the argument *ticket n is not the winning ticket* does not attack any other argument *ticket m is not the winning ticket*. That is, all arguments coexist in the same extension. However, it seems necessary to be able to take the first 999 arguments together (i.e., ticket 1 *is not a winning ticket*, ticket 2 *is not a winning ticket*, ..., ticket 999 *is not a winning ticket*) and that together they should then attack the remaining argument (i.e., ticket 1000 *is not a winning ticket*). Thus, intuitively, the 1,000 arguments do *not* form a conflict-free set. Now because it is not possible to take arguments together for an attack on another argument, the formalization by an argument framework is wrong for this kind of example. However, there is a proposal for a generalization of argument frameworks that supports a notion of joint attacks [HP06], and this may offer an interesting way of addressing the lottery paradox.

2.6 Bibliographic Notes

The seminal work on abstract argumentation is by Dung [Dun95]. This framework has been used for a number of logic-based argumentation systems (e.g., [AC02, DKT06]). Dung's proposal has also led to alternative proposals (e.g., [JV00, CDM05]) and extensions such as value-based argument frameworks [Ben03], argument frameworks with constraints [CDM06], bipolar argument frameworks with gradual handling of conflict [ACL04, CL05b], development of credulous and skeptical perspectives [DM04], and merging of argument frameworks [CDK+05]. There are also some proposals for algorithms for argumentation [CDM01, BG02, CDM03], and there is a comprehensive analysis of the computational complexity of some of the key decision problems for abstract argumentation [DB02].

3 Logical Argumentation

To move beyond abstract argumentation, we introduce in this chapter a framework for argumentation in which more details about each argument are considered. In so doing, we distinguish the reasons (i.e., premises), the claim, and the method of inference by which the claim is meant to follow from the reasons. The nature of inference is diverse and includes analogical inference, causal inference, and inductive inference.

We focus on deductive inference and hence on deductive arguments, that is, the claim is a deductively valid consequence of the reasons (the support). We investigate the formalization of such arguments in the setting of classical logic. Thus, our starting position is that a deductive argument consists of a claim entailed by a collection of statements such that the claim as well as the statements are denoted by formulae of classical logic and entailment is identified with deduction in classical logic.

In our framework, an argument is simply a pair where the first item in the pair is a minimal consistent set of formulae that proves the second item. That is, we account for the support and the claim of an argument, though we do not indicate the method of inference, since it does not differ from one argument to another: We only consider deductive arguments; hence, the method of inference for each and every argument is always entailment according to classical logic.

Most proposals for modeling argumentation in logic are very limited in the way that they combine arguments for and against a particular claim. A simple form of argumentation is that a claim follows if and only if there is an argument for the claim and no argument against the claim. In our approach, we check how each argument is challenged by other arguments and by recursion for these counterarguments. Technically, an argument is undercut if and only if some of the reasons for the argument are rebutted (the reader may note that "undercut" is given a different meaning by some authors). Each undercut to a counterargument is itself an argument and so may be undercut, and so by recursion each undercut needs to be considered. Exploring systematically the universe of arguments in order to present an exhaustive synthesis of the relevant chains of undercuts for a given argument is the basic principle of our approach.

Following this, our notion of an argument tree is that it is a synthesis of all the arguments that challenge the argument at the root of the tree, and it also contains all counterarguments that challenge these arguments, and so on, recursively.

3.1 Preliminaries

Prior to any definitions, we first assume a fixed Δ (a finite set of formulae) and use this Δ throughout. We further assume that every subset of Δ is given an enumeration $\langle \alpha_1, \ldots, \alpha_n \rangle$ of its elements, which we call its canonical enumeration. This really is not a demanding constraint: In particular, the constraint is satisfied whenever we impose an arbitrary total ordering over Δ. Importantly, the order has no meaning and is not meant to represent any respective importance of formulae in Δ. It is only a convenient way to indicate the order in which we assume the formulae in any subset of Δ are conjoined to make a formula logically equivalent to that subset.

The paradigm for our approach is a large repository of information, represented by Δ, from which arguments can be constructed for and against arbitrary claims. Apart from information being understood as declarative statements, there is no a priori restriction on the contents and the pieces of information in the repository can be arbitrarily complex. Therefore, Δ is not expected to be consistent. It need even not be the case that individual formulae in Δ are consistent.

The formulae in Δ can represent certain or uncertain information, and they can represent objective, subjective, or hypothetical statements as suggested in chapter 1. Thus, Δ can represent facts, beliefs, views,... Furthermore, the items in Δ can be beliefs from different agents who need not even have the same opinions. It can indeed be the case that an argument formed from such a Δ takes advantage of partial views from different agents. In any case, it is quite possible for Δ to have two or more formulae that are logically equivalent (e.g., Δ can be such that it contains both $\alpha \vee \beta$ and $\beta \vee \alpha$). However, wherever they come from, all formulae in Δ are on a par and treated equitably.

Note that we do not assume any metalevel information about formulae. In particular, we do not assume some preference ordering or "certainty ordering" over formulae. This is in contrast to numerous proposals for argumentation that do assume some form of ordering over formulae. Such orderings can be useful to resolve conflicts by, for example, selecting formulae from a more reliable source. However, this, in a sense, pushes the problem of dealing with conflicting information to one of finding and using orderings over formulae, and as such raises further questions such as the following: Where does the knowledge about reliability of the sources come from? How can it be assessed? How can it be validated? Besides, reliability is not universal; it usually comes in specialized instances. This is not to say priorities are not useful. Indeed it is important to use them in some situations when they are available, but we believe that to understand the elements of argumentation, we need to avoid drawing on them—we need to have a comprehensive framework for argumentation that works without recourse to priorities over formulae.

3.2 Arguments

Here we adopt a very common intuitive notion of an argument and consider some of the ramifications of the definition. Essentially, an argument is a set of appropriate formulae that can be used to classically prove some claim, together with that claim (formulae represent statements, including claims).

Definition 3.2.1 An **argument** is a pair $\langle \Phi, \alpha \rangle$ such that

1. $\Phi \nvdash \bot$.
2. $\Phi \vdash \alpha$.
3. Φ is a minimal subset of Δ satisfying 2.

If $A = \langle \Phi, \alpha \rangle$ is an argument, we say that A is an argument for α (which in general is not an element of Δ) and we also say that Φ is a support for α. We call α the **claim** or the **consequent** of the argument, and we write $\mathsf{Claim}(A) = \alpha$. We call Φ the **support** of the argument, and we write $\mathsf{Support}(A) = \Phi$. Ω denotes the set of all arguments, given Δ.

Example 3.2.1 Let $\Delta = \{\alpha, \alpha \to \beta, \gamma \to \neg\beta, \gamma, \delta, \delta \to \beta, \neg\alpha, \neg\gamma\}$. Some arguments are as follows:

$\langle \{\alpha, \alpha \to \beta\}, \beta \rangle$

$\langle \{\gamma \to \neg\beta, \gamma\}, \neg\beta \rangle$

$\langle \{\delta, \delta \to \beta\}, \beta \rangle$

$\langle \{\neg\alpha\}, \neg\alpha \rangle$

$\langle \{\neg\gamma\}, \neg\gamma \rangle$

$\langle\{\alpha \to \beta\}, \neg\alpha \vee \beta\rangle$

$\langle\{\neg\gamma\}, \delta \to \neg\gamma\rangle$

The need for condition 1 of definition 3.2.1 can be illustrated by means of the next example, adapted from [Rea89].

Example 3.2.2 The story is that René Quiniou contends that John Slaney was in St-Malo on a certain day, and that Sophie Robin denies it. The situation is described by three statements, represented by the formulae to the left:

$s \to q$	If Slaney was in St-Malo, Quiniou is right.
$\neg(r \to q)$	It is not the case that if Robin is right, so is Quiniou.
$\neg(s \to r)$	It is not the case that if Slaney was in St-Malo, Robin is right.

Intuitively, nothing there provides grounds for an argument claiming that I am the Pope (the latter statement is denoted p). Still, note that $\{\neg(r \to q), \neg(s \to r)\}$ is a minimal set of formulae satisfying condition 2 with respect to deducing p:

$\langle\{\neg(r \to q), \neg(s \to r)\} \vdash p$

If it were not for condition 1 that is violated because $\{\neg(r \to q), \neg(s \to r)\}$ is inconsistent,

$\langle\{\neg(r \to q), \neg(s \to r)\}, p\rangle$

would be an argument in the sense of definition 3.2.1, to the effect that I am the Pope!

Condition 2 of definition 3.2.1 aims at ensuring that the support is sufficient for the consequent to hold, as is illustrated in the next example.

Example 3.2.3 The following is a sound deductive argument in free text.

Chemnitz can't host the Summer Olympics because it's a small city and it can host the Summer Olympics only if it is a major city.

Below is an attempt at formalizing the example:

$o \to m$	Chemnitz can host the Summer Olympics only if Chemnitz is a major city.
s	Chemnitz is a small city.

Hence

$\neg o$	Chemnitz cannot host the Summer Olympics.

According to classical logic, the purported conclusion fails to follow from the premises

$\{o \to m, s\} \nvdash \neg o$

because of a missing, implicit, statement:

$s \to \neg m$ If Chemnitz is a small city, then Chemnitz is not a major city.

An enthymeme is a form of reasoning in which some premises are implicit, most often because they are obvious. For example, "The baby no longer has her parents; therefore, she is an orphan" (in symbols, $\neg p$ hence o) is an enthymeme: The reasoning is correct despite omitting the trivial premise stating that "if a baby no longer has her parents, then she is an orphan" (in symbols, $\neg p, \neg p \to o \vdash o$).

Example 3.2.3 shows that, by virtue of condition 2 in definition 3.2.1, arguments in the form of enthymemes are formalized with all components made explicit.

Remember that the method of inference from support to consequent is deduction according to classical logic, which explains the first two conditions in definition 3.2.1.

Minimality (i.e., condition 3 in definition 3.2.1) is not an absolute requirement, although some properties below depend on it. Importantly, the condition is not of a mere technical nature.

Example 3.2.4 Here are a few facts about me ...

p I like paprika.

r I am retiring.

$r \to q$ If I am retiring, then I must quit my job.

It is possible to argue "I must quit my job because I am retiring and if doing so, I must quit" to be captured formally by the argument

$\langle \{r, r \to q\}, q \rangle$

In contrast, it is counterintuitive to argue "I must quit my job because I like paprika and I am retiring and if doing so, I must quit" to be captured formally by

$\langle \{p, r, r \to q\}, q \rangle$

which fails to be an argument because condition 3 is not satisfied.

The underlying idea for condition 3 is that an argument makes explicit the connection between reasons for a claim and the claim itself. However, that would not be the case if the reasons were not exactly identified—in other words, if reasons incorporated irrelevant information and so included formulae not used in the proof of the claim.

Arguments are not necessarily independent. In a sense, some encompass others (possibly up to some form of equivalence), which is the topic we now turn to.

Definition 3.2.2 An argument $\langle \Phi, \alpha \rangle$ is **more conservative** than an argument $\langle \Psi, \beta \rangle$ iff $\Phi \subseteq \Psi$ and $\beta \vdash \alpha$.

Example 3.2.5 $\langle \{\alpha\}, \alpha \vee \beta \rangle$ is more conservative than $\langle \{\alpha, \alpha \rightarrow \beta\}, \beta \rangle$. Here, the latter argument can be obtained from the former (using $\alpha \rightarrow \beta$ as an extra hypothesis), but the reader is warned that this is not the case in general (see proposition 3.2.4).

Roughly speaking, a more conservative argument is more general: It is, so to speak, less demanding on the support and less specific about the consequent.

Example 3.2.6 Now we consider some ruminations on wealth:

p	I own a private jet.
$p \rightarrow q$	If I own a private jet, then I can afford a house in Queens Gardens.

Hence "I can afford a house in Queens Gardens," which is captured by the following argument:

$\langle \{p, p \rightarrow q\}, q \rangle$

Let the same two considerations be supplemented with another one:

p	I own a private jet.
$p \rightarrow q$	If I own a private jet, then I can afford a house in Queens Gardens.
$q \rightarrow r$	If I can afford a house in Queens Gardens, then I am rich.

Hence "I am rich, and I can afford a house in Queens Gardens," which is captured by the following argument:

$\langle \{p, p \rightarrow q, q \rightarrow r\}, r \wedge q \rangle$

However, the previous argument $\langle \{p, p \rightarrow q\}, q \rangle$ is more conservative than $\langle \{p, p \rightarrow q, q \rightarrow r\}, r \wedge q \rangle$, which can somehow be retrieved from it:

$$\left.\begin{array}{l} \langle \{p, p \rightarrow q\}, q \rangle \\ \{q, q \rightarrow r\} \models r \wedge q \end{array}\right\} \Rightarrow \langle \{p, p \rightarrow q, q \rightarrow r\}, r \wedge q \rangle$$

The next section is devoted to a number of formal properties and can be skipped if desired.

3.2.1 Technical Developments

The subsections entitled "Technical developments" provide more detailed consideration of the framework and may be skipped on a first reading. All proofs are to be found in appendix D.

Based on refutation, there is a simple characterization of being an argument.

Proposition 3.2.1 $\langle \Phi, \alpha \rangle$ is an argument iff $\Phi \cup \{\neg\alpha\}$ is a minimal inconsistent set such that $\Phi \subseteq \Delta$.

What requirements are needed to make a new argument out of an existing one? The first possibility deals with the case that the new argument only differs by its consequent.

Proposition 3.2.2 Let $\langle \Phi, \alpha \rangle$ be an argument. If $\Phi \vdash \alpha \to \beta$ and $\beta \to \alpha$ is a tautology, then $\langle \Phi, \beta \rangle$ is also an argument.

It is not possible to loosen the conditions in proposition 3.2.2. Taking $\Phi = \{\beta, \alpha \leftrightarrow \beta\}$ gives an argument $\langle \Phi, \alpha \rangle$ such that $\Phi \vdash \alpha \to \beta$ and $\Phi \vdash \beta \to \alpha$, but $\langle \Phi, \beta \rangle$ fails to be an argument.

Proposition 3.2.3 Let Φ and Ψ be such that there exists a bijection f from Ψ to some partition $\{\Phi_1, \ldots, \Phi_n\}$ of Φ where $\mathsf{Cn}(\{\psi\}) = \mathsf{Cn}(f(\psi))$ for all $\psi \in \Psi$. If $\langle \Phi, \alpha \rangle$ is an argument, then $\langle \Psi, \alpha \rangle$ is also an argument.

The converse of proposition 3.2.3 fails: Let $\Delta = \{\alpha, \alpha \to \beta, \alpha \wedge \beta\}$. Let $\Psi = \{\alpha \wedge \beta\}$ and $\Phi = \{\alpha, \alpha \to \beta\}$. Now, $\langle \Psi, \alpha \rangle$ is an argument, but $\langle \Phi, \alpha \rangle$ is not.

Corollary 3.2.1 Let $\Phi = \{\phi_1, \ldots, \phi_n\} \subseteq \Delta$ and $\Psi = \{\psi_1, \ldots, \psi_n\} \subseteq \Delta$ such that $\phi_i \leftrightarrow \psi_i$ is a tautology for $i = 1 \ldots n$. Let α and β be such that $\alpha \leftrightarrow \beta$ is a tautology. Then, $\langle \Phi, \alpha \rangle$ is an argument iff $\langle \Psi, \beta \rangle$ is an argument.

It is not possible to extend corollary 3.2.1 substantially. Clearly, proposition 3.2.3 can neither be extended to the case $\mathsf{Cn}(\{\psi\}) \subseteq \mathsf{Cn}(f(\psi))$ (if $\Phi = \{\alpha\}$ and $\Psi = \{\alpha \vee \beta\}$, then $\langle \Phi, \alpha \rangle$ is an argument, but $\langle \Psi, \alpha \rangle$ is not) nor to the case $\mathsf{Cn}(f(\psi)) \subseteq \mathsf{Cn}(\{\psi\})$ (if $\Phi = \{\beta \wedge \gamma, \beta \wedge \gamma \to \alpha\}$ and $\Psi = \{\alpha \wedge \beta \wedge \gamma, \alpha \wedge \delta\}$, then $\langle \Phi, \alpha \rangle$ is an argument, but $\langle \Psi, \alpha \rangle$ is not).

Example 3.2.5 suggests that an argument $\langle \Psi, \beta \rangle$ can be obtained from a more conservative argument $\langle \Phi, \alpha \rangle$ by using $\Psi \backslash \Phi$ together with α in order to deduce β (in symbols, $\{\alpha\} \cup \Psi \backslash \Phi \vdash \beta$ or, equivalently, $\Psi \backslash \Phi \vdash \alpha \rightarrow \beta$). As already mentioned, this does not hold in full generality. A counterexample consists of $\langle\{\alpha \wedge \gamma\}, \alpha\rangle$ and $\langle\{\alpha \wedge \gamma, \neg\alpha \vee \beta \vee \neg\gamma\}, \beta\rangle$. However, a weaker property holds.

Proposition 3.2.4 If $\langle\Phi, \alpha\rangle$ is more conservative than $\langle\Psi, \beta\rangle$, then $\Psi \backslash \Phi \vdash \varphi \rightarrow (\alpha \rightarrow \beta)$ for some formula φ such that $\Phi \vdash \varphi$ and $\varphi \nvdash \alpha$ unless α is a tautology.

The interesting case, as in example 3.2.5, is when φ can be a tautology.

What is the kind of structure formed by the set of all arguments? That some arguments are more conservative than others provides the basis for an interesting answer.

Proposition 3.2.5 Being more conservative defines a pre-ordering over arguments. Minimal arguments always exist unless all formulae in Δ are inconsistent. Maximal arguments always exist: They are $\langle\emptyset, \top\rangle$, where $\top$ is any tautology.

A useful notion is then that of a normal form (a function such that any formula is mapped to a logically equivalent formula and, if understood in a strict sense as here, such that any two logically equivalent formulae are mapped to the same formula).

Proposition 3.2.6 Given a normal form, being more conservative defines an ordering provided that only arguments that have a consequent in normal form are considered. The ordered set of all such arguments is an upper semilattice (when restricted to the language of Δ). The greatest argument always exists; it is $\langle\emptyset, \top\rangle$.

Example 3.2.7 The greatest lower bound of $\langle\{\alpha \wedge \beta\}, \alpha\rangle$ and $\langle\{\alpha \wedge \neg\beta\}, \alpha\rangle$ does not exist. If $\Delta = \{\alpha \wedge \beta, \alpha \wedge \neg\beta\}$, then there is no least argument. Taking now $\Delta = \{\alpha, \beta, \alpha \leftrightarrow \beta\}$, there is no least argument either (although Δ is consistent). Even though $\Delta = \{\alpha, \beta \wedge \neg\beta\}$ is inconsistent, the least argument exists: $\langle\{\alpha\}, \alpha'\rangle$ (where α' stands for the normal form of α). As the last illustration, $\Delta = \{\alpha \vee \beta, \beta\}$ admits the least argument $\langle\{\beta\}, \beta'\rangle$ (where β' stands for the normal form of β).

In any case, $\langle\emptyset, \top\rangle$ is more conservative than any other argument.

The notion of being more conservative induces a notion of equivalence between arguments. However, another basis for equating two arguments with each other comes to mind: pairwise logical equivalence of the components of both arguments.

Definition 3.2.3 Two arguments $\langle\Phi, \alpha\rangle$ and $\langle\Psi, \beta\rangle$ are **equivalent** iff Φ is logically equivalent to Ψ and α is logically equivalent to β.

Proposition 3.2.7 Two arguments are equivalent whenever each is more conservative than the other. In partial converse, if two arguments are equivalent, then either each is more conservative than the other or neither is.

Thus, there exist equivalent arguments $\langle\Phi, \alpha\rangle$ and $\langle\Psi, \beta\rangle$ that fail to be more conservative than each other (as in example 3.2.8 below). However, if $\langle\Phi, \alpha\rangle$ is strictly more conservative than $\langle\Psi, \beta\rangle$ (meaning that $\langle\Phi, \alpha\rangle$ is more conservative than $\langle\Psi, \beta\rangle$, but $\langle\Psi, \beta\rangle$ is not more conservative than $\langle\Phi, \alpha\rangle$), then $\langle\Phi, \alpha\rangle$ and $\langle\Psi, \beta\rangle$ are not equivalent.

Example 3.2.8 Let $\Phi = \{\alpha, \beta\}$ and $\Psi = \{\alpha \vee \beta, \alpha \leftrightarrow \beta\}$. The arguments $\langle\Phi, \alpha \wedge \beta\rangle$ and $\langle\Psi, \alpha \wedge \beta\rangle$ are equivalent even though neither is more conservative than the other. This means that there exist two distinct subsets of Δ (namely, Φ and Ψ) supporting $\alpha \wedge \beta$.

While equivalent arguments make the same point (i.e., the same inference), we do want to distinguish equivalent arguments from each other. What we do not want is to distinguish between arguments that are more conservative than each other.

3.3 Defeaters, Rebuttals, and Undercuts

An intuitive idea of counterargument is captured with the notion of defeaters, which are arguments whose claim refutes the support of another argument [FKEG93, Nut94, Vre97, Ver99]. This gives us a general way for an argument to challenge another.

Definition 3.3.1 A **defeater** for an argument $\langle\Phi, \alpha\rangle$ is an argument $\langle\Psi, \beta\rangle$ such that $\beta \vdash \neg(\phi_1 \wedge \cdots \wedge \phi_n)$ for some $\{\phi_1, \ldots, \phi_n\} \subseteq \Phi$.

Example 3.3.1 Let $\Delta = \{\neg\alpha, \alpha \vee \beta, \alpha \leftrightarrow \beta, \gamma \rightarrow \alpha\}$. Then, $\langle\{\alpha \vee \beta, \alpha \leftrightarrow \beta\}, \alpha \wedge \beta\rangle$ is a defeater for $\langle\{\neg\alpha, \gamma \rightarrow \alpha\}, \neg\gamma\rangle$. A more conservative defeater for $\langle\{\neg\alpha, \gamma \rightarrow \alpha\}, \neg\gamma\rangle$ is $\langle\{\alpha \vee \beta, \alpha \leftrightarrow \beta\}, \alpha \vee \gamma\rangle$.

The notion of assumption attack to be found in the literature is less general than the above notion of defeater, of which special cases are undercut and rebuttal as discussed next.

Some arguments directly oppose the support of others, which amounts to the notion of an undercut.

Definition 3.3.2 An **undercut** for an argument $\langle \Phi, \alpha \rangle$ is an argument $\langle \Psi, \neg(\phi_1 \wedge \cdots \wedge \phi_n) \rangle$ where $\{\phi_1, \ldots, \phi_n\} \subseteq \Phi$.

Example 3.3.2 Let $\Delta = \{\alpha, \alpha \to \beta, \gamma, \gamma \to \neg\alpha\}$. Then, $\langle \{\gamma, \gamma \to \neg\alpha\}, \neg(\alpha \wedge (\alpha \to \beta)) \rangle$ is an undercut for $\langle \{\alpha, \alpha \to \beta\}, \beta \rangle$. A less conservative undercut for $\langle \{\alpha, \alpha \to \beta\}, \beta \rangle$ is $\langle \{\gamma, \gamma \to \neg\alpha\}, \neg\alpha \rangle$.

Presumably, the most direct form of a conflict between arguments is when two arguments have opposite claims. This case is captured in the literature through the notion of a rebuttal.

Definition 3.3.3 An argument $\langle \Psi, \beta \rangle$ is a **rebuttal** for an argument $\langle \Phi, \alpha \rangle$ iff $\beta \leftrightarrow \neg\alpha$ is a tautology.

Example 3.3.3 We now return to the Simon Jones affair in five statements:

p	Simon Jones is a Member of Parliament.
$p \to \neg q$	If Simon Jones is a Member of Parliament, then we need not keep quiet about details of his private life.
r	Simon Jones just resigned from the House of Commons.
$r \to \neg p$	If Simon Jones just resigned from the House of Commons, then he is not a Member of Parliament.
$\neg p \to q$	If Simon Jones is not a Member of Parliament, then we need to keep quiet about details of his private life.

The first two statements form an argument A whose claim is that we can publicize details about his private life. The next two statements form an argument whose claim is that he is not a Member of Parliament (contradicting an item in the support of A) and that is a counterargument against A. The last three statements combine to give an argument whose claim is that we cannot publicize details about his private life (contradicting the claim of A), and that, too, is a counterargument against A. In symbols, we obtain the following argument (below left) and counterarguments (below right).

$$\langle \{p, p \to \neg q\}, \neg q \rangle \quad \begin{cases} \text{An undercut is } \langle \{r, r \to \neg p\}, \neg p \rangle \\ \text{A rebuttal is } \langle \{r, r \to \neg p, \neg p \to q\}, q \rangle \end{cases}$$

Trivially, undercuts are defeaters, but it is also quite simple to establish the next result.

Proposition 3.3.1 If $\langle \Psi, \beta \rangle$ is a rebuttal for an argument $\langle \Phi, \alpha \rangle$, then $\langle \Psi, \beta \rangle$ is a defeater for $\langle \Phi, \alpha \rangle$.

If an argument has defeaters, then it has undercuts, naturally. It may happen that an argument has defeaters but no rebuttals as illustrated next.

Example 3.3.4 Let $\Delta = \{\alpha \wedge \beta, \neg\beta\}$. Then, $\langle\{\alpha \wedge \beta\}, \alpha\rangle$ has at least one defeater but no rebuttal.

Here are some details on the differences between rebuttals and undercuts.

An undercut for an argument need not be a rebuttal for that argument As a first illustration, $\langle\{\neg\alpha\}, \neg\alpha\rangle$ is an undercut for $\langle\{\alpha, \alpha \to \beta\}, \beta\rangle$ but is not a rebuttal for it. Clearly, $\langle\{\neg\alpha\}, \neg\alpha\rangle$ does not rule out β. Actually, an undercut may even agree with the consequent of the objected argument: $\langle\{\beta \wedge \neg\alpha\}, \neg\alpha\rangle$ is an undercut for $\langle\{\alpha, \alpha \to \beta\}, \beta\rangle$. In this case, we have an argument with an undercut that conflicts with the support of the argument but implicitly provides an alternative way to deduce the consequence of the argument (see the so-called overzealous arguments in section 5.3.1). This should make it clear that an undercut need not question the consequent of an argument but only the reason(s) given by that argument to support its consequent. Of course, there are also undercuts that challenge an argument on both counts: Just consider $\langle\{\neg\alpha \wedge \neg\beta\}, \neg\alpha\rangle$, which is such an undercut for $\langle\{\alpha, \alpha \to \beta\}, \beta\rangle$.

A rebuttal for an argument need not be an undercut for that argument As an example, $\langle\{\neg\beta\}, \neg\beta\rangle$ is a rebuttal for $\langle\{\alpha, \alpha \to \beta\}, \beta\rangle$ but is not an undercut for it because β is not in $\{\alpha, \alpha \to \beta\}$. Observe that there is not even an argument equivalent to $\langle\{\neg\beta\}, \neg\beta\rangle$ that would be an undercut for $\langle\{\alpha, \alpha \to \beta\}, \beta\rangle$: In order to be an undercut for $\langle\{\alpha, \alpha \to \beta\}, \beta\rangle$, an argument should be of the form $\langle\Phi, \neg\alpha\rangle$, $\langle\Phi, \neg(\alpha \to \beta)\rangle$ or $\langle\Phi, \neg(\alpha \wedge (\alpha \to \beta))\rangle$, but $\neg\beta$ is not logically equivalent to $\neg\alpha$, $\neg(\alpha \to \beta)$ or $\neg(\alpha \wedge (\alpha \to \beta))$.

Anyway, a rebuttal for an argument is a less conservative version of a specific undercut for that argument as we now prove.

Proposition 3.3.2 If $\langle\Psi, \beta\rangle$ is a defeater for $\langle\Phi, \alpha\rangle$, then there exists an undercut for $\langle\Phi, \alpha\rangle$ that is more conservative than $\langle\Psi, \beta\rangle$.

Corollary 3.3.1 If $\langle\Psi, \beta\rangle$ is a rebuttal for $\langle\Phi, \alpha\rangle$, then there exists an undercut for $\langle\Phi, \alpha\rangle$ that is more conservative than $\langle\Psi, \beta\rangle$.

The undercut mentioned in proposition 3.3.2 and corollary 3.3.1 is strictly more conservative than $\langle\Psi, \beta\rangle$ whenever $\neg\beta$ fails to be logically equivalent with Φ.

$$\begin{array}{c} A \\ \uparrow \\ A_2 \\ \uparrow \\ A_3 \\ \uparrow \\ \vdots \end{array}$$

Figure 3.1
If A has a defeater, there is an infinite series of counterarguments, starting with a counterargument for A, and then for each counterargument, by recursion, there is a counterargument.

Proposition 3.3.3 If an argument has a defeater, then there exists an undercut for its defeater.

Using $\mathbb{N}$ to denote the nonnegative integers, a noticeable consequence can be stated as follows.

Corollary 3.3.2 If an argument A has at least one defeater, then there exists an infinite sequence of arguments $(A_n)_{n \in \mathbb{N}^*}$ such that A_1 is A and A_{n+1} is an undercut of A_n for every $n \in \mathbb{N}^*$.

As an illustration of corollary 3.3.2, see figure 3.1.

Example 3.3.5 Let $\Delta = \{\alpha, \alpha \to \beta, \gamma \wedge \neg\alpha\}$. Then,

$$\langle\{\alpha, \alpha \to \beta\}, \alpha \wedge \beta\rangle$$

is an argument for which a defeater is

$$\langle\{\gamma \wedge \neg\alpha\}, \gamma \wedge \neg\alpha\rangle$$

The argument $\langle\{\alpha, \alpha \to \beta\}, \alpha \wedge \beta\rangle$ gives rise to an infinite series of arguments where each one is an undercut for the previous argument in the series:

$$\begin{array}{c} \langle\{\alpha, \alpha \to \beta\}, \alpha \wedge \beta\rangle \\ \uparrow \\ \langle\{\gamma \wedge \neg\alpha\}, \neg\alpha\rangle \\ \uparrow \\ \langle\{\alpha\}, \neg(\gamma \wedge \neg\alpha)\rangle \\ \uparrow \\ \langle\{\gamma \wedge \neg\alpha\}, \neg\alpha\rangle \\ \vdots \end{array}$$

Corollary 3.3.2 is obviously a potential concern for representing and comparing arguments. We address this question in section 3.5.

Section 2.1 introduced the notions of self-attacking arguments and controversial arguments. The next two results deal with such special arguments.

Proposition 3.3.4 Let $\langle\Phi,\alpha\rangle$ be an argument for which $\langle\Psi,\beta\rangle$ is a defeater. Then, $\Psi \not\subseteq \Phi$.

Proposition 3.3.4 proves that, in the sense of definition 3.2.1 and definition 3.3.1, no argument is self-defeating.

Proposition 3.3.5 If $\langle\Gamma,\neg\psi\rangle$ is an undercut for $\langle\Psi,\neg\phi\rangle$, which is itself an undercut for an argument $\langle\Phi,\alpha\rangle$, then $\langle\Gamma,\neg\psi\rangle$ is not a defeater for $\langle\Phi,\alpha\rangle$.

Proposition 3.3.5 shows that no undercut is controversial, again in the sense of definition 3.2.1 and definition 3.3.1. This does not extend to defeaters as illustrated by the following example.

Example 3.3.6 Consider the following arguments:

A_1 $\langle\{\alpha\wedge\neg\beta\},\alpha\rangle$

A_2 $\langle\{\neg\alpha\wedge\beta\},\neg(\alpha\wedge\neg\beta)\rangle$

A_3 $\langle\{\neg\alpha\wedge\neg\beta\},\neg\alpha\wedge\neg\beta\rangle$

Thus, A_2 is an undercut for A_1 and A_3 is a defeater for A_2. Furthermore, A_3 is also a defeater for A_1.

As arguments can be ordered from more conservative to less conservative, there is a clear and unambiguous notion of maximally conservative defeaters for a given argument (the ones that are representative of all defeaters for that argument).

Definition 3.3.4 $\langle\Psi,\beta\rangle$ is a **maximally conservative defeater** of $\langle\Phi,\alpha\rangle$ iff for all defeaters $\langle\Psi',\beta'\rangle$ of $\langle\Phi,\alpha\rangle$, if $\Psi' \subseteq \Psi$ and $\beta \vdash \beta'$, then $\Psi \subseteq \Psi'$ and $\beta' \vdash \beta$.

Equivalently, $\langle\Psi,\beta\rangle$ is a maximally conservative defeater of $\langle\Phi,\alpha\rangle$ iff $\langle\Psi,\beta\rangle$ is a defeater of $\langle\Phi,\alpha\rangle$ such that no defeaters of $\langle\Phi,\alpha\rangle$ are strictly more conservative than $\langle\Psi,\beta\rangle$.

Proposition 3.3.6 If $\langle\Psi,\beta\rangle$ is a maximally conservative defeater of $\langle\Phi,\alpha\rangle$, then $\langle\Psi,\beta'\rangle$ is an undercut of $\langle\Phi,\alpha\rangle$ for some β' that is logically equivalent with β.

Proposition 3.3.6 suggests that we focus on undercuts when seeking counterarguments to a given argument as is investigated from now on.

3.3.1 Technical Developments

The first question to be investigated here is under what condition can an argument defeat its defeaters?

Proposition 3.3.7 Given two arguments $\langle\Phi, \alpha\rangle$ and $\langle\Psi, \beta\rangle$ such that $\{\alpha, \beta\} \vdash \phi$ for each $\phi \in \Phi$, if $\langle\Psi, \beta\rangle$ is a defeater for $\langle\Phi, \alpha\rangle$, then $\langle\Phi, \alpha\rangle$ is a defeater for $\langle\Psi, \beta\rangle$.

Corollary 3.3.3 Let α be logically equivalent with Φ. If $\langle\Psi, \beta\rangle$ is a defeater for $\langle\Phi, \alpha\rangle$, then $\langle\Phi, \alpha\rangle$ is a defeater for $\langle\Psi, \beta\rangle$.

A follow-up is the case about rebuttals.

Corollary 3.3.4 If $\langle\Psi, \beta\rangle$ is a rebuttal for $\langle\Phi, \alpha\rangle$, then $\langle\Phi, \alpha\rangle$ is a rebuttal for $\langle\Psi, \beta\rangle$.

A similar question is when are two arguments a defeater of each other?

Proposition 3.3.8 Given two arguments $\langle\Phi, \alpha\rangle$ and $\langle\Psi, \beta\rangle$ such that $\neg(\alpha \wedge \beta)$ is a tautology, $\langle\Psi, \beta\rangle$ is a defeater for $\langle\Phi, \alpha\rangle$, and $\langle\Phi, \alpha\rangle$ is a defeater for $\langle\Psi, \beta\rangle$.

While proposition 3.3.4 expresses that the defeat relation is anti-reflexive, proposition 3.3.7 and proposition 3.3.8 show that the defeat relation is symmetric on parts of the domain (i.e., it is symmetric for some arguments).

Returning to features of individual counterarguments, what does it take for a defeater to be a rebuttal?

Proposition 3.3.9 Let $\langle\Psi, \beta\rangle$ be a defeater for an argument $\langle\Phi, \alpha\rangle$. If $\alpha \vee \beta$ is a tautology and $\{\alpha, \beta\} \vdash \phi$ for each $\phi \in \Phi$, then $\langle\Psi, \beta\rangle$ is a rebuttal for $\langle\Phi, \alpha\rangle$.

Proposition 3.3.10 If $\langle\Phi, \alpha\rangle$ is an argument where Φ is logically equivalent with α, then each defeater $\langle\Psi, \beta\rangle$ of $\langle\Phi, \alpha\rangle$ such that $\alpha \vee \beta$ is a tautology is a rebuttal for $\langle\Phi, \alpha\rangle$.

Note: In proposition 3.3.10, the assumption that $\alpha \vee \beta$ is a tautology can be omitted when considering the alternative definition of a rebuttal where $\langle\Psi, \beta\rangle$ is a rebuttal for $\langle\Phi, \alpha\rangle$ iff $\neg\alpha \vee \neg\beta$ is a tautology (and the proof gets simpler, of course).

It has been exemplified above that the notion of a rebuttal and the notion of an undercut are independent. The next result characterizes the cases in which both notions coincide.

Proposition 3.3.11 Let $\langle \Phi, \alpha \rangle$ and $\langle \Psi, \beta \rangle$ be two arguments. $\langle \Psi, \beta \rangle$ is both a rebuttal and an undercut for $\langle \Phi, \alpha \rangle$ iff Φ is logically equivalent with α and β is $\neg(\phi_1 \wedge \cdots \wedge \phi_n)$ such that $\Phi = \{\phi_1, \ldots, \phi_n\}$.

Interestingly enough, the support of a maximally conservative defeater leaves no choice about its consequent as is proven in proposition 3.3.12.

Proposition 3.3.12 Let $\langle \Psi, \beta \rangle$ be a maximally conservative defeater for an argument $\langle \Phi, \alpha \rangle$. Then, $\langle \Psi, \gamma \rangle$ is a maximally conservative defeater for $\langle \Phi, \alpha \rangle$ iff γ is logically equivalent with β.

Proposition 3.3.12 does not extend to undercuts because they are syntax dependent (in an undercut, the consequent is always a formula governed by negation).

Lastly, in what way does the existence of a defeater relate to inconsistency for Δ?

Proposition 3.3.13 Δ is inconsistent if there exists an argument that has at least one defeater. Should there be some inconsistent formula in Δ, the converse is untrue. When no formula in Δ is inconsistent, the converse is true in the form: If Δ is inconsistent, then there exists an argument that has at least one rebuttal.

Corollary 3.3.5 Δ is inconsistent if there exists an argument that has at least one undercut. The converse is true when each formula in Δ is consistent.

3.4 Canonical Undercuts

As defined above, an undercut for an argument $\langle \Phi, \alpha \rangle$ is an argument $\langle \Psi, \neg(\phi_1 \wedge \cdots \wedge \phi_n) \rangle$ where $\{\phi_1, \ldots, \phi_n\} \subseteq \Phi$ and $\Phi \cup \Psi \subseteq \Delta$ by the definition of an argument.

While proposition 3.3.2 and proposition 3.3.6 point to undercuts as candidates to be representative of all defeaters for an argument, maximally conservative undercuts are even better candidates.

Definition 3.4.1 $\langle \Psi, \beta \rangle$ is a **maximally conservative undercut** of $\langle \Phi, \alpha \rangle$ iff for all undercuts $\langle \Psi', \beta' \rangle$ of $\langle \Phi, \alpha \rangle$, if $\Psi' \subseteq \Psi$ and $\beta \vdash \beta'$ then $\Psi \subseteq \Psi'$ and $\beta' \vdash \beta$.

Evidently, $\langle \Psi, \beta \rangle$ is a maximally conservative undercut of $\langle \Phi, \alpha \rangle$ iff $\langle \Psi, \beta \rangle$ is an undercut of $\langle \Phi, \alpha \rangle$ such that no undercuts of $\langle \Phi, \alpha \rangle$ are strictly more conservative than $\langle \Psi, \beta \rangle$.

Stressing the relevance of maximally conservative undercuts, it can be proved (proposition 3.4.6) that each maximally conservative undercut is a maximally conservative defeater (but not vice versa, of course).

Example 3.4.1 now shows that a collection of counterarguments to the same argument can sometimes be summarized in the form of a single maximally conservative undercut of the argument, thereby avoiding some amount of redundancy among counterarguments.

Example 3.4.1 Consider the following formulae concerning who is going to a party:

$r \to \neg p \wedge \neg q$	If Rachel goes, neither Paul nor Quincy goes.
p	Paul goes.
q	Quincy goes.

Hence both Paul and Quincy go (initial argument):

$\langle \{p, q\}, p \wedge q \rangle$

Now assume the following additional piece of information:

r Rachel goes.

Hence Paul does not go (a first counterargument):

$\langle \{r, r \to \neg p \wedge \neg q\}, \neg p \rangle$

Hence Quincy does not go (a second counterargument):

$\langle r, r \to \neg p \wedge \neg q\}, \neg q \rangle$

A maximally conservative undercut (for the initial argument) that subsumes both counterarguments above is

$\langle \{r, r \to \neg p \wedge \neg q\}, \neg(p \wedge q) \rangle$

The fact that the maximally conservative undercut in example 3.4.1 happens to be a rebuttal of the argument is only accidental. Actually, the consequent of a maximally conservative undercut for an argument is exactly the negation of the full support of the argument.

Proposition 3.4.1 If $\langle \Psi, \neg(\phi_1 \wedge \cdots \wedge \phi_n) \rangle$ is a maximally conservative undercut for an argument $\langle \Phi, \alpha \rangle$, then $\Phi = \{\phi_1, \ldots, \phi_n\}$.

Note that if $\langle \Psi, \neg(\phi_1 \wedge \cdots \wedge \phi_n)\rangle$ is a maximally conservative undercut for an argument $\langle \Phi, \alpha\rangle$, then so are $\langle \Psi, \neg(\phi_2 \wedge \cdots \wedge \phi_n \wedge \phi_1)\rangle$ and $\langle \Psi, \neg(\phi_3 \wedge \cdots \wedge \phi_n \wedge \phi_1 \wedge \phi_2)\rangle$ and so on. However, they are all identical (in the sense that each is more conservative than the others). We can ignore the unnecessary variants by just considering the canonical undercuts defined as follows.

Definition 3.4.2 An argument $\langle \Psi, \neg(\phi_1 \wedge \cdots \wedge \phi_n)\rangle$ is a **canonical undercut** for $\langle \Phi, \alpha\rangle$ iff it is a maximally conservative undercut for $\langle \Phi, \alpha\rangle$ and $\langle \phi_1, \ldots, \phi_n\rangle$ is the canonical enumeration of Φ.

Next is a simple and convenient characterization of the notion of a canonical undercut.

Proposition 3.4.2 An argument $\langle \Psi, \neg(\phi_1 \wedge \cdots \wedge \phi_n)\rangle$ is a canonical undercut for $\langle \Phi, \alpha\rangle$ iff it is an undercut for $\langle \Phi, \alpha\rangle$ and $\langle \phi_1, \ldots, \phi_n\rangle$ is the canonical enumeration of Φ.

Corollary 3.4.1 A pair $\langle \Psi, \neg(\phi_1 \wedge \cdots \wedge \phi_n)\rangle$ is a canonical undercut for $\langle \Phi, \alpha\rangle$ iff it is an argument and $\langle \phi_1, \ldots, \phi_n\rangle$ is the canonical enumeration of Φ.

Clearly, an argument may have more than one canonical undercut. What do the canonical undercuts for the same argument look like? How do they differ from one another?

Proposition 3.4.3 Any two different canonical undercuts for the same argument have the same consequent but distinct supports.

Proposition 3.4.4 Given two different canonical undercuts for the same argument, neither is more conservative than the other.

Example 3.4.2 Let $\Delta = \{\alpha, \beta, \neg\alpha, \neg\beta\}$. Both of the following

$\langle \{\neg\alpha\}, \neg(\alpha \wedge \beta)\rangle$

$\langle \{\neg\beta\}, \neg(\alpha \wedge \beta)\rangle$

are canonical undercuts for $\langle \{\alpha, \beta\}, \alpha \leftrightarrow \beta\rangle$, but neither is more conservative than the other.

Proposition 3.4.5 For each defeater $\langle \Psi, \beta\rangle$ of an argument $\langle \Phi, \alpha\rangle$, there exists a canonical undercut for $\langle \Phi, \alpha\rangle$ that is more conservative than $\langle \Psi, \beta\rangle$.

That is, the set of all canonical undercuts of an argument represents all the defeaters of that argument (informally, all its counterarguments). This is to be taken advantage of in section 3.5.

3.4.1 Technical Developments

Restricting ourselves to maximally conservative undercuts forces us to check that they capture maximally conservative defeaters. In fact, they do, in a strong sense as given by proposition 3.4.6.

Proposition 3.4.6 If $\langle \Psi, \beta \rangle$ is a maximally conservative undercut for $\langle \Phi, \alpha \rangle$, then $\langle \Psi, \beta \rangle$ also is a maximally conservative defeater for $\langle \Phi, \alpha \rangle$.

As to a different matter, the converse of proposition 3.4.1 also holds.

Proposition 3.4.7 If $\langle \Psi, \neg(\phi_1 \wedge \cdots \wedge \phi_n) \rangle$ is an undercut for $\langle \Phi, \alpha \rangle$ such that $\Phi = \{\phi_1, \ldots, \phi_n\}$, then it is a maximally conservative undercut for $\langle \Phi, \alpha \rangle$.

Corollary 3.4.2 Let $\langle \Psi, \neg(\phi_1 \wedge \cdots \wedge \phi_n) \rangle$ be an undercut for $\langle \Phi, \alpha \rangle$. $\langle \Psi, \neg(\phi_1 \wedge \cdots \wedge \phi_n) \rangle$ is a maximally conservative undercut for $\langle \Phi, \alpha \rangle$ iff $\Phi = \{\phi_1, \ldots, \phi_n\}$.

Although not strictly necessary, a sufficient condition for being a maximally conservative undercut is as follows.

Proposition 3.4.8 If $\langle \Psi, \beta \rangle$ is both a rebuttal and an undercut for $\langle \Phi, \alpha \rangle$ then $\langle \Psi, \beta \rangle$ is a maximally conservative undercut for $\langle \Phi, \alpha \rangle$.

In partial converse, there is a special case of the following general result.

Proposition 3.4.9 Let $\langle \Phi, \alpha \rangle$ be an argument. Each maximally conservative undercut for $\langle \Phi, \alpha \rangle$ is a rebuttal for $\langle \Phi, \alpha \rangle$ iff Φ is logically equivalent with α.

In general, when an argument is a maximally conservative undercut for another argument, the converse does not hold but it is almost the case as shown next.

Proposition 3.4.10 If $\langle \Psi, \beta \rangle$ is a maximally conservative undercut for $\langle \Phi, \alpha \rangle$, then there exists $\Phi' \subseteq \Phi$ and γ such that $\langle \Phi', \gamma \rangle$ is a maximally conservative undercut for $\langle \Psi, \beta \rangle$.

Corollary 3.4.3 If $\langle \Psi, \beta \rangle$ is a canonical undercut for $\langle \Phi, \alpha \rangle$, then there exists $\Phi' \subseteq \Phi$ such that $\langle \Phi', \neg(\psi_1 \wedge \cdots \wedge \psi_n) \rangle$ is a canonical undercut for $\langle \Psi, \beta \rangle$ (where $\langle \psi_1, \ldots, \psi_n \rangle$ is the canonical enumeration of Ψ).

Under certain circumstances, the notion of a rebuttal and the notion of maximally conservative coincide. One of these circumstances is given by the following result.

Proposition 3.4.11 If $\langle \Phi, \alpha \rangle$ is an argument such that Φ is logically equivalent with α, then each rebuttal of $\langle \Phi, \alpha \rangle$ is equivalent with a canonical undercut for $\langle \Phi, \alpha \rangle$.

We will consider, in the next section, how canonical undercuts constitute a key concept for forming constellations of arguments and counterarguments.

3.5 Argument Trees

How does argumentation usually take place? Argumentation starts when an initial argument is put forward, making some claim. An objection is raised in the form of a counterargument. The latter is addressed in turn, eventually giving rise to a counter-counterargument, if any. And so on. However, there often is more than one counterargument to the initial argument, and if the counterargument actually raised in the first place had been different, the counter-counterargument would have been different, too, and similarly the counter-counter-counterargument, if any, and so on. Argumentation would have taken a possibly quite different course.

So do we find all the alternative courses that could take place from a given initial argument? And is it possible to represent them in a rational way, let alone to answer the most basic question of how do we make sure that no further counterargument can be expressed from the information available?

Answers are provided below through the notion of argument trees, but we first present an example to make things a little less abstract. Just a word of warning: The details of reasoning from support to consequent for each (counter-)argument do not really matter; they are made explicit in the example for the sake of completeness only.

Example 3.5.1 There are rumors about Ms. Shy expecting Mr. Scoundrel to propose to her, although she may not wish to get married. (To respect privacy, the names have been changed.) Here is the situation:

- If he doesn't propose to her unless he finds she is rich, then he has no qualms.
- If he proposes to her only if she looks sexy, then it is not the case that he has no qualms.
- He finds out that she is rich.
- She looks sexy.
- He doesn't propose to her!

There are grounds for various arguments about whether Mr. Scoundrel has no qualms about marrying Ms. Shy:

An argument claiming that Mr. Scoundrel has no qualms As stated, if he doesn't propose to her unless he finds she is rich, then he has no qualms. In other words, if it is false that he has no qualms, it is false that he doesn't propose to her unless he finds she is rich. Equivalently, if it is false that he has no qualms, then he proposes to her while he does not find she is rich—which is not the case: He finds she is rich, and it is possible to conclude, by modus tollens, that he has no qualms.

A first counterargument The sentence "at least in the event that he proposes to her, she looks sexy" is true: See the fourth statement. However, the sentence means the same as "he proposes to her only if she looks sexy." According to "if he proposes to her only if she looks sexy, then it is not the case that he has no qualms," it follows that it is not the case that he has no qualms.

A second counterargument The claim in the initial argument is based on the condition "he proposes to her only if he finds she is rich" that can be challenged: Clearly, not both conclusions of "if he proposes to her only if he finds she is rich, then he has no qualms" and "if he proposes to her only if she looks sexy, then it is not the case that he has no qualms" are true. Hence, "he proposes to her only if he finds she is rich" is false if "he proposes to her only if she looks sexy" is true—which is the case as detailed in the first counterargument on the ground that she looks sexy.

A counter-counterargument It is asserted that if he doesn't propose to her unless he finds she is rich, then he has no qualms. Stated otherwise, if it is false that he has no qualms, then it is false that he doesn't propose to her unless he finds she is rich. Equivalently, if it is false that he has no qualms, then he proposes to her while he does not find she is rich—which is not the case: He doesn't propose to her, and it is possible to conclude, by modus tollens, that he has no qualms.

Using the following propositional atoms

p He proposes to her

q He has qualms

r He finds out that she is rich

s She looks sexy

the statements are formalized as

$(\neg r \rightarrow \neg p) \rightarrow \neg q$	If he doesn't propose to her unless he finds she is rich, then he has no qualms
$(p \rightarrow s) \rightarrow \neg\neg q$	If he proposes to her only if she looks sexy, then it is not the case that he has no qualms
r	He finds out that she is rich
s	She looks sexy
$\neg p$	He does not propose to her

and the arguments presented above take the form

$\langle\{r, (\neg r \rightarrow \neg p) \rightarrow \neg q\}, \neg q\rangle$ (initial argument I)

$\langle\{s, (p \rightarrow s) \rightarrow \neg\neg q\}, \neg\neg q\rangle$ (counterargument C_1)

$\langle\{s, (p \rightarrow s) \rightarrow \neg\neg q, (\neg r \rightarrow \neg p) \rightarrow \neg q\},$
$\neg(p \rightarrow r)\rangle$ (counterargument C_2)

$\langle\{\neg p, (\neg r \rightarrow \neg p) \rightarrow \neg q\}, \neg q\rangle$ (counter-counterargument C)

There are still very many other counter^n-arguments (whatever n), despite the fact that definition 3.2.1 already rules out a number of informal ones.

Indeed, the argumentation about Mr. Scoundrel's qualms can take *different courses*. One is $I, C_1, C, \ldots$, another is $I, C_2, C, \ldots$, and there are many others issued from I. It would thus be useful to have an exhaustive account of the possible arguments and how they relate with respect to the initial argument (allowing us to reconstruct every possible course of argumentation starting with a given initial argument). And this is what argument trees are meant to do: An argument tree describes the various ways a given initial argument can be challenged, as well as how the counterarguments to the initial argument can themselves be challenged, and so on, recursively. However, some way of forming sequences of counterarguments is desirable, if not imperative, in view of corollary 3.3.2, which points out that argumentation often takes a course consisting of an infinite sequence of arguments, each being a counterargument to the preceding one.

Example 3.5.2 Argumentation sometimes falls on deaf ears, most often when simplistic arguments are uttered as illustrated below with a case of the "chicken and egg dilemma." Here the same arguments are used in a repetitive cycle:

Dairyman Egg was first.

Farmer Chicken was first.

Dairyman Egg was first.

Farmer Chicken was first.

... ...

The following propositional atoms are introduced:

p Egg was first.

q Chicken was first.

r The chicken comes from the egg.

s The egg comes from the chicken.

Admittedly, that the chicken was first and that the egg was first are not equivalent (i.e., $\neg(p \leftrightarrow q)$). Also, the egg comes from the chicken (i.e., r), and the chicken comes from the egg (i.e., s). Moreover, if the egg comes from the chicken, then the egg was not first. (i.e., $r \rightarrow \neg q$). Similarly, if the chicken comes from the egg, then the chicken was not first (i.e., $s \rightarrow \neg p$). Then, the above dispute can be represented as follows:

$$
\begin{array}{c}
\langle \{s \rightarrow \neg p, s, \neg(p \leftrightarrow q)\}, q \rangle \\
\uparrow \\
\langle \{r \rightarrow \neg q, r, \neg(p \leftrightarrow q)\}, p \rangle \\
\uparrow \\
\langle \{s \rightarrow \neg p, s, \neg(p \leftrightarrow q)\}, q \rangle \\
\uparrow \\
\langle \{r \rightarrow \neg q, r, \neg(p \leftrightarrow q)\}, p \rangle \\
\uparrow \\
\vdots
\end{array}
$$

We are now ready for our definition (below) of an argument tree in which the root of the tree is an argument of interest, and the children for any node are the canonical undercuts for that node. In the definition, we avoid the circularity seen in the above example by incorporating an intuitive constraint.

Definition 3.5.1 An **argument tree** for α is a tree where the nodes are arguments such that

1. The root is an argument for α.
2. For no node $\langle \Phi, \beta \rangle$ with ancestor nodes $\langle \Phi_1, \beta_1 \rangle, \dots, \langle \Phi_n, \beta_n \rangle$ is Φ a subset of $\Phi_1 \cup \cdots \cup \Phi_n$.
3. The children nodes of a node N consist of all canonical undercuts for N that obey 2.

Note, for the definition of argument tree, in chapter 4 onwards, we assume a relaxed version of condition 3 in which "the children nodes of a node N consist of some or all of the canonical undercuts for N that obey 2." In addition, for chapter 4 onwards, we use the term **complete argument tree** when we need to stress that condition 3 is "the children nodes of a node N consist of all the canonical undercuts for N that obey 2."

We illustrate the definition of an argument tree in the following examples.

Example 3.5.3 Speaking of Simon Jones, once again ...

p	Simon Jones is a Member of Parliament.
$p \to \neg q$	If Simon Jones is a Member of Parliament, then we need not keep quiet about details of his private life.
r	Simon Jones just resigned from the House of Commons.
$r \to \neg p$	If Simon Jones just resigned from the House of Commons, then he is not a Member of Parliament.
$\neg p \to q$	If Simon Jones is not a Member of Parliament, then we need to keep quiet about details of his private life.

The situation can be depicted as follows:

$$\begin{array}{c}\langle\{p, p \to \neg q\}, \neg q\rangle \\ \uparrow \\ \langle\{r, r \to \neg p\}, \neg p\rangle\end{array}$$

Example 3.5.4 An obnoxious vice-president ...

Vice-President The only one not taking orders from me is the president; you are not the president but a regular employee; hence you take orders from me.

Secretary I am the president's secretary, not a regular employee.

Secretary Anyway, I don't take orders from you.

This can be captured in terms of the sentences below (as recounted by the secretary).

t	I take orders from the vice-president.
p	I am the president.
r	I am a member of the regular staff.
s	I am the president's secretary.

The statements uttered are as follows:

$\neg t \rightarrow p$	The only one not taking orders from me is the president.
$\neg p \wedge r$	You are not the president but a regular employee.
$s \wedge \neg r$	I am the president's secretary, not a regular employee.
$\neg t$	I don't take orders from you.

We obtain the following argument tree:

$$\begin{array}{ccc} & \langle\{\neg t \rightarrow p, \neg p \wedge r\}, t\rangle & \\ \nearrow & & \nwarrow \\ \langle\{s \wedge \neg r\}, \neg((\neg t \rightarrow p) \wedge (\neg p \wedge r))\rangle & & \langle\{\neg t\}, \neg((\neg t \rightarrow p) \wedge (\neg p \wedge r))\rangle \end{array}$$

We give a further illustration of an argument tree in example 3.5.5, and then we motivate the conditions of definition 3.5.1 as follows: Condition 2 is meant to avoid the situation illustrated by example 3.5.6, and condition 3 is meant to avoid the situation illustrated by example 3.5.7.

Example 3.5.5 Given $\Delta = \{\alpha, \alpha \rightarrow \beta, \gamma, \gamma \rightarrow \neg\alpha, \neg\gamma \vee \neg\alpha\}$, we have the following argument tree:

$$\begin{array}{ccc} & \langle\{\alpha, \alpha \rightarrow \beta\}, \beta\rangle & \\ \nearrow & & \nwarrow \\ \langle\{\gamma, \gamma \rightarrow \neg\alpha\}, \neg(\alpha \wedge (\alpha \rightarrow \beta))\rangle & & \langle\{\gamma, \neg\gamma \vee \neg\alpha\}, \neg(\alpha \wedge (\alpha \rightarrow \beta))\rangle \end{array}$$

Note the two undercuts are equivalent. They do count as two arguments because they are based on two different items of the knowledgebase (even though these items turn out to be logically equivalent).

We adopt a lighter notation, writing $\langle\Psi, \diamond\rangle$ for a canonical undercut of $\langle\Phi, \beta\rangle$. Clearly, $\diamond$ is $\neg(\phi_1 \wedge \cdots \wedge \phi_n)$ where $\langle\phi_1, \ldots, \phi_n\rangle$ is the canonical enumeration for Φ.

Example 3.5.6 Let $\Delta = \{\alpha, \alpha \rightarrow \beta, \gamma \rightarrow \neg\alpha, \gamma\}$.

$$\begin{array}{c} \langle\{\alpha, \alpha \rightarrow \beta\}, \beta\rangle \\ \uparrow \\ \langle\{\gamma, \gamma \rightarrow \neg\alpha\}, \diamond\rangle \\ \uparrow \\ \langle\{\alpha, \gamma \rightarrow \neg\alpha\}, \diamond\rangle \end{array}$$

This is not an argument tree because condition 2 is not met. The undercut to the undercut is actually making exactly the same point (that α and γ are incompatible) as the undercut itself does, just by using modus tollens instead of modus ponens.

Example 3.5.7 Given $\Delta = \{\alpha, \beta, \alpha \to \gamma, \beta \to \delta, \neg\alpha \vee \neg\beta\}$, consider the following tree:

$$\begin{array}{ccc}
 & \langle\{\alpha, \beta, \alpha \to \gamma, \beta \to \delta\}, \gamma \wedge \delta\rangle & \\
\nearrow & & \nwarrow \\
\langle\{\alpha, \neg\alpha \vee \neg\beta\}, \neg\beta\rangle & & \langle\{\beta, \neg\alpha \vee \neg\beta\}, \neg\alpha\rangle
\end{array}$$

This is not an argument tree because the two children nodes are not maximally conservative undercuts. The first undercut is essentially the same argument as the second undercut in a rearranged form (relying on α and β being incompatible, assume one and then conclude that the other doesn't hold). If we replace these by the maximally conservative undercut $\langle\{\neg\alpha \vee \neg\beta\}, \diamond\rangle$, we obtain an argument tree.

The following result is important in practice—particularly in light of corollary 3.3.2 and also other results we present in section 3.6.

Proposition 3.5.1 Argument trees are finite.

The form of an argument tree is not arbitrary. It summarizes all possible courses of argumentation about the argument in the root node. Each node except the root node is the starting point of an implicit series of related arguments. What happens is that for each possible course of argumentation (from the root), an initial sequence is provided as a branch of the tree up to the point that no subsequent counter^n-argument needs a new item in its support (where "new" means not occurring somewhere in that initial sequence). Also, the counterarguments in a course of argumentation may somewhat differ from the ones in the corresponding branch of the argument tree.

Example 3.5.8 The statements about Ms. Shy and Mr. Scoundrel can be captured in $\Delta = \{(\neg r \to \neg p) \to \neg q, (p \to s) \to \neg\neg q, r, s, \neg p\}$. The argument tree with $\langle\{r, (\neg r \to \neg p) \to \neg q\}, \neg q\rangle$ as its root is

$$\begin{array}{ccc}
 & \langle\{r, (\neg r \to \neg p) \to \neg q\}, \neg q\rangle & \\
\nearrow & & \nwarrow \\
\langle\{s, (p \to s) \to \neg\neg q\}, \diamond\rangle & & \langle\{\neg p, (p \to s) \to \neg\neg q\}, \diamond\rangle \\
\uparrow & & \\
\langle\{\neg p, (\neg r \to \neg p) \to \neg q\}, \diamond\rangle & &
\end{array}$$

The left-most branch clearly captures any courses $I, C_1, C, \ldots$ Note that the next element (after C) could be C_2, but that is accounted for because the support of C_2 is a subset of the set-theoretic union of the supports of I, C_1, and C (actually, the resulting set-theoretic union gives Δ). Less immediate is the fact that the left-most branch also captures any courses $I, C_2, C, \ldots$ The key fact is that the support of C_1 is a subset of the support of C_2. What about any courses $I, C_1, I, \ldots$? These are captured through the fact that C_1 being a canonical undercut of I, it happens that appropriately changing just the consequent in I gives a canonical undercut of C_1 (cf. corollary 3.4.3). That is the idea that each node (except for the root) is the starting point of an implicit series of related arguments. Lastly, the right-most branch captures courses that were not touched upon in the earlier discussion of the example.

Example 3.5.9 Let us return to the "chicken and egg dilemma":

Dairyman Egg was first.

Farmer Chicken was first.

Dairyman Egg was first.

Farmer Chicken was first.

... ...

Here are the formulae again:

p	Egg was first.
q	Chicken was first.
r	The chicken comes from the egg.
s	The egg comes from the chicken.
$\neg(p \leftrightarrow q)$	That the egg was first and that the chicken was first are not equivalent.
$r \to \neg q$	The chicken comes from the egg implies that the chicken was not first.
$s \to \neg p$	The egg comes from the chicken implies that the egg was not first.

Thus, $\Delta = \{\neg(p \leftrightarrow q), r \to \neg q, s \to \neg p, r, s\}$. The argument tree with the dairyman's argument as its root is

$$\begin{array}{c} \langle\{r \to \neg q, r, \neg(q \leftrightarrow p)\}, p\rangle \\ \uparrow \\ \langle\{s \to \neg p, s, \neg(q \leftrightarrow p)\}, \Diamond\rangle \end{array}$$

but it does **not** mean that the farmer has the last word nor that the farmer wins the dispute! The argument tree is merely a **representation** of the argumentation (in which the dairyman provides the initial argument).

Although the argument tree is finite, the argumentation here is infinite and unresolved.

3.5.1 Some Useful Subsidiary Definitions

Here we provide a few further definitions that will be useful in the following chapters.

For an argument tree T, $\mathsf{Depth}(T)$ is the length of the longest branch of T, and $\mathsf{Width}(T)$ is the number of leaf nodes in T (see Appendix B for further details on trees). For an argument tree T, $\mathsf{Nodes}(T)$ is the set of nodes (i.e., arguments) in T and $\mathsf{Root}(T)$ is the root of T. We call the argument at the root of an argument tree T, i.e., $\mathsf{Root}(T)$, the **root argument** (or, equivalently, **initiating argument**). For an argument tree T, $\mathsf{Subject}(T)$ is $\mathsf{Claim}(\mathsf{Root}(T))$, and if $\mathsf{Subject}(T)$ is α, we call α the **subject** of T. Given an argument tree T, for an argument A, $\mathsf{Parent}(A)$ is the parent of A in T (i.e., parenthood is defined in the context of a particular argument tree).

For an argument tree T, and an argument A, $\mathsf{Siblings}(T, A)$ is the set of siblings of A in T (i.e., it is the set of children of $\mathsf{Parent}(A)$) and $\mathsf{Undercuts}(T, A)$ is the set of children of A. For an argument tree T, $\mathsf{Siblings}(T)$ is the set of sibling sets in T (i.e., $S \in \mathsf{Siblings}(T)$ iff $S = \mathsf{Undercuts}(T, A)$ for some A in T).

For an argument tree T, each argument in T is either an **attacking argument** or a **defending argument**. If A_r is the root, then A_r is a defending argument. If an argument A_i is a defending argument, then any A_j whose parent is A_i is an attacking argument. If an argument A_j is an attacking argument, then any A_k whose parent is A_j is a defending argument. For an argument tree T, $\mathsf{Defenders}(T)$ is the set of defending arguments in T, and $\mathsf{Attackers}(T)$ is the set of attacking arguments in T.

3.5.2 Technical Developments

The question investigated here is as follows: Looking at the form of argument trees, what can be assessed about Δ?

Inconsistency for Δ was related, as stated in proposition 3.3.13 and corollary 3.3.5, to the existence of a defeater. This can be turned into a relationship between inconsistency for Δ and the case that argument trees consist of a single node as shown in the next result.

Proposition 3.5.2 If Δ is consistent, then all argument trees have exactly one node. The converse is true when each formula in Δ is consistent.

Conversely, looking at some properties about Δ, what can be said about argument trees?

Proposition 3.5.3 Let T be an argument tree whose root node $\langle \Phi, \alpha \rangle$ is such that no subset of Δ is logically equivalent with α. Then, no node in T is a rebuttal for the root.

Example 3.5.10 $\Delta = \{\alpha \wedge \beta \wedge \gamma, \delta \wedge \neg(\alpha \wedge \beta), \neg\alpha \wedge \neg\delta, \delta \to (\beta \wedge \gamma), \alpha \vee \alpha\}$.

$$\begin{array}{c}
\langle \{\alpha \wedge \beta \wedge \gamma\}, \alpha \rangle \\
\uparrow \\
\langle \{\delta \wedge \neg(\alpha \wedge \beta)\}, \neg(\alpha \wedge \beta \wedge \gamma) \rangle \\
\uparrow \\
\langle \{\neg\alpha \wedge \neg\delta\}, \neg(\delta \wedge \neg(\alpha \wedge \beta)) \rangle \\
\uparrow \\
\langle \{\alpha \vee \alpha\}, \neg(\neg\alpha \wedge \neg\delta) \rangle \\
\uparrow \\
\langle \{\delta \wedge \neg(\alpha \wedge \beta), \delta \to (\beta \wedge \gamma)\}, \neg(\alpha \vee \alpha) \rangle
\end{array}$$

The above argument tree contains a rebuttal of the root that is not the child of the root.

Example 3.5.11 $\Delta = \{\neg\alpha \wedge \beta, \neg\alpha \wedge \gamma, \alpha \wedge \delta, \alpha \vee \alpha\}$.

$$\begin{array}{ccc}
 & \langle \{\alpha \wedge \delta\}, \alpha \rangle & \\
\nearrow & & \nwarrow \\
\langle \{\neg\alpha \wedge \beta\}, \neg(\alpha \wedge \delta) \rangle & & \langle \{\neg\alpha \wedge \gamma\}, \neg(\alpha \wedge \delta) \rangle \\
\uparrow & & \uparrow \\
\langle \{\alpha \vee \alpha\}, \neg(\neg\alpha \wedge \beta) \rangle & & \langle \{\alpha \vee \alpha\}, \neg(\neg\alpha \wedge \gamma) \rangle \\
\uparrow & & \uparrow \\
\langle \{\neg\alpha \wedge \gamma\}, \neg(\alpha \vee \alpha) \rangle & & \langle \{\neg\alpha \wedge \beta\}, \neg(\alpha \vee \alpha) \rangle
\end{array}$$

The above argument tree contains rebuttals of the root that are not children of the root.

3.6 Duplicates

Equivalent arguments are arguments that express the same reason for the same point. For undercuts, a more refined notion than equivalent arguments is useful:

Definition 3.6.1 Two undercuts $\langle \Gamma \cup \Phi, \neg\psi \rangle$ and $\langle \Gamma \cup \Psi, \neg\phi \rangle$ are **duplicates** of each other iff ϕ is $\phi_1 \wedge \cdots \wedge \phi_n$ such that $\Phi = \{\phi_1, \ldots, \phi_n\}$ and ψ is $\psi_1 \wedge \cdots \wedge \psi_m$ such that $\Psi = \{\psi_1, \ldots, \psi_m\}$.

Duplicates introduce a symmetric relation that fails to be transitive (and reflexive). Arguments that are duplicates of each other are essentially the same argument in a rearranged form.

Example 3.6.1 The two arguments below are duplicates of each other:

$\langle\{\alpha, \neg\alpha \vee \neg\beta\}, \neg\beta\rangle$

$\langle\{\beta, \neg\alpha \vee \neg\beta\}, \neg\alpha\rangle$

Example 3.6.2 To illustrate the lack of transitivity in the duplicate relationship, the following two arguments are duplicates

$\langle\{\gamma, \alpha, \alpha \wedge \gamma \rightarrow \neg\beta\}, \neg\beta\rangle$

$\langle\{\gamma, \beta, \alpha \wedge \gamma \rightarrow \neg\beta\}, \neg\alpha\rangle$

and similarly the following two arguments are duplicates

$\langle\{\gamma, \beta, \alpha \wedge \gamma \rightarrow \neg\beta\}, \neg\alpha\rangle$

$\langle\{\alpha, \alpha \wedge \gamma \rightarrow \neg\beta\}, \neg(\beta \wedge \gamma)\rangle$

but the following two are not duplicates:

$\langle\{\alpha, \alpha \wedge \gamma \rightarrow \neg\beta\}, \neg(\beta \wedge \gamma)\rangle$

$\langle\{\gamma, \alpha, \alpha \wedge \gamma \rightarrow \neg\beta\}, \neg\beta\rangle$

The following proposition shows how we can systematically obtain duplicates. In this result, we see there is an explosion of duplicates for each maximally conservative undercut. This obviously is a potential concern for collating counterarguments.

Proposition 3.6.1 For every maximally conservative undercut $\langle\Psi, \beta\rangle$ to an argument $\langle\Phi, \alpha\rangle$, there exist at least $2^m - 1$ arguments, each of which undercuts the undercut (m is the size of Ψ). Each of these $2^m - 1$ arguments is a duplicate to the undercut.

Proposition 3.6.2 No two maximally conservative undercuts of the same argument are duplicates.

Corollary 3.6.1 No two canonical undercuts of the same argument are duplicates.

Proposition 3.6.3 No branch in an argument tree contains duplicates, except possibly for the child of the root to be a duplicate to the root.

These last two results are important. They show that argument trees are an efficient and lucid way of representing the pertinent counterarguments to each argument: Corollary 3.6.1 shows it regarding breadth, and proposition 3.6.3 shows it regarding depth. Moreover, they show that the intuitive need to eliminate duplicates from argument trees is taken care of through an efficient syntactical criterion (condition 2 of definition 3.5.1).

3.7 Argument Structures

We now consider how we can gather argument trees for and against a claim. To do this, we define argument structures.

Definition 3.7.1 An **argument structure** for a formula α is a pair of sets $\langle \mathscr{P}, \mathscr{C} \rangle$, where $\mathscr{P}$ is the set of argument trees for α and $\mathscr{C}$ is the set of argument trees for $\neg\alpha$.

Example 3.7.1 Let $\Delta = \{\alpha \vee \beta, \alpha \to \gamma, \neg\gamma, \neg\beta, \delta \leftrightarrow \beta\}$. For this, we obtain three argument trees for the argument structure for $\alpha \vee \neg\delta$.

$$\begin{tikzcd}
\langle \{\alpha \vee \beta, \neg\beta\}, \alpha \vee \neg\delta \rangle & \langle \{\delta \leftrightarrow \beta, \neg\beta\}, \alpha \vee \neg\delta \rangle \\
\langle \{\alpha \to \gamma, \neg\gamma\}, \Diamond \rangle \arrow[u] & \langle \{\alpha \vee \beta, \alpha \to \gamma, \neg\gamma\}, \Diamond \rangle \arrow[u]
\end{tikzcd}$$

$$\begin{tikzcd}
\langle \{\alpha \vee \beta, \alpha \to \gamma, \neg\gamma, \delta \leftrightarrow \beta\}, \neg\alpha \wedge \delta \rangle \\
\langle \{\neg\beta\}, \Diamond \rangle \arrow[u]
\end{tikzcd}$$

Example 3.7.2 Let $\Delta = \{\alpha \leftrightarrow \beta, \beta \vee \gamma, \gamma \to \beta, \neg\alpha \vee \neg\beta \vee \neg\gamma, \gamma \wedge \delta, \neg\delta\}$. From this, we obtain the following argument trees for and against $\beta \wedge \delta$, where $\Phi_1 = \{\alpha \leftrightarrow \beta, \neg\alpha \vee \neg\beta \vee \neg\gamma, \gamma \wedge \delta\}$ and $\Phi_2 = \{\beta \vee \gamma, \gamma \to \beta, \alpha \leftrightarrow \beta, \neg\alpha \vee \neg\beta \vee \neg\gamma\}$.

$$\begin{tikzcd}
 & \langle \{\gamma \wedge \delta, \gamma \to \beta\}, \beta \wedge \delta \rangle & \\
\langle \Phi_1, \Diamond \rangle \arrow[ur] & \langle \{\neg\delta\}, \Diamond \rangle \arrow[u] & \langle \Phi_2, \Diamond \rangle \arrow[ul] \\
\langle \{\neg\delta\}, \Diamond \rangle \arrow[u] & &
\end{tikzcd}$$

$$\begin{tikzcd}
 & \langle \{\alpha \leftrightarrow \beta, \neg\alpha \vee \neg\beta \vee \neg\gamma, \gamma \wedge \delta\}, \neg(\beta \wedge \delta) \rangle & \\
\langle \{\gamma \to \beta\}, \Diamond \rangle \arrow[ur] & & \langle \{\neg\delta\}, \Diamond \rangle \arrow[ul]
\end{tikzcd}$$

$$\begin{tikzcd}
\langle \{\neg\delta\}, \neg(\beta \wedge \delta) \rangle \\
\langle \{\gamma \wedge \delta\}, \Diamond \rangle \arrow[u]
\end{tikzcd}$$

Proposition 3.7.1 Let $\langle \mathscr{P}, \mathscr{C} \rangle$ be an argument structure. If there exists an argument tree in $\mathscr{P}$ that has exactly one node, then $\mathscr{C}$ is the empty set. The converse is untrue, even when assuming that $\mathscr{P}$ is nonempty.

Example 3.7.3 Let $\Delta = \{\alpha \vee \neg\beta, \beta, \neg\beta\}$. In the argument structure $\langle \mathscr{P}, \mathscr{C} \rangle$ for α, we have that $\mathscr{C}$ is the empty set while $\mathscr{P}$ contains an argument tree that has more than one node:

$$\begin{array}{c} \langle \{\alpha \vee \neg\beta, \beta\}, \alpha \rangle \\ \uparrow \\ \langle \{\neg\beta\}, \diamond \rangle \end{array}$$

Example 3.7.3 illustrates the last sentence in proposition 3.7.1. If Δ is augmented with $\alpha \wedge \gamma$, for instance, then $\langle \mathscr{P}, \mathscr{C} \rangle$ is such that $\mathscr{P}$ contains both an argument tree with more than one node and an argument tree consisting of just a root node.

Proposition 3.7.2 Let $\langle \mathscr{P}, \mathscr{C} \rangle$ be an argument structure. If Δ is consistent, then each argument tree in $\mathscr{P}$ has exactly one node and $\mathscr{C}$ is the empty set. The converse is untrue, even when assuming that $\mathscr{P}$ is nonempty and that each formula in Δ is consistent.

The last sentence in the statement of proposition 3.7.2 can be illustrated by the following counterexample.

Example 3.7.4 Let $\Delta = \{\alpha, \beta, \neg\beta\}$. The argument structure $\langle \mathscr{P}, \mathscr{C} \rangle$ for α is such that $\mathscr{P}$ contains a single argument tree consisting of just the root node below:

$$\langle \{\alpha\}, \alpha \rangle$$

In argument structures, $\mathscr{P}$ and $\mathscr{C}$ are symmetrical. Any property enjoyed by one has a counterpart, which is a property enjoyed by the other: Both are the same property, with $\mathscr{P}$ and $\mathscr{C}$ exchanged. For example, we have the result similar to proposition 3.7.1 stating that if there exists an argument tree in $\mathscr{C}$ that has exactly one node, then $\mathscr{P}$ is the empty set. Symmetry goes even deeper, inside the argument trees of $\mathscr{P}$ and $\mathscr{C}$. This is exemplified in the next result.

Proposition 3.7.3 Let $\langle [X_1, \ldots, X_n], [Y_1, \ldots, Y_m] \rangle$ be an argument structure. For any i and any j, the support of the root node of Y_j (resp. X_i) is a superset of the support of a canonical undercut for the root node of X_i (resp. Y_j).

Proposition 3.7.3 is reminiscent of the phenomenon reported in corollary 3.3.1.

Proposition 3.7.4 Let $\langle \mathscr{P}, \mathscr{C} \rangle$ be an argument structure. Then, both $\mathscr{P}$ and $\mathscr{C}$ are finite.

3.8 First-Order Argumentation

In many situations, it is apparent that there is a need to support first-order argumentation. As an example, consider a senior clinician in a hospital who may need to consider the pros and cons of a new drug regime in order to decide whether to incorporate the regime as part of hospital policy: This could be expedited by considering the pros and cons of a first-order statement formalizing that piece of policy.

As another example, consider an information systems consultant who is collating requirements from users within an organization. Due to conflicts between requirements from different users, the consultant may need to consider arguments for and against particular requirements being adopted in the final requirements specification. Towards this end, first-order statements provide a format for readily and thoroughly capturing constraints and compromises.

It is important to notice that using a propositional framework to encode first-order statements leads to mishaps, for example, when attempting to mimic $\forall x.\alpha[x]$ by means of its instances $\alpha[t]$ for all ground elements t in the universe of discourse: Due to circumstantial properties, it may happen that, whatever t, a particular argument for $\alpha[t]$ can be found but there is no guarantee that an argument for $\forall x.\alpha[x]$ exists. Here is an example. Consider the statements "if x satisfies p and q, then x satisfies r or s" and "if x satisfies q and r and s, then x satisfies t." Clearly, these do not entail the statement "if x satisfies p, then x satisfies t." Assume the set of all ground terms from the knowledgebase is $\{a, b\}$. The obvious idea is to consider $\forall x.\alpha[x]$ as being equivalent with both instances $\alpha[a]$ and $\alpha[b]$. Unfortunately, should $q(a)$ and $r(a) \vee s(a) \to t(a)$ be incidentally the case as well as $s(b)$ and $p(b) \to q(b) \wedge r(b)$, then "if x satisfies p then x satisfies t" would be regarded as argued for! The moral is that a propositional approach here cannot be substituted for a first-order one. In such situations, a first-order approach cannot be dispensed with.

To address this need for first-order argumentation, we generalize our proposal from the propositional case to the first-order case. For a first-order language $\mathscr{L}$, the set of formulae that can be formed is given by the usual inductive definitions for classical logic: Roman symbols $p, q, \ldots$ denote predicates, Greek symbols $\alpha, \beta, \ldots$ denote formulae.

All the definitions for argument, counterargument, rebuttal, undercut, maximally conservative undercut, canonical undercut, and argument tree are the same as for the propositional case, except that we assume Δ is a set of first-order formulae and that $\vdash$ is the first-order consequence relation. Given that this migration from the propositional case to the first-order case is straightforward, we do not repeat any of the definitions, but instead we just provide some examples to illustrate the use of these definitions in the first-order case.

Example 3.8.1 Consider the following knowledgebase:

$$\Delta = \{\forall x.(p(x) \to q(x) \vee r(x)), p(a), \neg\forall x.s(x), \neg\exists x.r(x),$$
$$\neg\exists x.(p(x) \to q(x) \vee r(x))\}$$

Some arguments from the knowledgebase are listed below:

$\langle\{p(a), \forall x.(p(x) \to q(x) \vee r(x))\}, q(a) \vee r(a)\rangle$
$\langle\{\neg\forall x.s(x)\}, \neg\forall x.s(x)\rangle$
$\langle\{\neg\exists x.r(x)\}, \forall x.\neg r(x)\rangle$

Example 3.8.2 Given Δ as in example 3.8.1, the first argument (below) is a more conservative argument than the second:

$\langle\{p(a), \forall x.(p(x) \to q(x) \vee r(x))\}, q(a) \vee r(a)\rangle$
$\langle\{p(a), \forall x.(p(x) \to q(x) \vee r(x)), \neg\exists x.r(x)\}, q(a)\rangle$

Example 3.8.3 Again, Δ is as in example 3.8.1. It is easy to find an undercut for the argument $\langle\{p(a), \forall x.(p(x) \to q(x) \vee r(x))\}, q(a) \vee r(a)\rangle$; an obvious one is $\langle\{\neg\exists x.(p(x) \to q(x) \vee r(x))\}, \neg\forall x.(p(x) \to q(x) \vee r(x))\rangle$. Now, there is another one, which actually is more conservative: $\langle\{\neg\exists x.(p(x) \to q(x) \vee r(x))\}, \neg(p(a) \wedge \forall x.(p(x) \to q(x) \vee r(x)))\rangle$.

Example 3.8.4 Given an appropriate Δ and provided the conditions for definition 3.2.1 are met, we have the general cases below:

$\langle\{\forall x.\alpha[x]\}, \alpha[a]\rangle$ is undercut by $\langle\{\neg\exists x.\alpha[x]\}, \neg\forall x.\alpha[x]\rangle$
$\langle\{\forall x.\alpha[x]\}, \alpha[a]\rangle$ is undercut by $\langle\{\exists x.\neg\alpha[x]\}, \neg\forall x.\alpha[x]\rangle$
$\langle\{\forall x.\alpha[x]\}, \alpha[a]\rangle$ is undercut by $\langle\{\neg\alpha[b]\}, \neg\forall x.\alpha[x]\rangle$
$\langle\{\forall x.\alpha[x]\}, \alpha[a]\rangle$ is undercut by $\langle\{\neg\alpha[c]\}, \neg\forall x.\alpha[x]\rangle$

Example 3.8.5 If Δ is as in example 3.8.1, a maximally conservative undercut for the first argument below is the second argument.

$\langle\{p(a), \forall x.(p(x) \to q(x) \vee r(x))\}, q(a) \vee r(a)\rangle$
$\langle\{\neg\exists x.(p(x) \to q(x) \vee r(x))\}, \neg(p(a) \wedge \forall x.(p(x) \to q(x) \vee r(x)))\rangle$

Example 3.8.6 Let Δ be as in example 3.8.1. A complete argument tree for $q(a)$ is as follows:

$$\begin{array}{c} \langle\{p(a), \forall x.(p(x) \rightarrow q(x) \vee r(x)), \neg\exists x.r(x)\}, q(a)\rangle \\ \uparrow \\ \langle\{\neg\exists x.(p(x) \rightarrow q(x) \vee r(x))\}, \diamond\rangle \end{array}$$

Example 3.8.7 Let $\Delta = \{\forall x.(p(x) \wedge q(x)), \exists x.\neg(p(x) \wedge q(x)), \forall x.(r(x) \wedge \neg p(x)), \forall x \forall y.s(x, y), \forall x.(s(x, f(x)) \rightarrow \neg r(x))\}$. An argument tree for $\forall x.q(x)$ is as follows:

$$\begin{tikzcd}[column sep=tiny] & \langle\{\forall x.(p(x) \wedge q(x))\}, \forall x.q(x)\rangle & \\ \langle\{\exists x.\neg(p(x) \wedge q(x))\}, \diamond\rangle \arrow[ur] & & \langle\{\forall x.(r(x) \wedge \neg p(x))\}, \diamond\rangle \arrow[ul] \\ & & \langle\{\forall x \forall y.s(x, y), \forall x.(s(x, f(x)) \rightarrow \neg r(x))\}, \diamond\rangle \arrow[u] \end{tikzcd}$$

Thus, as in the propositional case, an argument tree is an efficient representation of the counterarguments, counter-counterarguments, $\ldots$, in the first-order case.

Proposition 3.8.1 Let $\alpha \in \mathscr{L}$. If Δ is finite, there are a finite number of argument trees with the root being an argument with consequent α that can be formed from Δ, and each of these trees has finite branching and a finite depth.

From these examples, and result, we see that we can straightforwardly use our framework for the first-order case. All the results we have presented for the propositional can be immediately generalized to the first-order case.

We finish this section with a larger example of first-order reasoning.

Example 3.8.8 Consider the following literals concerning a country called `Landia`:

$n1$ `ReportDate(Landia, 30May2000)`

$n2$ `Government(Landia, unstable)`

$n3$ `Democracy(Landia, strong)`

$n4$ `PublicSpending(Landia, excessive)`

$n5$ `OilExport(Landia, significant)`

$n6$ `OilPrice(Landia, increasing)`

$n7$ `LastElection(Landia, recent)`

$n8$ `Currency(Landia, strong)`

Consider also the following general knowledge about countries:

$d1$ $\forall$`X.Government(X, unstable)` $\rightarrow$ `credit-risk(X)`

$d2$ $\forall$`X.Democracy(X, strong)` $\wedge$ `LastElection(X, recent)` $\rightarrow \neg$`Government(X, unstable)`

$d3$ $\forall$`X.PublicSpending(X, excessive)` $\rightarrow$ `credit-risk(X)`

$d4$ $\forall$`X.OilExport(X, significant)` $\wedge$ `OilPrice(X, increasing)` $\rightarrow \neg$`PublicSpending(X, excessive)`

$d5$ $\forall$`X.Currency(X, strong)` $\rightarrow \neg$`credit-risk(X)`

Now assume the knowledgebase to be $\Delta = \{n1, \ldots, n8, d1, \ldots, d5\}$. Note, for a more lucid presentation, we use the labels for the formulae (rather than the formulae themselves) in the supports of the arguments. From this, we obtain the following argument trees for and against the inference `credit-risk(Landia)`.

$$\begin{array}{ccc}
 & \langle\{n2, d1\}, \texttt{credit-risk(Landia)}\rangle & \\
\nearrow & & \nwarrow \\
\langle\{n3, n7, d2\}, \Diamond\rangle & & \langle\{n8, d5\}, \Diamond\rangle \\
 & & \uparrow \\
 & & \langle\{n4, d3\}, \Diamond\rangle \\
 & & \uparrow \\
 & & \langle\{n5, n6, d4\}, \Diamond\rangle
\end{array}$$

$$\begin{array}{ccc}
 & \langle\{n4, d3\}, \texttt{credit-risk(Landia)}\rangle & \\
\nearrow & & \nwarrow \\
\langle\{n5, n6, d4\}, \Diamond\rangle & & \langle\{n8, d5\}, \Diamond\rangle \\
 & & \uparrow \\
 & & \langle\{n2, d1\}, \Diamond\rangle \\
 & & \uparrow \\
 & & \langle\{n3, n7, d2\}, \Diamond\rangle
\end{array}$$

$$\begin{array}{ccc}
 & \langle\{n8, d5\}, \neg\texttt{credit-risk(Landia)}\rangle & \\
\nearrow & & \nwarrow \\
\langle\{n2, d1\}, \Diamond\rangle & & \langle\{n4, d3\}, \Diamond\rangle \\
\uparrow & & \uparrow \\
\langle\{n3, n7, d2\}, \Diamond\rangle & & \langle\{n5, n6, d4\}, \Diamond\rangle
\end{array}$$

Given the simple nature of the knowledgebase in the above example, we see that each tree is a different arrangement of the same set of arguments. However, we stress that each tree is a stand-alone representation of the set of arguments from a particular perspective and so it is necessary to have all these trees.

3.9 Discussion

In this chapter, we have proposed a framework for modeling argumentation. The key features of this framework are the clarification of the nature of arguments and counterarguments; the identification of canonical undercuts, which we argue are the only undercuts that we need to take into account; and the representation of argument trees and argument structures that provide a way of exhaustively collating arguments and counterarguments.

Distinct but logically equivalent supports give rise to different canonical undercuts that all have to occur in a complete argument tree whose root node is attacked by these supports. The reader may wonder what the rationale is here, as in the following case, where $A_2, \ldots, A_8$ are canonical undercuts of A_1:

$$
\begin{array}{ccccccccc}
 & & & & A_1 & & & & \\
 & \nearrow & & & & & & \nwarrow & \\
A_2 & & \nearrow & & & & \nwarrow & & A_3 \\
 & A_4 & & \nearrow & \uparrow & \nwarrow & & A_5 & \\
 & & A_6 & & A_7 & & A_8 & &
\end{array}
$$

where

$A_1 = \langle\{\alpha \vee \beta\}, \alpha \vee \beta\rangle$

$A_2 = \langle\{\neg\alpha \wedge \neg\beta\}, \Diamond\rangle$

$A_3 = \langle\{\neg\beta \wedge \neg\alpha\}, \Diamond\rangle$

$A_4 = \langle\{\neg(\beta \leftrightarrow \neg\alpha), \alpha \wedge \beta \rightarrow \neg\alpha \vee \neg\beta\}, \Diamond\rangle$

$A_5 = \langle\{\neg\alpha \vee \neg\beta, \alpha \leftrightarrow \beta\}, \Diamond\rangle$

$A_6 = \langle\{\alpha \vee \beta \rightarrow \neg\alpha \wedge \neg\beta\}, \Diamond\rangle$

$A_7 = \langle\{\neg\alpha, \neg\beta\}, \Diamond\rangle$

$A_8 = \langle\{\neg(\alpha \vee \beta)\}, \Diamond\rangle$

There are so many (infinitely many, in fact) *logically equivalent* ways to refute $\alpha \vee \beta$, so why spell out many counterarguments that are so close to one another? Here is an explanation. An argument tree is intended to be exhaustive in recording the ways the argument can *actually* be challenged, but that does not mean that the argument tree lists all the ways suggested by classical logic: Remember that an argument must have a subset of Δ as its support. And Δ is *not* closed under logical equivalence! Thus, only those logically equivalent supports that are *explicitly* mentioned by means of Δ give rise to arguments to be included in an argument tree. That makes a big difference, and it is where the rationale stands: If logically equivalent forms have been explicitly provided, it must be for some reason. For example, the above tree is not to be an argument tree unless there were some good reason to have all these many variants of $\{\neg\alpha, \neg\beta\}$ in Δ.

If the argument tree is used for presenting the arguments and counterarguments to a user, then the user would only want to see those arguments that have a good reason to be there. Redundant arguments are unlikely to be welcomed by the user. Should an argument tree eventually to be used when doing some kind of evaluation based on reinforcement, distinct albeit logically equivalent evidence may prove useful (e.g., if a surgeon has two distinct arguments for undertaking a risky operation, these arguments would reinforce each other), whereas statements that merely happen to be logically equivalent should not be included in the constellation.

Superficially, an argument structure could be viewed as an argument framework in Dung's system. An argument in an argument tree could be viewed as an argument in a Dung argument framework, and each arc in an argument tree could be viewed as an attack relation. However, the way sets of arguments are compared is different.

Some differences between Dung's approach and our approach can be seen in the following examples.

Example 3.9.1 Consider a set of arguments $\{a_1, a_2, a_3, a_4\}$ with the attack relation $\mathscr{R}$ such that $a_2 \mathscr{R} a_1$, $a_3 \mathscr{R} a_2$, $a_4 \mathscr{R} a_3$, and $a_1 \mathscr{R} a_4$. Here there is an admissible set $\{a_1, a_3\}$. We can try to construct an argument tree with a_1 at the root. As a counterpart to the attack relation, we regard that a_1 is undercut by a_2, a_2 is undercut by a_3, and so on. However, the corresponding sequence of nodes a_1, a_2, a_3, a_4, a_1 is not an argument tree because a_1 occurs twice in the branch (violating condition 2 of definition 3.5.1). Thus, the form of the argument tree for a_1 fails to represent the fact that a_1 attacks a_4.

Example 3.9.2 Let $\Delta = \{\beta, \beta \to \alpha, \delta \wedge \neg\beta, \neg\delta \wedge \neg\beta\}$, giving the following argument tree for α:

$$
\begin{array}{ccc}
 & \langle\{\beta, \beta \to \alpha\}, \alpha\rangle & \\
\nearrow & & \nwarrow \\
\langle\{\delta \wedge \neg\beta\}, \Diamond\rangle & & \langle\{\neg\delta \wedge \neg\beta\}, \Diamond\rangle \\
\uparrow & & \uparrow \\
\langle\{\neg\delta \wedge \neg\beta\}, \Diamond\rangle & & \langle\{\delta \wedge \neg\beta\}, \Diamond\rangle
\end{array}
$$

For this, let a_1 be $\langle\{\beta, \beta \to \alpha\}, \alpha\rangle$, a_2 be $\langle\{\delta \wedge \neg\beta\}, \Diamond\rangle$ and a_3 be $\langle\{\neg\delta \wedge \neg\beta\}, \Diamond\rangle$. Disregarding the difference between the occurrences of $\Diamond$, this argument tree rewrites as $a_2\mathcal{R}a_1$, $a_3\mathcal{R}a_1$, $a_3\mathcal{R}a_2$, and $a_2\mathcal{R}a_3$, where a_1 denotes the root node $\langle\{\beta, \beta \to \alpha\}, \alpha\rangle$. In this argument tree, each defeater of the root node is defeated. Yet no admissible set of arguments contains a_1.

Finally, we can consider capturing a class of arguments that fail to be deductive. In other words, we can revisit one of the basic assumptions made at the start of this chapter. For this, the basic principle for our approach still applies: An argument comes with a claim, which relies on reasons by virtue of some given relationship between the reasons and the claim. Thus, arguments can still be represented by pairs, but the relationship is no longer entailment in classical logic; it is a binary relation of some kind capturing "tentative proofs" or "proofs using nonstandard modes of inference" instead of logical proofs.

This relationship can be taken to be almost whatever pleases you provided that you have a notion of consistency. Observe that this does not mean that you need any second element of a pair to stand for "absurdity": You simply have to specify a subset of the pairs to form the cases of inconsistency. Similarly, our approach is not necessarily restricted to a logical language, and another mode of representation can be chosen.

3.10 Bibliographic Notes

This chapter is based on a particular proposal for logic-based argumentation for the propositional case [BH00, BH01] and generalized to the first-order case [BH05]. We have compared this proposal with abstract argumentation and shown that by having a logic-based notion of argumentation, we can have a much deeper understanding of the individual arguments and of the counterarguments that impact on each argument. Furthermore, by introducing logical knowledgebases, we can automatically construct individual arguments and constellations of arguments and counterarguments.

This proposal is not the first proposal for logic-based argumentation. However, most proposals are not based on classical logic. Since we believe that understanding argumentation in terms of classical logic is an ideal starting point for understanding the elements of argumentation, we have focused on classical logic.

In comparison with the previous proposals based on classical logic (e.g., [AC98, Pol92]), our proposal provides a much more detailed analysis of counterarguments, and ours is the first proposal to consider canonical undercuts. Canonical undercuts are a particularly important proposal for ensuring that all the relevant undercuts for an argument are presented, thereby ensuring that a constellation of arguments and counterarguments is exhaustive, and yet ensuring that redundant arguments are avoided from this presentation.

We leave a more detailed comparison with other proposals for logic-based argumentation (both proposals based on classical logic and proposals based on defeasible logic) until chapter 8.

4 Practical Argumentation

So far in this book, we have presented an outline of monological argumentation as an important cognitive process for dealing with conflicting information, and we have presented abstract argumentation and logical argumentation as ways of formalizing monological argumentation. In this chapter, we consider how well these formalisms, as presented so far, meet the needs of capturing monological argumentation. We will show that for "practical argumentation" we need further developments.

We will start by considering examples of practical arguments such as presented by journalists, politicians, and scientists, and we will use these examples to assess some of the shortcomings of the formal approaches presented in chapters 2 and 3 for practical argumentation.

As a result, this chapter motivates the need for being selective (in the sense of choosing some arguments in preference to others) and formative (in the sense of shaping and molding arguments) in argumentation, and thus for using the better practical arguments based on (1) taking intrinsic aspects of the arguments into account such as their relative consistency, the exhaustiveness of the consideration of counterarguments, and the relative similarities between arguments and (2) taking extrinsic factors into account such as impact on the audience and how convincing the arguments are to the audience. Therefore, this chapter will provide the setting for much of the rest of the book.

4.1 Argumentation for Thinking about Questions

Factual argumentation is the simplest form of monological argumentation in that it uses only objective information for the support of each argument, and the aim of the argumentation is to inform the audience about the key arguments and counterarguments, perhaps as a precursor to making some decision. To explore the notion of factual argumentation, in a practical context, we will start by considering the following extended example, which is based on an article that equivocates on the question of whether government proposals for charging fees to university students are a good idea. While the article is discussing the situation in England and Wales at that time, the issues being discussed would be familiar to a reader from a university in most countries, and we believe that there is no further background that is necessary to use this article for our purposes in this chapter.

Example 4.1.1 The following are quotes from an editorial from The Sunday Times of 26 January 2003. The overall tone of the article is relatively neutral, as it does not appear to push for any particular outcome. It centers on the question of whether "the government proposals are good," and it does not draw an overall conclusion on the question, though it does raise some issues and doubts with some of the details. Overall it attempts to lay out facts, observations, and proposals by others. The information presented can be described as being objective in that it concerns propositions that can be observed and verified in the world.

> In proposing a shake-up of the universities and raising tuition fees to £3000 a year, Charles Clark ... [the] education secretary has warned ... students ... that the days of universities as a taxpayer-subsidised "finishing school" are over.
>
> Most Labour MPs ... balk at a system which will make tomorrow's higher earners pay more of their university costs. They believe the fees will discourage students in poorer families from going to university and that it will create a two-tier system, making the top colleges even stronger bastions of the middle class. To appease this old Labour instinct, Mr Clarke has thrown in an "access regulator".... [to] encourage universities to lobby poorer students to apply.
>
> Seeing that most universities are only too delighted to take poorer students if they meet entry requirements, the obvious implications is that universities will themselves be forced into a two-tier entrance procedure. Students from poor backgrounds and bad schools will have their A-level results "weighted" to favour them against middle-class students from selective state or independent schools. In a few cases this may be justified, especially if during the interview a student from a deprived background shows an intelligence and potential not demonstrated in his or her exam results. Otherwise it is profoundly unjust and must open the government up to a challenge under human rights legislation.

By paraphrasing the above and splitting it into the following propositional letters, we can formalize the above text. The abstraction is meant to capture the key points in the article, though no specific methodology has been used to undertake the formalization. We use the letters $p_1, \ldots, p_{12}$ to denote these propositional letters and give the corresponding meaning below:

p_1 Universities cannot continue as finishing schools using taxpayers' money.

p_2 Students need to pay towards their university education.

p_3 It is ok to charge a student 3,000 pounds per year for tuition fees.

p_4 The government proposals are good.

p_5 Charging tuition fees will discourage students from poorer families.

p_6 There will be a two-tier university system.

p_7 Top universities will become more middle-class.

p_8 There will be a "regulator" to check universities take enough poorer students.

p_9 Universities will increase the proportion of poorer students.

p_{10} Universities will apply positive discrimination to poorer students.

p_{11} Positive discrimination violates European human rights legislation.

p_{12} Universities are already keen to recruit students from poorer families.

We can now compose the above propositions into the following set of formulae. These formulae are meant to capture the key knowledge in the article, though again no specific methodology has been used to undertake the formalization.

$$\left\{\begin{array}{ccc} p_1 \rightarrow p_2 & p_2 \rightarrow p_3 & p_3 \rightarrow p_4 \\ p_5 \rightarrow p_6 & p_6 \rightarrow p_7 & p_7 \rightarrow \neg p_4 \\ p_8 \rightarrow p_9 & p_9 \rightarrow \neg p_7 & p_{12} \rightarrow \neg p_9 \\ p_8 \rightarrow p_{10} & p_{10} \rightarrow p_{11} & p_{11} \rightarrow \neg p_9 \end{array}\right\}$$

Let Δ be the above implicational formulae together with the set of atomic propositions $\{p_1, p_5, p_8, p_{12}\}$. From Δ, we can now construct the following argument trees for p_4:

$$\begin{array}{c} \langle\{p_1, p_1 \rightarrow p_2, p_2 \rightarrow p_3, p_3 \rightarrow p_4\}, p_4\rangle \\ \uparrow \\ \langle\{p_5, p_5 \rightarrow p_6, p_6 \rightarrow p_7, p_7 \rightarrow \neg p_4\}, \Diamond\rangle \\ \uparrow \\ \langle\{p_8, p_8 \rightarrow p_9, p_9 \rightarrow \neg p_7\}, \Diamond\rangle \\ \nearrow \qquad\qquad \nwarrow \\ \langle\{p_{12}, p_{12} \rightarrow \neg p_9\}, \Diamond\rangle \quad \langle\{p_8, p_8 \rightarrow p_{10}, p_{10} \rightarrow p_{11}, p_{11} \rightarrow \neg p_9\}, \Diamond\rangle \end{array}$$

This shows how an example of factual argumentation can be identified in a newspaper article and that the example can be formalized in propositional logic. Obviously there are subtleties in the original article that are lost in this translation into propositional logic. However, the main points are clearly represented in the formal representation, and moreover, the key arguments and counterarguments are formalized explicitly.

The main observation that we want to draw from this example is that the information in the article is only a small subset of the information that the writer would have available on the subject of funding of universities, and hence the writer has only presented a small subset of the possible arguments that could be put forward on the subject. In other words, from this example, we can see that the writer has been selective in the argumentation. This, we believe, is a common theme in monological argumentation. It is something we want to discuss further in this chapter, and we present techniques for characterizing and automating it in subsequent chapters.

Selectivity in argumentation is driven by a number of factors. First, it is often completely impractical to represent all the possible arguments and counterarguments. In the example above, if it were exhaustive in terms of what the writer knew about the subject, then the article could run to tens or even hundreds of pages. This length would not be feasible for a newspaper article. Second, many arguments are very similar in some respects, and they can be summarized using a simpler and more generic argument that ignores the subtleties and the details. Third, readers do not want and/or are unable to read large constellations of arguments. They want summaries, and they do not want to be bored. This is particularly important in the case of newspaper articles.

4.2 Argumentation That Raises Further Questions

Factual argumentation is often used to lay out collated knowledge as a step to better understanding that knowledge, and perhaps as a precursor to drawing an inference. The next example is also an article that considers a question, in this case that of whether mobile phones are safe, and in so doing maintains an ambivalence. The net message in the article is that it is still unclear whether they are safe. However, in the process of acknowledging the ambivalence, it raises further questions for further investigation. These questions are not intended to be rhetorical questions but rather questions that should be addressed if we are going to make progress on the question of whether mobile phones are safe. The questions are information in their own right. The journalist has added value to the collated information by identifying these questions. This is interesting in that it reflects a very important and common way that conflicting information is handled in cognition: If we have a conflict in our information that we cannot resolve with the available information, and the conflict is of sufficient importance, then we seek further information to help us resolve the conflict.

Example 4.2.1 The following are quotes from an article from The Economist of 15 January 2005. The overall tone of this article is also neutral, like the previous example, and it does not draw an overall conclusion on the question. Rather it just attempts to lay out facts, observations, and opinions by others.

> "There is as yet no hard evidence of adverse health effects on the general public" Sir William Stewart, chairman of the National Radiological Protection Board,... "But because of the current uncertainties we recommend a continued precautionary approach to the use of mobile-phone technologies."
>
> One possible danger identified by the report concerns mobiles with a high specific energy absorption rate (SAR), a measure of the energy the head absorbs while the user is talking. Different phones have different SAR values, and the report recommends choosing a phone with a low one. But the SAR also depends on things like how close a user is to mast (the further the mast is, the more your brain gets cooked) and how garrulous the user is (people who listen rather than talk will have cooler skulls). One way to reduce the SAR is to have lots of masts. But people tend to think they are dangerous too, so that would be unpopular.
>
> Even when a phone does heat up its user's head, it is not clear what is particularly bad about that ..... humans survive warm baths. A Swedish study has shown an increase in acoustic neuromas (a non-malignant tumour of the nerve that supplies the ear) in people who have used mobiles for ten years. But critics point out that the study has not been replicated, and that old mobiles might have had different effects from current ones. Even so, the nervous will be relieved to know that a hands-free kit reduces the SAR by half.
>
> Anyone who wants a definitive answer will probably have to wait a long time.

From this text, we can identify the following atoms that capture some of the important propositions being discussed in the article. We use the propositional letters $q_1, \ldots, q_8$ to denote these atoms and give the corresponding meaning below:

q_1 There is evidence that mobile phones have an adverse health effect.

q_2 Mobile phones are safe.

q_3 Mobile phones have problems with SAR.

q_4 There is a high density of mobile phone masts.

q_5 Hot baths heat the brain.

q_6 Mobile phones heat the brain.

q_7 Mobile phones can be used with a hands-free kit.

q_8 Mobile phone masts are unpopular.

Using these propositional letters, we can identify the following set of implicational formulae, and the propositions $\{\neg q_1, q_3, q_4, q_5, q_7, q_8\}$, which together make up the knowledgebase Δ, and which reflects the knowledge being used in the arguments in the article.

$$\{q_7 \to \neg q_3, \neg q_2 \to q_1, q_3 \to q_6, q_6 \to q_1, q_4 \to \neg q_3, q_5 \to \neg(q_6 \to q_1),$$
$$q_8 \to \neg q_4\}$$

Using these formulae, we can construct an argument tree as follows for the claim q_2.

$$\begin{array}{ccc}
 & \langle\{\neg q_1, \neg q_2 \to q_1\}, q_2\rangle & \\
 & \uparrow & \\
 & \langle\{q_3, q_3 \to q_6, q_6 \to q_1\}, \diamond\rangle & \\
\nearrow & \uparrow & \nwarrow \\
\langle\{q_4, q_4 \to \neg q_3\}\diamond\rangle & \langle\{q_7, q_7 \to \neg q_3\}, \diamond\rangle & \langle\{q_5, q_5 \to \neg(q_6 \to q_1)\}, \diamond\rangle \\
\uparrow & & \\
\langle\{q_8, q_8 \to \neg q_4\}, \diamond\rangle & &
\end{array}$$

The above example further supports our view that monological argumentation as presented in articles can be translated into argument trees. While there is considerable latitude in how this translation can be done, it is evident that argument trees can provide a formalized representation of the issues raised in the article, and of the conflicts within the information presented.

One further point we want to raise regarding these examples, which we will cover in detail in subsequent chapters, is that we do not automatically subscribe to the view that the structure of the tree is sufficient to judge the validity of the argument at the root. For example, in the above argument tree, it may be tempting to consider that the undercut to the root is defeated, and that therefore the root is undefeated (and therefore in a sense warranted). While we acknowledge this could be a reasonable interpretation in general, we want to consider other factors involving the perspective of the audience (i.e., extrinsic factors), such as the impact of, or the empathy for, the arguments involved, that may lead to alternative interpretations of the validity of the root.

4.3 Polemical Argumentation

We now shift from factual argumentation to polemical argumentation. By polemical argumentation, we mean positional, persuasional, or provocational argumentation. Thus, polemical argumentation contrasts with factual argumentation in that the aim of the argumentation is different and in that the type of information used is different (since subjective and hypothetical information is also drawn upon). Perhaps the clearest way of differentiating factual argumentation and polemical argumentation is to note that with factual argumentation the proponent is neutral with regard to the outcome of the argumentation, whereas in polemical argumentation the proponent is nonneutral. In other words, the proponent is not just laying out the available objective information as an "honest broker."

So far in this chapter, we have highlighted the use of selectivity in factual argumentation. This is motivated by the need to provide the appropriate level of details and interest for the audience of the constellation of arguments and counterarguments produced by factual argumentation.

However, once we acknowledge that writers of articles are not using all the available information in a constellation of arguments and counterarguments, then we are opening the opportunity for writers to be selective in a biased way. To take an extreme case, suppose a writer is not neutral, and the writer wants to get the reader to accept the argument at the root of the argument tree (e.g., to judge whether the initiating argument is warranted); then the writer could select just that argument and ignore other arguments. This may be described as being "economical with the truth." Often polemical argumentation results in this tactic.

Though, one needs to be careful with the use of this tactic. If it is used too often or inappropriately, it can be described as being dogmatic or "blind to the issues." Therefore, writers or speakers who do wish to undertake positional or persuasional argumentation will acknowledge the existence of counterarguments, particularly when they have counterarguments to those counterarguments and/or if the speaker or writer knows that the audience does not care about or does not believe those counterarguments.

Positional argumentation is like factual argumentation, except there is a tendency to select assumptions that favor the presenter's viewpoint, and there is a tendency to use subjective information. For example, organic food is now increasingly popular for ordinary shoppers. These ordinary shoppers are not normally strong advocates of green politics, the ecology movement, or sustainable developments, in general. Rather they believe that there is some merit in the arguments for organic food, and if pressed, shoppers who have bought some organic food would construct some arguments and counterarguments around the ideas that maybe it is more healthy and perhaps it does less damage to the environment, in order to justify their decision to buy the organic food. This would be positional in that they would use their subjective beliefs together with some objective information to justify their position. It is unlikely that they would seek to persuade anyone else to buy organic food; they just want to present and justify their position, perhaps drawing on the same kind of knowledge about the audience, in terms of what they believe and what they regard as important and relevant, as is used for factual argumentation. Thus, in moving from factual argumentation to positional argumentation, the proponent is no longer "an honest broker" because he or she is bringing his or her own subjective beliefs into play.

Now contrast shoppers of organic food with salespeople from an organic food company. They would want to do more than present a position; they would want to persuade members of the public (i.e., potential customers) that organic food is so good for them and so good for the environment that they should buy it. They would aim to present arguments, and counterarguments, for their audience not only to accept the overall claim of their argumentation but also to go out and buy their products. Here, the presenter of the persuasional argumentation would rely more on detailed knowledge about the audience than is often used for factual argumentation or positional argumentation. The presenter would have some goal for the persuasion—whether it is to agree to something, to vote for something, or to buy something. In addition, the presenter would want information about what the audience currently believes, desires, and regards as relevant in order to undertake this argumentation.

Example 4.3.1 Consider a car salesperson who wants to sell a particular model of car with a diesel engine (perhaps because there is a good bonus for the member of the sales team who sells it) to a particular customer. Thus, the objective is clear: The salesperson wants to persuade the customer to buy the car. However, before the salesperson starts a sales pitch, he will find out more about the customer in order to gauge her requirements, and what she thinks is relevant, and what she believes in general about cars. The information gleaned by the salesperson might be the following:

- The customer has a husband and two children.
- The customer needs a car that is relatively economical to run.
- The customer believes that gasoline engine cars have better performance than diesel.

Thus, the salesperson may make a sales pitch by presenting the following arguments, using this gleaned information about the customer:

> This car with a diesel engine is a high-performance family car that both men and women find easy to drive but nonetheless is equipped with four comfortable seats for long journeys and has plenty of room for baggage or shopping. While, in general, diesel engines have inferior performance compared with gasoline engines, with these new engines, the difference in performance between gasoline and diesel is negligible. In any case, diesel engines in fact have good performance, taking into account that they use less fuel and the fuel is cheaper.

The main argument for persuading the customer to buy the car is given in the first sentence. The first clause in the second sentence is an undercut to the main argument. It is given because the salesperson knows it is an issue for the customer, and the salesperson is really only repeating what the customer has said anyway. It will help the salesperson build trust with the customer. And, of course, it provides the structure to the argumentation, since the attack in the first half of the second sentence is itself attacked by the second half of the second sentence and by the attack given in the third and fourth sentences.

We can represent the information in the sales pitch by the following abstract arguments.

(a_1) The car with a diesel engine is a high-performance family car that both men and women find easy to drive but nonetheless is equipped with four comfortable seats for long journeys and has plenty of room for baggage or shopping.

(a_2) In general, diesel engines have inferior performance compared with gasoline engines.

(a_3) With these new engines, the difference in performance between gasoline and diesel is negligible.

(a_4) Diesel engines in fact have good performance, taking into account that they use less fuel and the fuel is cheaper.

The arguments can be represented in the following abstract argument graph with the main argument a_1 at the root:

$$\begin{tikzcd} & a_1 & \\ & a_2 \arrow[u] & \\ a_3 \arrow[ur] & & a_4 \arrow[ul] \end{tikzcd}$$

Thus, we see that the salesperson has made a pitch with one argument that is attacked by a second argument, and this second argument is in turn attacked by two further arguments.

Now we turn to provocational argumentation. The key difference between positional and provocational argumentation is that in provocational argumentation, the audience (e.g., a reader) does not believe that the presenter (e.g., an article writer) believes all the assumptions, whereas in positional argumentation, the audience does tend to believe that the presenter believes the assumptions.

The key difference between persuasional and provocational argumentation is as follows: In persuasional argumentation, the presenter tries to maximize the empathy that the audience might have for the arguments and to maximize the antipathy the audience might have for counterarguments, whereas in provocational argumentation, the presenter is in a sense less subtle, since the presenter does not worry so much about optimizing the believability of the arguments (i.e., maximizing the empathy that the audience might have for the arguments and maximizing the antipathy the audience might have for counterarguments), and is in a sense more subtle in the way that he or she invokes a mild degree of indignation in the audience as part of the process of getting some point across to the audience. Furthermore, this indignation is not meant to be crude anger or irritation, but rather upsetting the balance of beliefs in the audience in a way that the audience does not find easy to handle—so that while the audience of provocational argumentation might be inclined to disagree with key assumptions and the overall claim of the argumentation, they find it difficult to reject the presentation outright, and they may indeed admire the form of the argumentation. This later phenomenon is often seen in newspaper opinion articles where the journalist writes a provocational argument in part for generating interest and in part as a form of entertainment.

4.4 Speculational Argumentation

Speculational argumentation is argumentation with hypothetical information for some constructive problem analysis. This is information that could be described as speculative. At one extreme, it could be information that is reasonable to believe, but at the present time it cannot be confirmed to be true. At the other extreme, it could be information that is quite unreasonable to believe but that would be useful or interesting to use as assumptions for argumentation in order to consider some of the consequences of the assumptions.

It is useful to undertake speculational argumentation for many tasks. Often it is useful when we need to reason about some problem and we lack adequate factual information, so we add speculative information as assumptions. Doing this may help us to look at possibilities and thereby help us search for factual information. We see this in many situations. Consider, for example, a clinician speculating about the possible disorders that could cause the symptoms in a particular patient or a detective speculating about who committed a particular crime that is under investigation.

Let us consider the example of a detective investigating a murder. The detective has to speculate on the circumstances leading up to the murder of the victim, on who would have a motive for murdering the victim, and on who would have the facilities for committing the murder. The detective may consider some suspects, where they were at the time of the murder, their access to the murder weapon, their motives, and so forth, and then construct arguments for and against each suspect being the murderer. The speculation may include a construction of the scenario leading up to the murder and after the murder. If an interesting argument could be constructed using some speculative information for the suspect being the murderer, then the detective may use this as a driver for seeking evidence to back up the argument and for defeating any counterarguments. If appropriate evidence is found, then speculative information changes into factual information.

In some situations, abduction may be required as part of the process of identifying the required hypothetical information. Given a knowledgebase Γ and a formula ϕ such that $\Gamma \nvdash \phi$, abduction is the process by which a set of assumptions Θ is found for which $\Gamma \cup \Theta \vdash \phi$, and $\Gamma \cup \Theta \nvdash \perp$, and there is no $\Theta' \subset \Theta$ such that $\Gamma \cup \Theta' \vdash \phi$. Often there is some specific knowledgebase from which Θ is obtained.

To illustrate the use of abduction in speculational argumentation, consider an example of a detective speculating about who may have committed a murder of a wealthy person. Here, in order to construct an argument that a particular suspect committed the murder, the detective may abduce that the suspect has a particular motive—say, an urgent need for money—and if this abduction (i.e., the hypothetical information that the suspect has an urgent need for money) proves to be useful in the argumentation, then the detective may focus some resources on determining whether or not the suspect did indeed have this motive. If it turns out the suspect did have an urgent need for money, then the hypothetical information changes status to objective information.

The way speculation works in police work is reflected in many other fields such as medical practice, machine maintenance, and software maintenance, where identifying cause and effect (i.e., diagnosis) is a key step in any of the work. In a sense, there is an overlap here between the way speculative information is used as a driver for finding evidence and for the way abduction is used in diagnosis. There is a strong sense of looking at the past (i.e., what has already happened) in detective work and in diagnosis, and then trying to fill in the gaps in the information required to fully explain what has happened. In deductive work and diagnosis, the emphasis is on finding the most likely explanation (given our current understanding of the situation), but in other kinds of speculation, there is a tendency to look more widely to consider also less likely explanations but those with more import for some reason.

Another kind of speculation involves considering potential problems that might arise, and here there is a strong sense of looking at the future (i.e., considering what could happen). These may include considering situations that will (or are very likely to) eventually happen in some form such as the following:

- The world is prepared for when the oil runs out.
- Coastal cities are prepared for when the sea level will rise 1 meter.
- I have saved enough money securely for my retirement.

Speculation about the future also includes situations that are highly unlikely, or even virtually impossible, but that are interesting or informative to consider, or that we may still wish to prepare for even if they are highly unlikely.

- The world is prepared for when a UFO lands.
- We will have enough air-conditioning units by 2025 for a rise of 15 °C in ambient temperature.
- I have decided how I wish to spend my $50 million lottery win.

Using speculational argumentation to consider possibilities in the future is important at a personal level, at a government level, and at a corporate level. Some of it is risk management and sound planning for possible dangers or problems—for example, civil defense planning (such as checking that there are adequate equipment and personnel to deal with flooding, earthquakes, and airline accidents), military defense planning, and corporate disaster planning (such as checking that there are adequate procedures for dealing with a fire occurring in a computer center or factory of the company). And some of this speculational argumentation is for seeking opportunities. Government agencies and corporations are constantly undertaking exercises such as SWOT (Strengths, Weaknesses, Opportunities, and Threats) analyses that can involve a considerable amount of speculational argumentation. Opportunities that can be identified by speculational argumentation can lead to further research and development, eventually leading to new products and services. Similarly, threats can be identified by speculational argumentation, and they can lead to strategies and tactics to defend or offset those dangers. Again, the speculational argumentation can be important in exploring possibilities, and speculative information can be turned into factual information on further investigation.

Speculational argumentation is also important as an intellectual exercise and for scholarly research. As an illustration, in the following example, speculative information about the existence of time machines is used to explore some of the conceptual problems that could arise.

Example 4.4.1 The following are quotes from an article by Paul Davies in Scientific American (August 13, 2002) entitled "How to Build a Time Machine: It Wouldn't Be Easy, but It Might Be Possible":

> Assuming that the engineering problems could be overcome, the production of a time machine could open up a Pandora's box of causal paradoxes. Consider, for example, the time traveller who visits the past and murders his mother when she was a young girl. How do we make sense of this? If the girl dies, she cannot become the time traveller's mother. But if the time traveller was never born, he could not go back and murder his mother.
>
> Paradoxes of this kind arise when the time traveller tries to change the past, which is obviously impossible. But that does not prevent someone from being a part of the past. Suppose the time traveller goes back and rescues a young girl from murder, and this girl grows up to become his mother. The causal loop is now self-consistent and no longer paradoxical. Causal consistency might impose restrictions on what a time traveller is able to do, but it does not rule out time travel per se.

We can represent some of the arguments in the first paragraph with the following formulae where we consider a particular hypothetical individual called John: t_1 is the hypothetical information that John was born in 1970; t_2 is the fact that 1970 is over ten years after 1948; t_3 is the hypothetical information that John murdered John's mother in 1948; t_4 is the hypothetical information that if X murders Y in T_0 and X was born over ten years later in T_1, then X must have travelled back in time to T_0; t_5 is the categorical information that if X murders Y at time T_0, then Y becomes dead at T_0; t_6 is the categorical information that if Y is dead at T_0, then at any time T_1 that is over ten years later, it is not the case that Y gives birth to a child Z at T_1; and t_7 is the categorical information that if it is not the case that the mother of X gives birth to X at T_1, then it is not the case that X is born at T_1.

t_1 $\texttt{Born(John, 1970)}$

t_2 $\texttt{OverTenYearsLater(1948, 1970)}$

t_3 $\texttt{Murder(John, Mother(John), 1948)}$

t_4 $\forall \texttt{X,Y,T}_0\texttt{,T}_1\ \texttt{Murder(X,Y,T}_0\texttt{)} \wedge \texttt{Born(X,T}_1\texttt{)} \wedge \texttt{OverTenYearsLater(T}_0\texttt{,T}_1\texttt{)} \rightarrow \texttt{TimeTravel(X,T}_0\texttt{)}$

t_5 $\forall \texttt{X,Y,T}_0\ \texttt{Murder(X,Y,T}_0\texttt{)} \rightarrow \texttt{Dead(Y,T}_0\texttt{)}$

t_6 $\forall \texttt{Y,T}_0\texttt{,T}_1\texttt{,Z}\ \texttt{Dead(Y,T}_0\texttt{)} \wedge \texttt{OverTenYearsLater(T}_0\texttt{,T}_1\texttt{)} \rightarrow \neg\texttt{GiveBirth(Y,Z,T}_1\texttt{)}$

t_7 $\neg\texttt{GiveBirth(Mother(X),X,T}_1\texttt{)} \rightarrow \neg\texttt{Born(X,T}_1\texttt{)}$

Using these formulae, we can construct the following argument tree with the subject being that our hypothetical character John undertake time travel to take him back to 1948. Here we use the labels for the formulae (rather than the formulae themselves) in the support for each argument:

$$\langle\{t_1, t_2, t_3, t_4\}, \texttt{TimeTravel(John, 1948)}\rangle$$

$$\uparrow$$

$$\langle\{t_2, t_3, t_5, t_6, t_7\}, \neg\texttt{Born(John, 1970)}\rangle$$

Thus, using simple categorical knowledge and a few hypothetical pieces of information, together with the speculative information t_4, we can construct an argument for the existence of travel, and we can undercut that argument with the undercut that if he did do the time travel, he couldn't have been born to undertake the time travel.

From this example, we see that we can reflect speculational argumentation in terms of argument trees if we can identify the required hypothetical information, together with appropriate objective and subjective information. As with other kinds of argumentation, such as factual and polemical argumentation, we may need to take the audience into account to improve the believability and impact of the argumentation.

4.5 Coalitions of Arguments Working in Harmony

Obviously, argumentation involves conflicts. If we had no conflicts in our knowledge, then we would have no need for monological argumentation. Formalization of argumentation captures conflicts in terms of arguments and counterarguments. In abstract argumentation, we saw that arguments are atomic and conflicts are represented by an "attacks" relation. In logical argumentation, we see conflicts in knowledge in terms of the existence of conflicts between sets of formulae. This, in turn, draws on the logical notion of inconsistency based on the inference of a formula and its negation. Thus, conflict in knowledge has a clear formalization in both abstract argumentation and logical argumentation. However, formalizations of argumentation must do more than just acknowledge the existence of conflicts. They have to flag where they take place, and they have to consider the types of conflicts.

To address these requirements, we start by considering how arguments work together. Arguments are not used in isolation in practical argumentation. Normally there is some connection between an argument and the other arguments being presented (excluding at this point the consideration of a non sequitur). Furthermore, if we think in terms of working out whether a particular formula, or argument, is made valid (i.e., warranted) or invalid (i.e., unwarranted) in a constellation of arguments and counterarguments, then we see there are coalitions of arguments that work together to bring about the decision on validity. In an argument tree T, we can think of $\mathsf{Defenders}(T)$ as a coalition, and we can think of $\mathsf{Attackers}(T)$ as a coalition. Once we have identified coalitions of arguments, then we can consider whether they work together in harmony (i.e., without conflict), or whether there is some form of disharmony (i.e., conflict) in their coalition. The following is an extreme example of how there is a complete lack of harmony in the coalitions.

Example 4.5.1 Let $\Delta = \{\alpha \wedge \beta, \neg\alpha \wedge \beta, \alpha \wedge \neg\beta\}$. Consider the following argument tree. Here the same support is used as for both attacking and defending arguments (e.g., $\{\neg\alpha \wedge \beta\}$ is the support for an attacking argument and for a defending argument).

$$\begin{array}{ccc} & \langle\{\alpha\wedge\beta\},\alpha\wedge\beta\rangle & \\ \nearrow & & \nwarrow \\ \langle\{\neg\alpha\wedge\beta\},\Diamond\rangle & & \langle\{\alpha\wedge\neg\beta\},\Diamond\rangle \\ \uparrow & & \uparrow \\ \langle\{\alpha\wedge\neg\beta\},\Diamond\rangle & & \langle\{\neg\alpha\wedge\beta\},\Diamond\rangle \end{array}$$

Example 4.5.2 Consider the following three viewpoints (a_1)–(a_3) (as summarized by Pierre Marquis) that were presented by different proponents at the time of the 2005 French referendum on the European Constitution. The European Constitution was intended to help the member countries of the European Union make decisions more efficiently. This is reflected by argument (a_1), with arguments (a_2) and (a_3) each attcking it. Here there is a lack of harmony between arguments (a_2) and (a_3).

(a_1) The European Constitution is good for Europeans; therefore, Europeans should vote for it.

(a_2) The European Constitution is too liberal with respect to workers' rights; therefore, it is not good for Europeans.

(a_3) The European Constitution is not sufficiently liberal with respect to workers' rights; therefore, it is not good for Europeans.

We can represent this information using formulae based on the following propositions:

p The European Constitution is a good for Europeans.

q Europeans should vote for the European Constitution.

r The European Constitution is too liberal w.r.t. workers' rights.

s The European Constitution is not sufficiently liberal w.r.t. workers' rights.

Letting $\Delta = \{p, r, s, p \rightarrow q, r \rightarrow \neg p, s \rightarrow \neg p, r \leftrightarrow \neg s\}$, we can view these informal arguments in the following argument tree:

$$\begin{array}{ccc} & \langle\{p, p\rightarrow q\}, q\rangle & \\ \nearrow & & \nwarrow \\ \langle\{r, r\rightarrow\neg p\},\Diamond\rangle & & \langle\{s, s\rightarrow\neg p\},\Diamond\rangle \\ \uparrow & & \uparrow \\ \langle\{s, r\leftrightarrow\neg s\},\Diamond\rangle & & \langle\{r, r\leftrightarrow\neg s\},\Diamond\rangle \end{array}$$

Here, we see that the attackers (i.e., the children to the root) are mutually inconsistent given the integrity constraint $r \leftrightarrow \neg s$ in Δ. Hence there is a lack of harmony in the arguments being presented.

We now consider a pair of examples with identical structure but for which, intuitively, the degree of harmony is higher in the second example than the first.

Example 4.5.3 This example is adapted from an example by Prakken [Pra02]. Consider the following propositions:

p The suspect murdered the victim.

q The suspect is assumed innocent.

r Bob says that the suspect shot the victim.

s John says that the suspect stabbed the victim.

Letting $\Delta = \{p, q, r, s, q \to \neg p, r \to \neg q, s \to \neg q, r \leftrightarrow \neg s\}$, the argument tree is as follows:

$$
\begin{array}{ccc}
 & \langle\{p\}, p\rangle & \\
 & \uparrow & \\
 & \langle\{q, q \to \neg p\}, \Diamond\rangle & \\
\nearrow & & \nwarrow \\
\langle\{r, r \to \neg q\}, \Diamond\rangle & & \langle\{s, s \to \neg q\}, \Diamond\rangle \\
\uparrow & & \uparrow \\
\langle\{s, r \leftrightarrow \neg s\}, \Diamond\rangle & & \langle\{r, r \leftrightarrow \neg s\}, \Diamond\rangle
\end{array}
$$

Thus, the conclusion we could draw is that it is not the case that "The suspect murdered the victim." This may be a reasonable inference to draw if we think that the conflict between r and s invalidates the evidence r and s (i.e., the evidence given by Bob and John). In any case, there is significant disharmony in the support of the defenders and in the support of the attackers in the argument tree.

Now consider the following example, which has exactly the same structure as above but for which we may want to draw a different inference.

Example 4.5.4 Consider the following propositions:

p The suspect murdered the victim.

q The suspect is assumed innocent.

r Bob says that the suspect shot the victim with three bullets.

s John says that the suspect shot the victim with four bullets.

Again, the argument tree is as follows:

$$
\begin{array}{ccc}
 & \langle\{p\}, p\rangle & \\
 & \uparrow & \\
 & \langle\{q, q \to \neg p\}, \diamond\rangle & \\
 \nearrow & & \nwarrow \\
\langle\{r, r \to \neg q\}, \diamond\rangle & & \langle\{s, s \to \neg q\}, \diamond\rangle \\
\uparrow & & \uparrow \\
\langle\{s, r \leftrightarrow \neg s\}, \diamond\rangle & & \langle\{r, r \leftrightarrow \neg s\}, \diamond\rangle
\end{array}
$$

Here we still have r and s used in arguments that defeat each other. However, in this new example, the inconsistency between r and s seems much smaller. One could easily believe that the inconsistency should be ignored, and so the arguments involving them should not defeat each other.

The conclusions we want to draw from these examples is that we want to evaluate the significance of the conflicts arising between arguments and, in some cases, ignore conflicts. In chapter 9, we will discuss how recent developments in measuring degree and significance of inconsistency can be used to analyze harmony of arguments.

4.6 Making the Same Point in Different Ways

When presenting arguments in a real-world situation, the audience has mental faculties for reasoning with the information given. If we underestimate this, we can render the presentation repetitive and uninteresting. Imagine you ask a car salesman why you should buy a new BMW 320, and he says *the first reason is that it has airbags and ABS and the second reason is that it has ABS and airbags*. You would regard the two reasons as being equivalent, and the answer given by the car salesman as being flawed. We consider this further in the following example.

Example 4.6.1 We return to the text presented in example 4.1.1 and recall the atoms p_1 to p_4 as follows:

- p_1 Universities cannot continue as finishing schools using taxpayers' money.
- p_2 Students need to pay towards their university education.
- p_3 It is ok to charge a students 3,000 pounds per year for tuition fees.
- p_4 The government proposals are good.

Given these atoms, we can form two arguments as follows. The first was presented in example 4.1.1, and the second involves some new assumptions involving the atoms p_1 to p_4:

$$\langle \{p_1, p_1 \to p_2, p_2 \to p_3, p_3 \to p_4\}, p_4 \rangle$$

$$\langle \{p_1, p_2, p_3, p_1 \wedge p_2 \wedge p_3 \to p_4\}, p_4 \rangle$$

The supports of the two arguments are equivalent. In other words, the following holds:

$$\begin{aligned} &\mathsf{Cn}(\{p_1, p_1 \to p_2, p_2 \to p_3, p_3 \to p_4\}) \\ &\quad = \mathsf{Cn}(\{p_1, p_2, p_3, p_1 \wedge p_2 \wedge p_3 \to p_4\}) \end{aligned}$$

Presenting both arguments together could be regarded as introducing redundancy. If the arguments were presented in natural language, it is likely that most readers would realize they were equivalent.

For auto-argumentation, the redundant arguments obscure the picture for the user. For one-to-many argumentation, the redundancy reduces the quality for the audience and may undermine the confidence the audience has in the presenter of the arguments.

The nature of redundancy can take various forms, and we need to take a number of issues into account if we want to address the problem. We illustrate some of these issues with the following examples.

Example 4.6.2 Consider the following argument tree, where the left undercut seems to be redundant:

$$\begin{tikzcd} & \langle \{\alpha, \beta, \gamma\}, \alpha \wedge \beta \wedge \gamma \rangle & \\ \langle \{\neg\alpha \vee \neg\beta\}, \diamond \rangle \arrow[ur] & & \langle \{\neg\alpha \wedge \neg\beta\}, \diamond \rangle \arrow[ul] \end{tikzcd}$$

Here the support of the right undercut subsumes, in the sense it implies, the support of the left argument.

Example 4.6.3 Consider the following argument tree, where the left undercut seems to be redundant. Again the right undercut subsumes (i.e., implies) the left undercut. In a sense, it is more informative. It indicates why we have $\neg\delta$:

$$\begin{tikzcd} & \langle \{\beta, \beta \to \gamma \wedge \delta\}, \delta \rangle & \\ \langle \{\neg\delta\}, \diamond \rangle \arrow[ur] & & \langle \{\alpha, \alpha \to \beta \wedge \neg\delta\}, \diamond \rangle \arrow[ul] \end{tikzcd}$$

However, with this example, we may take an alternative view: After the principle of Occam's razor, we may regard the right undercut as redundant because the smaller and simpler undercut on the left is adequate for the task of attacking the argument at the root.

Example 4.6.4 Consider the following argument tree, where the left undercut is subsumed by the right undercut but the right undercut is a more concise support than the left undercut (Occam's razor again):

$$\begin{array}{ccc} & \langle\{\gamma, \gamma \to \delta\}, \delta\rangle & \\ \nearrow & & \nwarrow \\ \langle\{\alpha, \alpha \to \beta, \beta \to \neg\gamma\}, \Diamond\rangle & & \langle\{\beta, \beta \to \neg\gamma\}, \Diamond\rangle \end{array}$$

Example 4.6.5 Consider the following argument trees, where T_1 has been constructed from Δ_1 and T_2 has been constructed from Δ_2. There is a strong sense that T_1 and T_2 convey the same information, but T_2 does so more efficiently (in the sense that it provides a more concise reason):

T_1

$$\begin{array}{ccc} & \langle\{\alpha, \alpha \to \beta\}, \beta\rangle & \\ \nearrow & & \nwarrow \\ \langle\{\gamma, \gamma \to \neg\beta\}, \Diamond\rangle & & \langle\{\delta, \delta \to \neg\beta\}, \Diamond\rangle \end{array}$$

T_2

$$\begin{array}{c} \langle\{\alpha, \alpha \to \beta\}, \beta\rangle \\ \uparrow \\ \langle\{\gamma \wedge \delta, \gamma \vee \delta \to \neg\beta\}, \Diamond\rangle \end{array}$$

This example illustrates how we can compare argument trees. More interestingly, it indicates how we can take an argument tree like T_1 and edit Δ_1 in a limited way to obtain Δ_2 (note that $\{\gamma, \gamma \to \neg\beta, \delta, \delta \to \neg\beta\}$ and $\{\gamma \wedge \delta, \gamma \vee \delta \to \neg\beta\}$ are logically equivalent) and thereby obtain T_2.

Example 4.6.6 Consider the following argument trees, where $\Delta = \{\alpha \wedge \beta, \alpha \wedge \gamma, \neg\alpha\}$:

$$\begin{array}{ccc} & \langle\{\neg\alpha\}, \neg\alpha\rangle & \\ \nearrow & & \nwarrow \\ \langle\{\alpha \wedge \beta\}, \Diamond\rangle & & \langle\{\alpha \wedge \gamma\}, \Diamond\rangle \end{array} \qquad \begin{array}{c} \langle\{\alpha \wedge \beta\}, \alpha\rangle \\ \uparrow \\ \langle\{\neg\alpha\}, \Diamond\rangle \\ \uparrow \\ \langle\{\alpha \wedge \gamma\}, \Diamond\rangle \end{array} \qquad \begin{array}{c} \langle\{\alpha \wedge \gamma\}, \alpha\rangle \\ \uparrow \\ \langle\{\neg\alpha\}, \Diamond\rangle \\ \uparrow \\ \langle\{\alpha \wedge \beta\}, \Diamond\rangle \end{array}$$

In a sense, these three trees are equivalent. They all involve three arguments. There is a bijection, based on logical equivalence of supports, between these arguments. Thus, we could contend that there is some redundancy here.

In this section, we have highlighted the need to consider how similar arguments can make the same point in different ways. This may result in redundancy in an individual argument tree, and it raises the possibility of editing a knowledgebase so as to reduce redundancy. However, in doing this, we may also need to take into account the trade-off between the informativeness of a support versus the conciseness of a support. These issues have not been adequately addressed by the abstract or logical techniques presented so far in this book, and yet they are clearly important features of practical argumentation undertaken by real intelligent agents. To start to address these needs in formalizing practical argumentation, we will pick up on some of these issues in chapter 5.

4.7 Analysis of Intrinsic and Extrinsic Factors

Given an argument tree, there are various aspects of it that we can evaluate as intelligent agents. These evaluations help us to understand and assess arguments and counterarguments produced by other agents, and they help to refine our own arguments and counterarguments. For formalizing practical argumentation, we need to conceptualize these kinds of evaluation. We start with introducing intrinsic factors and then move on to introducing extrinsic factors.

Analysis of **intrinsic factors** involves evaluating the logical features of individual arguments and relationships between two or more arguments. Some examples of intrinsic factors for which we will consider analytical techniques include the following:

Degree of undercut Some undercuts appear to negate their parent more than others. For example, consider the argument $\langle\{\alpha \wedge \beta\}, \alpha \wedge \beta\rangle$ and the undercuts $\langle\{\neg\alpha\}, \Diamond\rangle$ and $\langle\{\neg\alpha \wedge \neg\beta\}, \Diamond\rangle$. The second undercut intuitively appears to negate the parent more than the first undercut. We formalize this measure of undercut in chapter 5.

Degree of harmony We introduced the idea that we may measure harmony between a set of arguments that are in a coalition (in section 4.5). In chapter 9, we will discuss how recent developments in measuring degree and significance of inconsistency can be used to analyze harmony of arguments.

Thus, analysis of intrinsic factors is based on the information available in the constellation of arguments and counterarguments presented. In contrast, analysis of **extrinsic factors** is based on extending this information with information about the intended audience so that argumentation can be tailored for that audience. Some examples of extrinsic factors for which we will consider analytical techniques include the following:

Degree of resonance Given a set of concerns (a set of formulae representing what a stereotypical member of the audience thinks is important), the echo of an argument is the subset of the concerns that are either implied or negated by the support of the argument. The degree of resonance of an argument is defined in terms of the weighted sum of the echo of the argument. We motivate and define the degree of resonance in our coverage of higher impact argumentation in section 6.1.

Degree of empathy Assuming we have a set of formulae that represent the beliefs of a stereotypical member of the intended audience, the degree of empathy that the stereotypical member has for a particular argument is measured in terms of how much the beliefs of the stereotypical member entail the support of the argument. This can, for example, be evaluated in terms of the proportion of models of the beliefs that are also models of the support. This measure is motivated and defined in section 6.2 for making arguments more believable.

Degree of antipathy Assuming we have a set of formulae that represent the beliefs of a stereotypical member of the intended audience, the degree of antipathy that the stereotypical member has for a particular argument is measured in terms of how much the beliefs of the stereotypical member conflict with the support of the argument. This can, for example, be evaluated in terms of the Dalal distance (i.e., the Hamming distance) between the models of the beliefs and the models of the support. This measure is also motivated and defined in section 6.2 as part of our coverage of making argumentation more believable.

Cost of understanding For an audience to accept an argument may involve some cognitive cost. There are various ways this could be measured depending on the nature of the arguments and some understanding of what the audience may regard as more difficult to understand. An intuitive example is that an argument that has a support with more formulae costs more, or an argument with more formulae composed from more propositional symbols costs more, or an argument with more complex formulae (reflected by the number of logical symbols and the nesting of the logical symbols) costs more. We will provide a general framework for cost in our coverage of higher impact argumentation in section 6.3.

For undertaking analysis of extrinsic factors, we assume information (such as particular kinds of information about the audience) in addition to the knowledgebase that we use to construct the arguments and counterarguments. This extra information is then used to evaluate individual arguments, argument trees, and/or argument structures.

A constellation of arguments and counterarguments presented to a user can be annotated with one or more of these measures as required by the user. Furthermore, as we will argue in the next section, we can use these measures to tailor a presentation of arguments and counterarguments for a particular audience. Moreover, in the following chapters, we will be demonstrating how the analysis of intrinsic and extrinsic factors in argumentation will give us invaluable information for developing more intelligent practical argumentation.

4.8 Selective and Formative Argumentation

In virtually any real-world situation where monological argumentation is being undertaken, there is some selectivity of arguments presented. Not all arguments that can be constructed from the assumptions are presented in the constellation. We have seen this in the examples of articles considered in this chapter (such as example 4.1.1), where we have constructed a set of assumptions corresponding to an article and then shown that only a subset of the arguments that the proponent could have used have indeed been used.

Thus, for formalizing argumentation, we need to develop **selective argumentation**. If we are to automatically generate constellations of arguments and counterarguments, then we need ways to assess which are, in some sense, the best arguments to include. The role of selective argumentation is to make these assessments as part of the process of monological argumentation.

A first step towards selective argumentation is the use of canonical arguments. In chapter 3, we argued that by restricting consideration to canonical arguments, we do not lose any useful information. Thus, an argument structure for a claim α consisting of all the argument trees for α, where each is composed of all the canonical undercuts, is an exhaustive representation of the constellation of arguments and counterarguments for α. However, it is clear that we need to do much more than this to be able to capture selective argumentation for practical use.

Furthermore, we need to go beyond selective argumentation. We need also to develop **formative argumentation**. By this, we mean that we reform (i.e., logically manipulate) the arguments in a constellation of arguments and counterarguments. Since these are formal entities, we can define principled means for manipulating them. In our framework, we can consider replacing arguments in an argument tree with logically equivalent arguments, perhaps in a simpler syntactical form, or we can merge siblings in an argument tree, resulting in a tree that is in some sense equivalent but with fewer nodes. And once we start to be formative, by manipulating our argument trees, and argument structures, we can consider going back to the knowledgebase and manipulating the formulae there and then construct arguments afresh from the new knowledgebase as suggested in section 4.6.

Thus, for practical argumentation we require both selective argumentation and formative argumentation. In turn, to support these activities, we need to be able to assess our arguments as they arise in constellations. For this, we advocate the use of the results from analyzing both intrinsic and extrinsic factors. These will provide means for formalizing techniques, called **rationalization techniques**, for selective argumentation and formative argumentation. We will consider rationalization techniques based on intrinsic factors in chapter 5 and on extrinsic factors in chapter 6.

As we unfold our presentation of practical argumentation, we will see a number of interesting new ideas for formalizing argumentation. However, describing practical argumentation as being logical argumentation based on selective argumentation and formative argumentation is only a very partial characterization of a very complex cognitive process, but it does open numerous directions for further development.

4.9 Reflecting Natural Language in Logic

This chapter has been about practical argumentation as it occurs in the real world. From our discussions, it is clear that we need to consider how arguments are presented in natural language and how those arguments can be reflected in logic.

There is an enormous literature on presenting natural language text in formal logic. Indeed, attempting to understand the nature of natural language has been a great driver for the development of formal logic. These developments include mathematical logic for avoiding the ambiguities, imprecision, and lack of clarity, of natural language for presenting definitions and theorems (see, e.g., [Ham78, van80]); philosophical logics, such as tense logics and modal logics, for understanding diverse philosophical issues surrounding natural language such as paradoxes, ontologies for time, generalized quantification, and modalities such as possibility and necessity (see, e.g., [Haa78, GHR95, GHR98]); logics for formalizing syntax and semantics of natural languages (see, e.g., [Can93, KR93]); and logics for formalizing parsing of natural language (see, e.g., [GM89, KMG01]).

Furthermore, there is a large literature on analyzing information in natural language using logic. Many introductory textbooks on logic, particularly for philosophy, consider examples in natural language text, followed by a presentation of these examples using classical propositional and predicate logic (see, e.g., [Fis88, Hod77]). However, it is interesting to note that the textbooks, such as [Fis88], are focused on getting a set of formulae out of a text where there is no inconsistency. In this chapter, we have shown how conflicts do naturally arise in short pieces of text such as published in newspaper and magazine articles and that these conflicts are intended to be seen in the information being presented by the writer. The inconsistencies are not errors but reflections of knowledge about the world.

While the understanding of the underlying principles of natural language are complex and incomplete, it is clear that representing information from natural language in a logical language is a valuable and viable process, even though there is considerable latitude on the precise encoding used in the translation process. From the current state of the art in computational linguistics, the automation of the process is difficult to scale up beyond restricted vocabularies and grammars.

In the examples that we have considered in this chapter of translating free text into logic and thereby into arguments, we see that a simple structure occurs in much of the information: Often each formula is an implication with both the antecedent and consequent being literals. However, in general, it is clear that a much more complex structure of formulae is required. Turning to the textbooks of logic, such as [Fis88], we see plenty of examples where more complex use of the logical connectives is required to capture pieces of information from text.

As a further illustration of how an individual argument in textual form can be translated into a set of logical formulae involving a range of connectives, we examine St. Anselm's ontological argument.

Example 4.9.1 Simply put, St. Anselm's ontological argument is as follows:

God is a being for which no greater being can be imagined. However, that for which a greater cannot be imagined cannot exist in the mind only. For if it is actually in the mind only, it can be imagined as existing also in reality, which is greater. Therefore, there exists a being for which a greater cannot be imagined, both in the mind and in reality: God exists.

Informally, the idea runs from the following premises:

(I) God is a being for which no greater being can be imagined.

(II) God exists in the mind (roughly speaking, the concept of God is captured by human understanding).

(III) If God exists in the mind but not in reality (i.e., in the mind only), then God existing can be imagined, and that is a greater being than God (*inasmuch as existing is viewed as greater than nonexisting*).

(IV) If a greater being than God can be imagined, then God is not a being than which no greater being can be imagined (trivially).

Premises (I) and (IV) imply that a greater being than God cannot be imagined. But it could be, if both parts in the antecedent of (III) were true; then we would have the following:

(i) God exists in the mind.

(ii) God does not exist in reality.

Since (i) is a premise (it is (II)—no more, no less), (ii) must be false: hence the conclusion that God exists in reality.

A formalization in terms of classical propositional logic is possible as follows:

p God is a being for which no greater being can be imagined.

q God exists in the mind.

r God exists in reality.

s A greater being than God can be imagined.

To make the comments below more readable, let "a being for which a greater cannot be imagined" be denoted as "a *G*-being":

1. p	God is a *G*-being.
2. q	God exists in the mind.
3. $q \wedge \neg r \rightarrow s$	If God exists in the mind only, then a greater being than God can be imagined.

4. $s \rightarrow \neg p$ — If a greater being than God can be imagined, then God is not a G-being.

5. $\neg s$ — Due to 1 and 4.

6. r — Due to 2, 3, and 5.

Hence the following argument can be identified:

$$\langle \{p, q, q \wedge \neg r \rightarrow s, s \rightarrow \neg p\}, r \rangle$$

There have been many criticisms to St.Anselm's argument. The most common criticism rejects premise (I) on the grounds that the concept of a being for which no greater being can be imagined is contradictory. Stated otherwise, the objection is that premise (I) implies a contradiction. Hence the undercut below aimed at premise (I)

$$\langle \{p \rightarrow \perp\}, \diamond \rangle$$

The rejoinder here takes various forms that can be captured by the statement that the concept is not contradictory:

$$\langle \{\neg(p \rightarrow \perp)\}, \diamond \rangle$$

A criticism originated with Aquinas denying that the concept of God can be captured by human understanding, flatly negating the statement conveyed by premise (II). Hence the following undercut aimed at premise (II):

$$\langle \{\neg q\}, \diamond \rangle$$

Again, there are rejoinders. Within the setting of classical propositional logic, they are not of a form worth representing; hence, they are omitted here (observe that by no means does the resulting argument tree indicate that the undercut $\langle \{\neg q\}, \diamond \rangle$ has no undercut itself—it has because q occurs in the support of the initial argument $\langle \{p, q, q \wedge \neg r \rightarrow s, s \rightarrow \neg p\}, r \rangle$).

Among the many criticisms challenging the postulate that existing is greater than nonexisting, there are even arguments to the effect that the contrary is the case, eventually concluding the nonexistence of God (in symbols, $\{p, s \rightarrow \neg p, r \rightarrow s\} \vdash \neg r$). An illustration is provided by the following undercut aimed at premise (III):

$$\langle \{r \rightarrow s\}, \diamond \rangle$$

Putting together the few formal arguments just presented gives the following oversimplified argument tree:

$$
\begin{tikzcd}
& \langle\{p, q, q \wedge \neg r \rightarrow s, s \rightarrow \neg p\}, r\rangle & \\
\langle\{\neg q\}, \Diamond\rangle \arrow[ur] & \langle\{p \rightarrow \bot\}, \Diamond\rangle \arrow[u] & \langle\{r \rightarrow s\}, \Diamond\rangle \arrow[ul] \\
& \langle\{\neg(p \rightarrow \bot)\}, \Diamond\rangle \arrow[u] &
\end{tikzcd}
$$

All criticisms to St. Anselm's argument attack either some of the premises or the inference. It is also the case for all rejoinders to the criticisms. They all belong to a special class of counterarguments that do not attack (at least directly) the consequent of the argument. In fact, most criticisms (do not forget the exception mentioned above) against St. Anselm's argument do not deny God's existence.

While following techniques suggested in basic textbooks, such as by Fisher [Fis88], allows us to extract by hand individual arguments from text and represent them in propositional logic, we also need to extract arguments in the form of first-order logic. There are numerous techniques developed in computational linguistics that offer some possibilities for either semi-automated or automated translation of text into first-order logic (see, e.g., [GM89, KMG01]).

It is also worth noting that in newspaper, magazine, and journal articles, a lot of argumentation involves one or more of the following kinds of knowledge: (1) obligations and permissions (unsurprising given a lot of editorials are about government legislation etc.), (2) possibility and probability, and (3) cause and effect. Since each of these kinds of knowledge has been addressed in the knowledge representation and reasoning literature by specialized logics, it would seem these logics should be reflected in argumentation systems. We return to this point in chapter 9.

4.10 Discussion

In this chapter, we have considered some aspects of the nature of practical argumentation in the real world and have shown how abstract argumentation (as formalized in chapter 2) and logical argumentation (as formalized in chapter 3) can be used to present diverse examples of practical argumentation in a formal way. However, we have discussed a number of shortcomings in these proposals vis-à-vis the lack of taking analysis of intrinsic and extrinsic factors into account when presenting arguments and the lack of techniques for selective and formative argumentation. The basic proposal for abstract and logical argumentation offers formal foundations, but there is a lack of techniques for bringing formal argumentation closer to capturing practical argumentation as seen in the real world. In the following chapters, we address some of these shortcomings.

4.11 Bibliographic Notes

Some of the issues discussed in this chapter have long since been recognized and studied in the informal logics and philosophy literature. This includes differentiating different roles for argumentation such as factual argumentation, different kinds of polemical argumentation, and speculational argumentation. Furthermore, there has been recognition of the need to consider the audience and, in particular, for the proponent to be selective in what is presented to the audience (see, in particular, [Per80, Per82]).

However, formal characterizations of argumentation systems that take the audience into account have been limited. Recently, some interest in these issues has arisen, starting with a proposal by Bench-Capon that extended Dung's abstract argumentation frameworks with a value judgment that is defined in terms of the audience [Ben03], and this has been extended to capture aspects of persuasive argumentation (see, e.g., [ABM05, GS05]). In [Ben03], abstract argumentation is extended to use priority information over arguments, with different audiences being represented by different orderings over arguments. This priority information is used to prefer some extensions over others, and so capturing audiences in this way allows different "values" (e.g., the values of a socialist or a capitalist or an ecologist) to be applied to reasoning with the argumentation framework. This setup is therefore restricted to that of reasoning with existing arguments. There is no conceptualization of how a given audience might respond to new arguments that might be brought into play. There is also no way that the preferences can be automatically generated from knowledge about the audience. And, of course, the more general shortcomings of abstract argumentation that were raised in chapters 2 and 3 apply.

The need for taking the audience into account for higher impact and more convincing logic-based argumentation was first raised in [Hun04b, Hun04a], and these proposals will be explored in chapter 6.

5 Comparing and Rationalizing Arguments

In this chapter, we present some techniques for comparing and rationalizing arguments within an argument tree.

In the first part of the chapter, we present some ways of comparing arguments based on the nature of the conflicts arising between arguments and their canonical undercuts in an argument tree. We start by considering a criterion for whether an argument at the root of an argument tree is warranted based on the structure of the argument tree. For this, we introduce the notion of a judging function. Considering the structure of the tree then leads us to the notion of "degree of undercut," which captures how much an undercut conflicts with its parent. Both the judge function and the framework for evaluating the degree of undercut are types of intrinsic analysis.

In the second part of the chapter, we present some techniques for rationalizing argument trees and argument structures using intrinsic analyses. By this, we mean techniques for simplifying arguments and trees. Even for a small knowledgebase, the number of arguments can be overwhelming, and therefore simplifying constellations of them can be helpful. Therefore, we propose rationalization techniques including (1) pruning arguments (such as those that have a degree of undercut that is below a certain threshold) and (2) merging arguments to create fewer undercuts but without losing vital information. For this, we will consider compressing and condensing as types of merging. We also consider how commitment for the support of arguments in an argument tree can be shifted from defenders to attackers in an argument tree.

Rationalization of arguments is part of a process of editing a set of arguments and counterarguments to allow focusing on key issues, and as such, it is an important approach to selection of arguments and reformation of constellations of arguments and counterarguments. In this chapter, the rationalization techniques are driven by intrinsic analysis.

5.1 Judging Argument Trees

In general, argument judgment involves taking a constellation of arguments and counterarguments relevant to a claim and evaluating whether the claim is warranted (in a sense, acceptable or valid) in that constellation according to some principled criteria. While it is not always necessary to undertake argument judgment with a constellation of arguments and counterarguments, there are plenty of situations in both professional and everyday life where it is essential to undertake argument judgment. Consider, for example, decision making in a clinical situation where a clinician is deciding on a diagnosis for a patient problem prior to considering a treatment plan. Here it may be essential to determine according to some criteria which diagnosis is warranted.

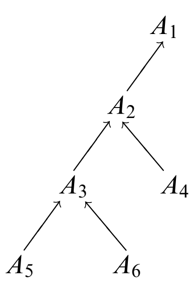
For argument judgment, we use a function, called the judge function, to evaluate whether the subject of an argument tree is, in a sense, warranted. The following definition for the judge function has been adapted from the "dialectical tree marking" of Garcia and Simari for defeasible logic programming (DeLP) [GS04]. In this, each node in an argument tree is labeled as either defeated or undefeated. Deciding whether a node is defeated or undefeated depends on whether or not all its children are defeated.

Definition 5.1.1 The **judge function**, denoted Judge, from the set of argument trees to $\{\mathsf{Warranted}, \mathsf{Unwarranted}\}$, is such that $\mathsf{Judge}(T) = \mathsf{Warranted}$ iff $\mathsf{Mark}(A_r) = \mathsf{Undefeated}$ where A_r is the root node of T and Mark is defined as follows:

1. For all $A_i \in \mathsf{Nodes}(T)$, $\mathsf{Mark}(A_i) = \mathsf{Undefeated}$
   iff for all children A_j of A_i, $\mathsf{Mark}(A_j) = \mathsf{Defeated}$.
2. For all $A_i \in \mathsf{Nodes}(T)$, $\mathsf{Mark}(A_i) = \mathsf{Defeated}$
   iff there is a child A_j of A_i such that $\mathsf{Mark}(A_j) = \mathsf{Undefeated}$.

Thus for all leaves A_l of an argument tree, $\mathsf{Mark}(A_l) = \mathsf{Undefeated}$.

Example 5.1.1 Consider the following argument tree T:



The arguments are listed below together with the mark for each of them:

Argument	Mark(A_i)
A_1 $\langle\{\delta, \delta \to \sigma\}, \sigma\rangle$	Undefeated
A_2 $\langle\{\neg\gamma, \neg\gamma \to \neg\delta\}, \diamond\rangle$	Defeated
A_3 $\langle\{\alpha, \beta, \alpha \wedge \beta \to \gamma\}, \diamond\rangle$	Defeated
A_4 $\langle\{\gamma \vee \delta\}, \diamond\rangle$	Undefeated
A_5 $\langle\{\neg\alpha\}, \diamond\rangle$	Undefeated
A_6 $\langle\{\neg\beta\}, \diamond\rangle$	Undefeated

Hence, Judge(T) = Warranted holds. The intuition in this example is that A_4 is sufficient to defeat A_2 irrespective of A_3, and so the existence of A_5 and/or A_6 does not affect the ability of A_4 to defeat A_2 and hence allow A_1 to be undefeated.

While using the tree structure for judging whether the root of an argument is warranted is in some respects intuitive, we cannot always assume that all undercuts are equal in their import. In other words, there may be other criteria that we need to take into account if we wish to decide whether an argument is defeated or undefeated. In the next section, we visit this question by introducing a more refined way of considering undercuts. In the next chapter, we will return to this question by considering how the audience may view the arguments. Thus, developments in both intrinsic and extrinsic analysis techniques offer new alternatives for judging argument trees.

5.2 Degree of Undercut

An argument conflicts with each of its undercuts by the very definition of an undercut. Now, some may conflict more than others, and some may conflict a little while others conflict a lot. To illustrate this, consider the following pairs of an argument (of the kind you often hear from a salesperson) together with a counterargument:

It is affordable, and it is both light and robust.

It isn't both light and robust.

We here have an argument whose reason is conjunctive, consisting of three propositions stating that the item is affordable as well as light and robust; hence, the claim that the item is affordable happens to be an easy conclusion. The counterargument does not challenge the claim but objects that one of the other two propositions must be wrong.

It is affordable, and it is both light and robust.

It isn't both light and robust; it is neither.

Again, the counterargument does not challenge the claim, but, in contrast to the previous case, the counterargument objects that both other propositions must be wrong. Hence, there seems to be more conflict here.

All this is meant to illustrate the idea that the amount of conflict between an argument and its undercuts need not always be the same. Next is a more abstract analysis in the case of canonical undercuts in order to pave the way for the definition of a measure of conflict in argument trees.

To begin with, remember that two canonical undercuts for the same argument have the same consequent. Therefore, if based on consequents, a measure of the conflict between an argument and its canonical undercuts would give the same result for all canonical undercuts of a given argument. In order to avoid such a collapse, a measure of undercut must then reflect the conflict between *supports* (the support of the argument and the support of the canonical undercut at hand).

We can now turn to investigating representative, albeit abstract, cases:

$$\begin{array}{ccc} T_1 & T_2 & T_3 \\ \langle\{\alpha\},\alpha\rangle & \langle\{\alpha\wedge\beta\wedge\gamma\},\alpha\rangle & \langle\{\alpha\wedge\beta\wedge\gamma\},\alpha\rangle \\ \uparrow & \uparrow & \uparrow \\ \langle\{\neg\alpha\},\diamond\rangle & \langle\{\neg\alpha\},\diamond\rangle & \langle\{\neg\beta\},\diamond\rangle \end{array}$$

All of $T_1,\ldots,T_3$ are argument trees (with respect to different Δ's) for α. In T_1, the support for the root is $\{\alpha\}$ and the support for the undercut is $\{\neg\alpha\}$. This can be described as a propositional conflict where $\neg\alpha$ is against α. In T_2, the support for the root is $\{\alpha\wedge\beta\wedge\gamma\}$ and the support for the undercut is $\{\neg\alpha\}$. This can be viewed as equivalent to T_1, since the conflict is merely with respect to one atom, namely, α. In T_3, the support for the root is $\{\alpha\wedge\beta\wedge\gamma\}$ but the support for the undercut is $\{\neg\beta\}$. This can also be viewed as equivalent to T_1, since the conflict is only with respect to one atom, β, disregarding the fact that the support of the undercut now fails to contradict the consequent of the root.

$$\begin{array}{ccc} T_4 & T_5 & T_6 \\ \langle\{\alpha\wedge\beta\wedge\gamma\},\alpha\wedge\beta\wedge\gamma\rangle & \langle\{\alpha\wedge\beta\wedge\gamma\},\alpha\wedge\beta\wedge\gamma\rangle & \langle\{\alpha\wedge\beta\wedge\gamma\},\alpha\wedge\beta\wedge\gamma\rangle \\ \uparrow & \uparrow & \uparrow \\ \langle\{\neg\gamma\},\diamond\rangle & \langle\{\neg(\alpha\wedge\beta\wedge\gamma)\},\diamond\rangle & \langle\{\neg\alpha\wedge\neg\beta\wedge\neg\gamma\},\diamond\rangle \end{array}$$

All of $T_4,\ldots,T_6$ are argument trees (with respect to different Δ's) for $\alpha\wedge\beta\wedge\gamma$. In T_4, the support for the root is $\{\alpha\wedge\beta\wedge\gamma\}$ and the support for the undercut is $\{\neg\gamma\}$. Still disregarding consequents, this can be viewed as involving as much conflict as T_2 (see T_3). In T_5, the support for the root is $\{\alpha \wedge \beta \wedge \gamma\}$ and the support for the undercut is $\{\neg(\alpha \wedge \beta \wedge \gamma)\}$. Since $\neg(\alpha \wedge \beta \wedge \gamma)$ is logically equivalent to $\neg\alpha \vee \neg\beta \vee \neg\gamma$, the conflict only necessarily involves one atom. Hence, this can also be viewed as involving as much conflict as T_2. It can be established more directly that T_5 involves as much conflict as T_1 simply by noticing that $\neg(\alpha \wedge \beta \wedge \gamma)$ is against $\alpha \wedge \beta \wedge \gamma$ exactly like $\neg\alpha$ is against α (cf. the case for T_1 above). In T_6, the support for the root is $\{\alpha \wedge \beta \wedge \gamma\}$ and the support for the undercut is $\{\neg\alpha \wedge \neg\beta \wedge \neg\gamma\}$. Here, the conflict is much more substantial, since it involves all three atoms.

By these simple examples, we see that there is an intuitive difference in the amount of conflict between supports, and hence this can be taken as an intuitive starting point for defining the degree of undercut that an argument has against its parent.

Conflict of an argument with each of its undercuts is to be reflected by a position in an ordering (possibly a partial one) but not necessarily a numerical value in some interval (i.e., orders of magnitude are not necessarily needed).

Definition 5.2.1 A **degree of undercut** is a mapping $\mathsf{Degree} : \Omega \times \Omega \to O$ where $\langle O, \leq \rangle$ is some poset such that for $A_i = \langle \Phi_i, \alpha_i \rangle$ and $A_j = \langle \Phi_j, \alpha_j \rangle$ in Ω,

1. $\mathsf{Degree}(A_j, A) \leq \mathsf{Degree}(A_i, A)$ for all $A \in \Omega$ if $\mathsf{Cn}(\Phi_j) \subseteq \mathsf{Cn}(\Phi_i)$.
2. $\mathsf{Degree}(A_i, A_j)$ is minimal if $\Phi_i \cup \Phi_j \nvdash \perp$.

Of course, the intuitive meaning is that $\mathsf{Degree}(A, A')$ indicates how strongly A is contradicted by A'.

In general, we only consider $\mathsf{Degree}(A, A')$ when A' is a canonical undercut of A. Thus, the other values for $\mathsf{Degree}(A, A')$, when A' is not a canonical undercut of A, could a priori be arbitrary. As a special case, however, it seems right that two arguments A and A' that do not conflict with each other (so, neither is an undercut of the other) are such that $\mathsf{Degree}(A, A')$ is minimal (cf. the last clause in definition 5.2.1).

Definition 5.2.1 allows for very many possibilities, leaving you to choose a suitable mapping. It is even possible to specify a degree of undercut such that $\mathsf{Degree}(A, A') = \mathsf{Degree}(A', A)$ may fail, including for some A and A' rebutting each other.

Technically, any poset can do but there is at least one puzzling question if $\langle O, \leq \rangle$ happens to be deprived of a least element. In such a case, there can exist some arguments A, A', and A'' that do not conflict with each other: $\text{Degree}(A, A') = r_1$ and $\text{Degree}(A, A'') = r_2$ where r_1 and r_2 are both minimal but are not equal, $r_1 \neq r_2$. The intuitions do not seem to allow for a natural way to interpret that.

To focus our presentation in this chapter, we will restrict ourselves to O being the interval $[0, 1]$.

5.2.1 Degree of Undercut Based on Dalal Distance

The first idea that comes to mind when considering specifying a degree of undercut is to use a distance. In this section, we investigate a degree of undercut based on the so-called Dalal distance between pairs of models [Dal88].

Definition 5.2.2 Let $w_i, w_j \in \wp(\Pi)$, where $\mathscr{AT}$ is the set of all atoms in $\mathscr{L}$ (as defined in Appendix C) and Π is a finite nonempty subset of $\mathscr{AT}$. The **Dalal distance** between w_i and w_j, denoted $\text{Dalal}(w_i, w_j)$, is the difference in the number of atoms assigned true:

$$\text{Dalal}(w_i, w_j) = |w_i - w_j| + |w_j - w_i|$$

Example 5.2.1 Let $w_1 = \{\alpha, \gamma, \delta\}$ and $w_2 = \{\beta, \gamma\}$, where $\{\alpha, \beta, \gamma, \delta\} \subseteq \Pi$. Then,

$$\begin{aligned}\text{Dalal}(w_1, w_2) &= |\{\alpha, \delta\}| + |\{\beta\}| \\ &= 3\end{aligned}$$

We are to evaluate the conflict between the support of an argument $\langle \Phi, \alpha \rangle$ and the support of a canonical undercut $\langle \Psi, \diamond \rangle$. We do it through the models of Φ and Ψ, restricted to Π. These are $\text{Models}(\Phi, \Pi)$ and $\text{Models}(\Psi, \Pi)$ (cf. definition C.2.7).

Example 5.2.2 Let $\Phi = \{\alpha \wedge \delta, \neg\phi, \gamma \vee \delta, \neg\psi, \beta \vee \gamma \vee \psi\} \subseteq \Delta$, and let Π be $\{\alpha, \beta, \gamma, \delta, \phi, \psi\} \subseteq \mathscr{AT}$. Should it be the case that $\Pi \neq \mathscr{AT}$, the set of all the models of Φ is a proper superset of $\text{Models}(\Phi, \Pi)$, as the latter consists exactly of the following models:

$\{\alpha, \beta, \gamma, \delta\}$

$\{\alpha, \beta, \delta\}$

$\{\alpha, \gamma, \delta\}$

To evaluate the conflict between two sets of formulae, we take a pair of models restricted to Π, one for each set, such that the Dalal distance is minimized. The degree of conflict is this distance divided by the maximum possible Dalal distance between a pair of models (i.e., $\log_2$ of the total number of models in $\wp(\Pi)$, which is $|\Pi|$).

Definition 5.2.3 The **degree of conflict** with respect to Π, denoted $\mathsf{Conflict}(\Phi, \Psi, \Pi)$, is as follows:

$$\frac{\min\{\mathsf{Dalal}(w_\Phi, w_\Psi) \mid w_\Phi \in \mathsf{Models}(\Phi, \Pi), w_\Psi \in \mathsf{Models}(\Psi, \Pi)\}}{|\Pi|}$$

Example 5.2.3 Let $\Pi = \{\alpha, \beta, \gamma, \delta\}$:

$\mathsf{Conflict}(\{\alpha \wedge \beta \wedge \gamma \wedge \delta\}, \{\neg\alpha \vee \neg\beta \vee \neg\gamma\}, \Pi) = 1/4$

$\mathsf{Conflict}(\{\alpha \wedge \beta \wedge \gamma \wedge \delta\}, \{\neg(\alpha \vee \beta)\}, \Pi) = 2/4$

$\mathsf{Conflict}(\{\alpha \wedge \beta \wedge \gamma \wedge \delta\}, \{\neg\alpha \wedge \neg\beta \wedge \neg\gamma\}, \Pi) = 3/4$

We obtain a degree of undercut by applying Conflict to supports.

Definition 5.2.4 Let $A_i = \langle \Phi_i, \alpha_i \rangle$ and $A_j = \langle \Phi_j, \alpha_j \rangle$ be arguments:

$$\mathsf{Degree}_\mathsf{C}(A_i, A_j) \stackrel{\text{def}}{=} \mathsf{Conflict}(\Phi_i, \Phi_j, \mathsf{Atoms}(\Phi_i \cup \Phi_j))$$

Clearly, if A_i is an undercut for A_j, then $\mathsf{Degree}_\mathsf{C}(A_i, A_j) > 0$.

Example 5.2.4 Below we give the values for the degree of undercut of the argument $\langle\{\alpha, \beta, \gamma\}, \ldots\rangle$ (the consequent is unimportant here) by some of its canonical undercuts:

$\mathsf{Degree}_\mathsf{C}(\langle\{\alpha, \beta, \gamma\}, \ldots\rangle, \langle\{\neg\alpha \wedge \neg\beta \wedge \neg\gamma\}, \diamond\rangle) = 1$

$\mathsf{Degree}_\mathsf{C}(\langle\{\alpha, \beta, \gamma\}, \ldots\rangle, \langle\{\neg\alpha \wedge \neg\beta\}, \diamond\rangle) = 2/3$

$\mathsf{Degree}_\mathsf{C}(\langle\{\alpha, \beta, \gamma\}, \ldots\rangle, \langle\{\neg\alpha \vee \neg\beta \vee \neg\gamma\}, \diamond\rangle) = 1/3$

$\mathsf{Degree}_\mathsf{C}(\langle\{\alpha, \beta, \gamma\}, \ldots\rangle, \langle\{\neg\alpha\}, \diamond\rangle) = 1/3$

Example 5.2.5 Consider the following argument trees:

$$\begin{array}{cccc}
 & T_7 & T_8 & T_9 \\
A & \langle\{\alpha \wedge \beta \wedge \gamma\}, \alpha\rangle & \langle\{\alpha \wedge \beta \wedge \gamma\}, \alpha\rangle & \langle\{\alpha \wedge \beta \wedge \gamma\}, \alpha\rangle \\
 & \uparrow & \uparrow & \uparrow \\
A' & \langle\{\neg\alpha \vee \neg\beta \vee \neg\gamma\}, \diamond\rangle & \langle\{\neg\alpha \wedge \neg\gamma\}, \diamond\rangle & \langle\{\neg\alpha \wedge \neg\beta \wedge \neg\gamma \wedge \neg\delta\}, \diamond\rangle \\
 & \mathsf{Degree}_\mathsf{C}(A, A') = 1/3 & \mathsf{Degree}_\mathsf{C}(A, A') = 2/3 & \mathsf{Degree}_\mathsf{C}(A, A') = 3/4
\end{array}$$

Example 5.2.6 More generally, let $A_1 = \langle\{\neg(\alpha_1 \vee \cdots \vee \alpha_n)\}, \neg(\alpha_1 \wedge \cdots \wedge \alpha_n)\rangle$, $A_2 = \langle\{\neg\alpha_1 \vee \cdots \vee \neg\alpha_n\}, \neg(\alpha_1 \wedge \cdots \wedge \alpha_n)\rangle$, $A_3 = \langle\{\neg\alpha_1\}, \neg(\alpha_1 \wedge \cdots \wedge \alpha_n)\rangle$, $A_4 = \langle\{\alpha_1 \wedge \cdots \wedge \alpha_n\}, \alpha_1\rangle$:

$$\mathsf{Degree}_\mathsf{C}(A_4, A_1) = n/n$$

$$\mathsf{Degree}_\mathsf{C}(A_4, A_2) = 1/n$$

$$\mathsf{Degree}_\mathsf{C}(A_4, A_3) = 1/n$$

Proposition 5.2.1 Let $A_i = \langle\Phi_i, \alpha_i\rangle$ and $A_j = \langle\Phi_j, \alpha_j\rangle$ be arguments:

1. $0 \leq \mathsf{Degree}_\mathsf{C}(A_i, A_j) \leq 1$
2. $\mathsf{Degree}_\mathsf{C}(A_i, A_j) = \mathsf{Degree}_\mathsf{C}(A_j, A_i)$
3. $\mathsf{Degree}_\mathsf{C}(A_i, A_j) = 0$ iff $\Phi_\mathrm{i} \cup \Phi_\mathrm{j} \nvdash \perp$

5.2.2 Degree of Undercut Based on Propositional Contradiction

Another possibility for a degree of undercut comes from counting how many of the atoms occurring in the support of the argument are contradicted by the counterargument (what $\mathsf{Degree}_\mathsf{C}$ does may sometimes give quite a different result; some examples are given later). Consider the argument $\langle\{\alpha\}, \alpha\rangle$ and its canonical undercut $\langle\{\neg\alpha \wedge \neg\beta \wedge \neg\gamma\}, \Diamond\rangle$. The argument is fully undermined because it asserts nothing about β and γ, but the only thing it asserts, namely α, is contradicted by the counterargument. We could hold, informally, that the argument is 100% contradicted. Technically, we need formulae to be in disjunctive normal form (cf. definition C.1.6) after tautologous pairs of disjuncts are removed.

Definition 5.2.5 Let $\langle\Phi, \alpha\rangle$ and $\langle\Psi, \beta\rangle$ be arguments. Let φ and ψ be disjunctive normal forms for Φ and Ψ, respectively,

$$\varphi = \bigvee_{i=1\ldots p_\varphi} \varphi_{i1} \wedge \cdots \wedge \varphi_{in_i}$$

$$\psi = \bigvee_{j=1\ldots p_\psi} \psi_{j1} \wedge \cdots \wedge \psi_{jm_j}$$

where φ_{ih} and ψ_{jk} are literals (i.e., atoms or negated atoms). Then,

$$\mathsf{Degree}_\mathsf{E}(\langle\Phi, \alpha\rangle, \langle\Psi, \beta\rangle) \stackrel{\text{def}}{=} \min_{\substack{i=1\ldots p_\varphi \\ j=1\ldots p_\psi}} \frac{|\{\varphi_{i1}, \ldots, \varphi_{in_i}\} \cap \{\overline{\psi_{j1}}, \ldots, \overline{\psi_{jm_j}}\}|}{|\{\varphi_{i1}, \ldots, \varphi_{in_i}\}|}$$

where

$$\overline{\psi_{jk}} = \begin{cases} \neg\gamma & \text{if } \psi_{jk} \text{ is an atom } \gamma \\ \gamma & \text{if } \psi_{jk} \text{ is a negated atom } \neg\gamma \end{cases}$$

Example 5.2.7 Consider two arguments A and A' whose support consists of a single formula as follows:

A: $(\alpha \wedge \beta \wedge \gamma) \vee (\neg\alpha \wedge \delta) \vee (\neg\gamma \wedge \varepsilon)$

A': $(\neg\alpha \wedge \neg\delta \wedge \neg\varepsilon) \vee (\alpha \wedge \neg\beta \wedge \neg\gamma \wedge \neg\varepsilon)$

Then, the normalized values for each intersection are as follows:

$$\begin{array}{llll}
\{\alpha,\beta,\gamma\} & \cap & \{\overline{\neg\alpha},\overline{\neg\delta},\overline{\neg\varepsilon}\} & 1/3 \\
\{\alpha,\beta,\gamma\} & \cap & \{\bar{\alpha},\overline{\neg\beta},\overline{\neg\gamma},\overline{\neg\varepsilon}\} & 2/3 \\
\{\neg\alpha,\delta\} & \cap & \{\overline{\neg\alpha},\overline{\neg\delta},\overline{\neg\varepsilon}\} & 1/2 \\
\{\neg\alpha,\delta\} & \cap & \{\bar{\alpha},\overline{\neg\beta},\overline{\neg\gamma},\overline{\neg\varepsilon}\} & 1/2 \\
\{\neg\gamma,\varepsilon\} & \cap & \{\overline{\neg\alpha},\overline{\neg\delta},\overline{\neg\varepsilon}\} & 1/2 \\
\{\neg\gamma,\varepsilon\} & \cap & \{\bar{\alpha},\overline{\neg\beta},\overline{\neg\gamma},\overline{\neg\varepsilon}\} & 1/2
\end{array}$$

Hence $\mathsf{Degree_E}(A, A') = 1/3$ (but $\mathsf{Degree_E}(A', A) = 1/4$).

Example 5.2.8 Consider the following argument trees:

$$\begin{array}{llll}
 & T_7 & T_8 & T_9 \\
A & \langle\{\alpha\wedge\beta\wedge\gamma\},\alpha\rangle & \langle\{\alpha\wedge\beta\wedge\gamma\},\alpha\rangle & \langle\{\alpha\wedge\beta\wedge\gamma\},\alpha\rangle \\
 & \uparrow & \uparrow & \uparrow \\
A' & \langle\{\neg\alpha\vee\neg\beta\vee\neg\gamma\},\Diamond\rangle & \langle\{\neg\alpha\wedge\neg\gamma\},\Diamond\rangle & \langle\{\neg\alpha\wedge\neg\beta\wedge\neg\gamma\wedge\neg\delta\},\Diamond\rangle
\end{array}$$

For these trees, we can make the following observations:

T_7: $\mathsf{Degree_E}(A, A') = 1/3$ because the support of the undercut need not contradict more than one atom in $\{\alpha \wedge \beta \wedge \gamma\}$

T_8: $\mathsf{Degree_E}(A, A') = 2/3$ because the support of the undercut contradicts two atoms in $\{\alpha \wedge \beta \wedge \gamma\}$

T_9: $\mathsf{Degree_E}(A, A') = 1$ because the support of the undercut contradicts all three atoms in $\{\alpha \wedge \beta \wedge \gamma\}$

We now briefly illustrate how these two notions of degree of undercut, $\mathsf{Degree_C}$ and $\mathsf{Degree_E}$, compare.

Example 5.2.9 For A and A' below, the undercut A' seems much less of a challenge to A than A is to A' as reflected by $\mathsf{Degree_E}(A, A') \neq \mathsf{Degree_E}(A', A) = 1$:

A: $\langle\{\alpha \wedge \beta \wedge \cdots \wedge \pi\}, \beta\rangle$

A': $\langle\{\neg\alpha \wedge \neg\pi\}, \neg\pi\rangle$

All conjuncts in the support of A' are refuted by the support of A, but not vice versa; still $\mathsf{Degree}_{\mathsf{C}}(A, A') = \mathsf{Degree}_{\mathsf{C}}(A', A)$.

Inversely, there are, however, examples where commutativity looks appealing for a degree of undercut.

Example 5.2.10 In contrast to $\mathsf{Degree}_{\mathsf{C}}(A, A') = \mathsf{Degree}_{\mathsf{C}}(A', A)$, consider the following two arguments:

A: $\langle\{\alpha \wedge \beta \wedge \gamma\}, \alpha\rangle$

A': $\langle\{\neg\alpha\}, \neg\alpha\rangle$

A and A' are rebuttals of each other, but $\mathsf{Degree}_{\mathsf{E}}(A, A') \neq \mathsf{Degree}_{\mathsf{E}}(A', A)$.

Finally, there are cases in which neither $\mathsf{Degree}_{\mathsf{C}}$ nor $\mathsf{Degree}_{\mathsf{E}}$ seems to give an intuitive result.

Example 5.2.11 For A and A' below, A' agrees with the consequent of A but both $\mathsf{Degree}_{\mathsf{E}}(A, A')$ and $\mathsf{Degree}_{\mathsf{C}}(A, A')$ are large:

A: $\langle\{\alpha \wedge \beta \wedge \cdots \wedge \pi\}, \alpha\rangle$

A': $\langle\{\alpha \wedge \neg\beta \wedge \cdots \wedge \neg\pi\}, \neg\pi\rangle$

Hence, the very many possibilities arising from definition 5.2.1 should be taken advantage of to fit with the requirements and peculiarities of a given application.

5.2.3 Degree of Undercut Based on Reserved Atoms

Some atoms may be regarded as important in the case of conflict; others may be regarded as less important. If we have an argument about whether to have a barbecue on the weekend, then contradictions involving some of the atoms may be regarded as critically important, for example, undercuts involving atoms such as *it will rain over the weekend* may be regarded as being very important, and those undercuts would be weighted accordingly, whereas undercuts involving other atoms such as *we are not good at barbecue cooking* may be regarded as unimportant, and undercuts involving those atoms would have less weight.

Definition 5.2.6 The **degree of undercut with reserved atoms** $\Pi \subseteq \mathcal{AT}$ is such that for $A_i = \langle\Phi, \alpha\rangle$ and $A_j = \langle\Psi, \beta\rangle$,

$$\mathsf{Degree}_{\Pi}(A_i, A_j) = |\mathsf{Atoms}(\Phi) \cap \mathsf{Atoms}(\Psi) \cap \Pi|$$

Intuitively, the reserved atoms in Π are those propositions that are felt to be important. The degree here is based on the important propositions that occur both in the support of the first argument (in symbols, $\mathsf{Atoms}(\Phi) \cap \Pi$) and in the support of the second argument (in symbols, $\mathsf{Atoms}(\Psi) \cap \Pi$). This amounts to $\mathsf{Atoms}(\Phi) \cap \mathsf{Atoms}(\Psi) \cap \Pi$, and enumerating it gives the degree of undercut with respect to Π.

Example 5.2.12 Consider the following arguments, such that A_2 is a canonical undercut for A_1 and A_3 is a canonical undercut for A_2:

$$A_1 = \langle \{\alpha, \alpha \to \beta\}, \beta \rangle$$

$$A_2 = \langle \{\delta, \delta \to \neg\alpha\}, \diamond \rangle$$

$$A_3 = \langle \{\gamma, \gamma \to \neg\delta\}, \diamond \rangle$$

Let the reserved atoms be $\Pi = \{\alpha, \beta\}$:

$$\{\alpha, \alpha \to \beta\} \cap \{\alpha, \beta\} = \{\alpha, \beta\} \qquad \text{(reserved atoms in } \mathsf{Support}(A_1))$$

$$\{\delta, \delta \to \neg\alpha\} \cap \{\alpha, \beta\} = \{\alpha\} \qquad \text{(reserved atoms in } \mathsf{Support}(A_2))$$

$$\{\gamma, \gamma \to \neg\delta\} \cap \{\alpha, \beta\} = \{\ \} \qquad \text{(reserved atoms in } \mathsf{Support}(A_3))$$

Therefore

$$\mathsf{Degree}_\Pi(A_1, A_2) = 1$$

$$\mathsf{Degree}_\Pi(A_2, A_3) = 0$$

Clearly, the idea of reserved atoms can also be used in combination with another approach (such as Dalal distance) to define other instances for the notion of a degree of undercut.

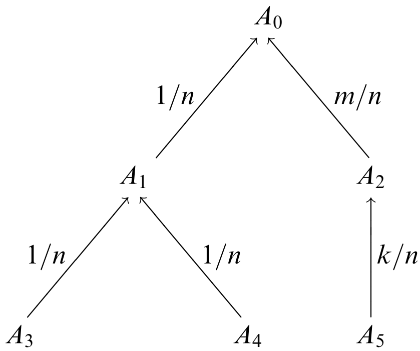
5.2.4 Labeled Argument Trees

We now introduce labeled argument trees. That is, we label each arc with the degree of undercut.

Definition 5.2.7 A **labeled argument tree** is an argument tree such that if A_j is a child of A_i in the tree, then the arc from A_j to A_i is labeled $\mathsf{Degree}(A_i, A_j)$.

A labeled argument tree provides extra information that leads to a useful enhancement of the original argument tree.

Example 5.2.13 Provided $A_0, A_1, A_2, \ldots, A_5$ as well as k, m, n (where $k < n$ and $m < n$) conform with definition 5.2.7, here is a labeled argument tree in abstract form:



In this labeled argument tree, if n is significantly greater than 1, then it may be natural to concentrate our attention on the right-most branch of the tree, since if, in addition, m is close to n, then A_2 is an important undercut of A_0.

The tension of an argument tree is the cumulative conflict obtained from all the undercuts in the tree.

Definition 5.2.8 Let T be an argument tree, and let A_r be the root node. The **degree of tension** in T, denoted $\mathsf{Tension}(T)$, is given by the value of $\mathsf{Retension}(A_r)$, where for any node A_i in the tree, if A_i is a leaf, then $\mathsf{Retension}(A_i) = 0$; otherwise $\mathsf{Retension}(A_i)$ is

$$\sum_{A_j \in \mathsf{Undercuts}(A_i)} \mathsf{Retension}(A_j) + \mathsf{Degree}(A_i, A_j)$$

Here, measuring conflicts between arguments and undercuts requires orders of magnitude. In fact, it must now be assumed that the poset $\langle O, \leq \rangle$ comes with an additive measure (written $+$ in the above definition).

Tension provides a useful way to refine the value given by the degree of undercut: The latter merely indicates that an undercut with degree $3k/n$ is three times more important than an undercut with degree k/n, but this may need to be reconsidered in view of the overall tension (e.g., if huge).

Tension also provides a useful summary of the contentiousness of the initiating argument (i.e., the root argument). If the tension is relatively low, then we have more confidence that the initiating argument is in some sense warranted. For example, if we are deciding to buy a house and we have a shortlist of house $H1$ and house $H2$, we may build an argument tree T_1 with subject *Buy* $H1$ and an argument tree T_2 with subject *Buy* $H2$. Thus, if the tension of T_1 is much higher than T_2, then we may be more confident in a decision to buy $H1$ rather than $H2$.

By definition 5.2.8, tension is fully determined once degree of undercut is specified. Tension is nonetheless a useful complement because definition 5.2.1 is fairly liberal as already mentioned. Indeed, definition 5.2.1 is rather meant to act as a guideline, and there are many possibilities in choosing Degree. In fact, Degree need not be definable from the non-labeled argument tree: Degree may convey extraneous information (e.g., a certain undercut gets more weight because it concerns a certain topic or it comes from a certain source, etc.). In other words, Degree and Tension may be tailored to the need of a particular application.

Labeled complete argument trees are of particular interest in that they contain all the information, but some are needlessly large. In the next section, we consider a further comparison relation for argumentation, and then we turn to the question of getting rid of redundant and/or negligible information in argument trees as a process of reformation of argument trees. As part of our presentation of rationalization, we will show that by using degree of undercut, we can focus on the key arguments in an argument tree.

5.3 Pruning of Argument Trees

Pruning an argument tree involves deleting one or more nodes from the tree such that when a node is deleted, all its children are deleted. In other words, for each node deleted, the subtree, for which the node is the root, is deleted. To illustrate pruning, we consider label-free pruning and label-based pruning in this section.

Careful: From now on, we relax the definition of an argument tree so that not all canonical undercuts need be present. We accordingly call an argument tree satisfying definition 3.5.1 a complete argument tree.

5.3.1 Label-Free Pruning of Argument Trees

Before considering degrees of undercuts, there is a way to ignore some weak subarguments without even taking into account any quantitative value about their disagreement with the undercut argument. This gives us our first proposal for rationalization.

Definition 5.3.1 Let $\langle \Psi, \beta \rangle$ be an undercut of $\langle \Phi, \alpha \rangle$. Then, $\langle \Psi, \beta \rangle$ is an **overzealous undercut** of $\langle \Phi, \alpha \rangle$ whenever $\Psi \vdash \alpha$.

The idea can be illustrated as follows: Andrew says that *Hume is not hard to understand, and the new BMW is a six cylinder, and the new BMW has ABS, and the new BMW has injection* so he claims *the new*

BMW is a luxury car. Basil replies saying *Hume is hard to understand and ... here are some impressive facts about the new BMW ... and the new BMW has injection*, so he contradicts Andrew by pointing out that the support for his argument is wrong. Certainly Basil's counterargument is weak, plagued with irrelevance (he contradicts something apart from the main subject of the new BMW).

There is a parallel to be drawn here: Remember that the fallacy of *argumentum ad hominem* consists of attacking the person who puts the argument forward instead of attacking the argument itself. Consider the (usual) situation in which the proponent Phil of an argument A regards himself to be a good person. Formally, the statement (of course, the example neither claims nor disclaims that Phil is a good person) that Phil is a good person becomes part of the support as a conjunct. Now, *argumentum ad hominem* amounts to denigrating the consequent of A just by disparaging Phil. Formally, this gives rise to an argument A' whose consequent denies that Phil is a good person.

Overzealous undercuts should be deleted, turning an argument tree T into a "focused" argument tree denoted $\mathsf{Zealousfree}(T)$.

Proposition 5.3.1 For an argument tree T, if $T' = \mathsf{Zealousfree}(T)$, then $|\mathsf{Nodes}(T')| \leq |\mathsf{Nodes}(T)|$, $\mathsf{Tension}(T') \leq \mathsf{Tension}(T)$, $\mathsf{Depth}(T') \leq \mathsf{Depth}(T)$, and $\mathsf{Width}(T') \leq \mathsf{Width}(T)$.

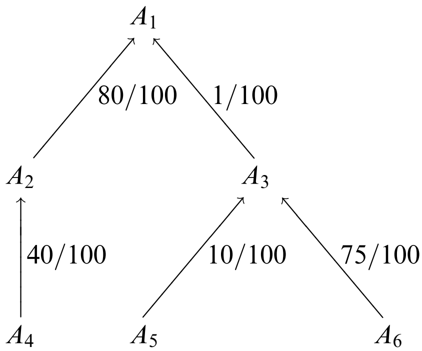
Observe that an argument need not have an overzealous undercut (Δ being fixed), but if it has one, then at least one, of its canonical undercuts is overzealous.

5.3.2 Label-Based Pruning of Argument Trees

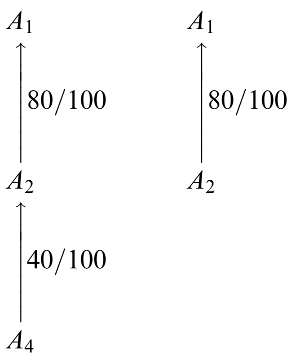
For our second proposal for rationalization, we consider pruning argument trees. For this, we introduce a threshold for a minimum degree of undercut. If an undercut has a degree of undercut below the threshold, then the undercut is dropped, together with any offspring of that undercut.

Definition 5.3.2 A threshold, denoted τ, is a value in O such that if T is an argument tree, $\mathsf{Prune}(T, \tau)$ is the **pruned argument tree** obtained from T by removing each undercut A_j of an argument A_i if $\mathsf{Degree}(A_i, A_j) < \tau$ and, for any undercut removed, all the offspring of the undercut are also removed.

Example 5.3.1 Let T be the following labeled argument tree:



Below, the left argument tree is $\mathsf{Prune}(T, 0.3)$, and the right one is $\mathsf{Prune}(T, 0.5)$:



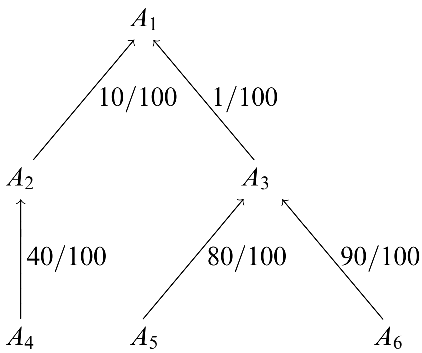
Thus, pruning of argument trees allows us to focus our attention on the most important undercuts. The utility of label-based pruning is particularly pronounced when considering first-order knowledgebases.

Proposition 5.3.2 For all $i \in O$, if $T' = \mathsf{Prune}(T, i)$, then $\mathsf{Tension}(T') \leq \mathsf{Tension}(T)$, $|\mathsf{Nodes}(T')| \leq |\mathsf{Nodes}(T)|$, $\mathsf{Depth}(T') \leq \mathsf{Depth}(T)$, and $\mathsf{Width}(T') \leq \mathsf{Width}(T)$.

Also, $\mathsf{Prune}(T, 0) = T$ and $\mathsf{Prune}(T, 1) = T_\emptyset$, where $T_\emptyset$ is the empty tree. For all $i \in [0, 1]$, if T is an argument tree, then $\mathsf{Prune}(T, i)$ is an argument tree. However, if T is a complete argument tree, then $\mathsf{Prune}(T, i)$ is not necessarily a complete argument tree.

We suggest that it is justifiable to undertake pruning based on degree of undercut and, as a result, ignore the offspring of the undercut even if an offspring goes above the threshold: If the offspring with large degree is an attacker of U, the undercut to be removed, then U is weaker, and this is a further reason not to consider it. If the offspring is a defender of U, then that defender anyway fails to make U stronger than it would if U was not attacked at all, in which case U is removed anyway (so it is coherent that U is removed).

Example 5.3.2 Consider the following argument tree. Here, there is a subtree rooted at A_3 that has a high tension:



If we remove the subtree rooted at A_3 because A_3 has a low degree of undercut, we reduce the tension of the overall argument tree. In other words, the original argument tree has a tension of 221/100, and the pruned tree (containing nodes A_1, A_2, and A_4) has a tension of 120/100. Thus, the tension is reduced by nearly half. However, this tension associated with A_3, A_5, and A_6 is really a distraction—the conflict between these arguments is not important for assessing the root argument A_1—so the influence of them on A_1 is low.

In a similar way to the above example, consider a big subtree with each labeled $1/n$ (i.e., small degree of undercut). Because the subtree is big, the tension for the subtree is large. However, $\mathsf{Prune}(T, \tau)$, where $\tau > 1/n$, means that all the big subtree is removed. This is appropriate—again, the conflict between these arguments is not important for assessing the root argument, and so the influence of them on the root argument is low.

5.4 Compressing Argument Trees

For our third proposal for rationalization, we consider how we may compress argument trees. Compression combines arguments without loss of essential information. Compression merges siblings in order to reduce the number of arguments and to reduce the redundancy arising by having arguments equivalent in some sense, and to make appropriate simplifications of the syntax of some arguments.

Definition 5.4.1 Let T_1 and T_2 be argument trees formed from Δ such that $\mathsf{Subject}(T_1) = \mathsf{Subject}(T_2)$. T_2 is a **compression** of T_1 iff there is a surjection $G : \mathsf{Nodes}(T_1) \mapsto \mathsf{Nodes}(T_2)$ such that for all $B \in \mathsf{Nodes}(T_2)$,

$$\mathsf{Cn}(\mathsf{Support}(B)) = \mathsf{Cn}\left(\bigcup_{A \in G^{-1}(B)} \mathsf{Support}(A)\right)$$

G is called a compression function.

Example 5.4.1 T_3 is a compression of T_2

$$\begin{tikzcd}[column sep=small] & T_2 & & T_3 \\ & \langle\{\alpha\wedge\beta\},\alpha\wedge\beta\rangle & & \langle\{\alpha\wedge\beta\},\alpha\wedge\beta\rangle \\ \langle\{\neg\alpha\vee\neg\beta\},\Diamond\rangle \arrow[ur] & & \langle\{\neg\alpha\wedge\neg\beta\},\Diamond\rangle \arrow[ul] & \langle\{\neg\alpha\wedge\neg\beta\},\Diamond\rangle \arrow[u] \end{tikzcd}$$

while each of T_2 and T_3 is a compression of T_1:

$$\begin{tikzcd}[column sep=small] & & T_1 & & \\ & & \langle\{\alpha\wedge\beta\},\alpha\wedge\beta\rangle & & \\ \langle\{\neg\alpha\vee\neg\beta\},\Diamond\rangle \arrow[urr] & \langle\{\neg\alpha\},\Diamond\rangle \arrow[ur] & & \langle\{\neg\beta\},\Diamond\rangle \arrow[ul] & \langle\{\neg\alpha\wedge\neg\beta\},\Diamond\rangle \arrow[ull] \end{tikzcd}$$

In contrast to the previous example where T_1 is a complete argument tree, next is an example of compression of an incomplete argument tree.

Example 5.4.2 T_2 is a compression of T_1 (assuming $\gamma \to \neg\alpha \wedge \neg\beta \in \Delta$):

$$\begin{tikzcd}[column sep=small] & T_1 & & T_2 \\ & \langle\{\alpha\wedge\beta\},\alpha\wedge\beta\rangle & & \langle\{\alpha\wedge\beta\},\alpha\wedge\beta\rangle \\ \langle\{\gamma,\gamma\to\neg\alpha\},\Diamond\rangle \arrow[ur] & & \langle\{\gamma,\gamma\to\neg\beta\},\Diamond\rangle \arrow[ul] & \langle\{\gamma,\gamma\to\neg\alpha\wedge\neg\beta\},\Diamond\rangle \arrow[u] \end{tikzcd}$$

Compression does not affect the depth of the tree, but it has a downward effect on the branching.

Proposition 5.4.1 If T_2 is a compression of T_1, then $\mathsf{Tension}(T_1) \leq \mathsf{Tension}(T_2)$, $\mathsf{Depth}(T_1) = \mathsf{Depth}(T_2)$, $|\mathsf{Nodes}(T_2)| \leq |\mathsf{Nodes}(T_1)|$, and $\mathsf{Width}(T_2) \leq \mathsf{Width}(T_1)$.

Compression is not unique, and there are limits to compression, for example, when an argument tree is a chain, and when all pairs of siblings have supports that are mutually contradictory.

Undercutting is preserved in compression. The following proposition explains the nature of offspring when some siblings are merged in a tree compression.

Proposition 5.4.2 Let T_1 and T_2 be argument trees. If T_2 is a compression of T_1, with compression function G, and $A \in \mathsf{Nodes}(T_1)$, and $C \in \mathsf{Undercuts}(T_1, A)$, and $G(A) = B$, then $G(C) \in \mathsf{Undercuts}(T_2, B)$.

Proposition 5.4.3 Let T_1 and T_2 be argument trees. (1) If T_2 is a compression of T_1 and T_1 is a compression of T_2, then T_1 and T_2 are isomorphic (the compression function is a bijection); (2) if T_3 is a compression of T_2 and T_2 is a compression of T_1, then T_3 is a compression T_1; and (3) T_1 is a compression of T_1. Thus, "is a compression of" is a partial ordering relation over nonisomorphic argument trees.

Proposition 5.4.4 Let T_1 be an argument tree. If A_1 and A_2 are siblings in T_1, and $\mathsf{Cn}(\mathsf{Support}(A_1)) = \mathsf{Cn}(\mathsf{Support}(A_2))$, then there is an argument tree T_2 such that T_2 is a compression of T_1, with compression function G, and $G(A_1) = G(A_2)$.

Compressing an argument tree always yields an argument tree. In contrast, compressing need not turn a complete argument tree into a complete argument tree. Actually, proper compressing always yields an incomplete argument tree.

How well, then, does compressing argument trees meet our needs? In general, an incomplete argument tree is not a reliable account of the counterarguments (and counter-counterarguments, ...) of an argument. However, starting with an exhaustive account, namely, a complete argument tree, compressing only discards redundant information: The resulting incomplete argument tree can still be viewed as exhaustive.

As a rationalization technique, compressing argument trees offers significant advantages in presenting a constellation of arguments and counterarguments more concisely. Assumptions can be substantially simplified, and the number of arguments presented can be substantially reduced.

5.5 Condensing Argument Trees

For our fourth proposal for rationalizing argument trees, we consider condensing argument trees. A condensed argument tree is obtained by editing the set of assumptions and then building a new argument tree using the edited set of assumptions rather than editing the tree directly. A requirement of condensation is that the original and the condensed argument trees are complete argument trees.

Definition 5.5.1 Let T_1 be a complete argument tree from Δ_1, and let T_2 be a complete argument tree from Δ_2. T_2 is a **condensation** of T_1 iff there exist $S_1 \in \mathsf{Siblings}(T_1)$ and $S_2 \in \mathsf{Siblings}(T_2)$ where $|S_2| \leq |S_1|$ such that

1. $\Delta_1 - \bigcup_{A \in S_1} \mathsf{Support}(A) = \Delta_2 - \bigcup_{A' \in S_2} \mathsf{Support}(A')$
2. $\mathsf{Cn}(\bigcup_{A \in S_1} \mathsf{Support}(A)) = \mathsf{Cn}(\bigcup_{A' \in S_2} \mathsf{Support}(A')) \subset \mathscr{L}$

T_2 is also called a condensation of T_1 if there is T_3 such that T_2 is condensation of T_3 and T_3 is a condensation of T_1.

The requirement $\subset \mathscr{L}$ in definition 5.5.1 rules out cases such as S_1 and S_2 (given below) with $\alpha_1 \wedge \alpha_2 \wedge \alpha_3$ inconsistent, so S_2 may have nothing to do with S_1 bar ϕ.

$$S_1 = \{\langle\{\phi \wedge \alpha_1\}, \diamond\rangle, \langle\{\phi \wedge \alpha_2\}, \diamond\rangle, \langle\{\phi \wedge \alpha_3\}, \diamond\rangle\}$$
$$S_2 = \{\langle\{\phi \wedge \beta\}, \diamond\rangle, \langle\{\phi \wedge \neg\beta\}, \diamond\rangle\}$$

Condensed argument trees are complete argument trees. A condensed argument tree is obtained by replacing a set of siblings with a new set of siblings and then adding all the canonical undercuts appropriate for these new arguments.

Example 5.5.1 Let $\Delta_1 = \{\alpha \wedge \beta, \neg\alpha \wedge \gamma, \neg\alpha \wedge \delta, \neg\gamma \vee \neg\delta, \gamma, \delta\}$. Let $\Delta_2 = \{\alpha \wedge \beta, \neg\alpha \wedge \gamma \wedge \delta, \neg\gamma \vee \neg\delta, \gamma, \delta\}$.

$$\begin{array}{ccc}
 & T_1 & \\
 & \langle\{\alpha \wedge \beta\}, \alpha \wedge \beta\rangle & \\
\nearrow & & \nwarrow \\
\langle\{\neg\alpha \wedge \gamma\}, \diamond\rangle & & \langle\{\neg\alpha \wedge \delta\}, \diamond\rangle \\
\uparrow & & \uparrow \\
\langle\{\neg\gamma \vee \neg\delta, \delta\}, \diamond\rangle & & \langle\{\neg\gamma \vee \neg\delta, \gamma\}, \diamond\rangle \\
\uparrow & & \uparrow \\
\langle\{\gamma\}, \diamond\rangle & & \langle\{\delta\}, \diamond\rangle
\end{array}
\qquad
\begin{array}{c}
T_2 \\
\langle\{\alpha \wedge \beta\}, \alpha \wedge \beta\rangle \\
\uparrow \\
\langle\{\neg\alpha \wedge \gamma \wedge \delta\}, \diamond\rangle \\
\uparrow \\
\langle\{\neg\gamma \vee \neg\delta\}, \diamond\rangle \\
\uparrow \\
\langle\{\gamma, \delta\}, \diamond\rangle
\end{array}$$

T_2 is a condensation of T_1, but T_2 is not a compression of T_1. The reason is that $\{\neg\alpha \wedge \gamma \wedge \delta\} \not\subseteq \Delta_1$; hence, T_2 is not an argument tree for Δ_1.

Example 5.5.2 Let Δ_2 be as in example 5.5.1, and let $\Delta_3 = \Delta_2 \cup \{\neg\alpha \vee \neg\gamma, \neg\alpha \wedge \delta\}$:

$$\begin{tikzcd}[column sep=small]
 & T_3 & & T_2 \\
 & \langle\{\alpha\wedge\beta\}, \alpha\wedge\beta\rangle & & \langle\{\alpha\wedge\beta\}, \alpha\wedge\beta\rangle \\
\langle\{\neg\alpha\vee\neg\gamma, \gamma\}, \Diamond\rangle \arrow[ur] & \langle\{\neg\alpha\wedge\delta\}, \Diamond\rangle \arrow[u] & \langle\{\neg\alpha\wedge\gamma\wedge\delta\}, \Diamond\rangle \arrow[ul] & \langle\{\neg\alpha\wedge\gamma\wedge\delta\}, \Diamond\rangle \arrow[u] \\
\langle\{\neg\gamma\vee\neg\delta, \delta\}, \Diamond\rangle \arrow[u] & \langle\{\neg\gamma\vee\neg\delta, \gamma\}, \Diamond\rangle \arrow[u] & \langle\{\neg\gamma\vee\neg\delta\}, \Diamond\rangle \arrow[u] & \langle\{\neg\gamma\vee\neg\delta\}, \Diamond\rangle \arrow[u] \\
 & \langle\{\delta\}, \Diamond\rangle \arrow[u] & \langle\{\gamma, \delta\}, \Diamond\rangle \arrow[u] & \langle\{\gamma, \delta\}, \Diamond\rangle \arrow[u]
\end{tikzcd}$$

T_2 is not a condensation of T_3 due to

$$S_3 = \{\langle\{\neg\alpha\vee\neg\gamma, \gamma\}, \Diamond\rangle, \langle\{\neg\alpha\wedge\delta\}, \Diamond\rangle, \langle\{\neg\alpha\wedge\gamma\wedge\delta\}, \Diamond\rangle\}$$

and

$$S_2 = \{\langle\{\neg\alpha\wedge\gamma\wedge\delta\}, \Diamond\rangle\}$$

that make the condensation fail:

- $\gamma \notin \Delta_3 - \bigcup_{A\in S_3} \mathsf{Support}(A)$ because $\gamma \in \{\neg\alpha\vee\neg\gamma, \gamma\}$
- $\gamma \in \Delta_2 - \bigcup_{A\in S_2} \mathsf{Support}(A)$ because $\gamma \notin \bigcup_{A\in S_2} \mathsf{Support}(A)$

As with compression, condensation is not unique, and there are limits to condensation, for example, when an argument tree is a chain, and when all pairs of siblings have supports that are mutually contradictory.

Proposition 5.5.1 The "is a condensation of" relation is a partial ordering relation.

An interesting issue is the interplay between all the above rationalizing techniques. The first comment is that presumably all pruning should be considered before any compression or condensation. The question of the order between label-free and label-based pruning is not obvious. They seem largely independent, and maybe the order in which they are performed is unimportant. In particular, they do not alter anything in the remaining subtree, unlike compression and condensation. Turning to these two rationalizing techniques, one observation that seems in order is that compression preserves the original data whereas condensation moves to a new knowledgebase—a radical shift. Thus, it looks like all compression should rather take place before any condensation step, relieving us from the question of whether compression and condensation "commute" to define a confluence property à la Church-Rosser.

5.6 Shifting Commitment from Defenders to Attackers

For our fifth proposal for rationalizing argument trees, we consider shifting commitment from defenders to attackers. This is a form of rationalization of an argument tree where a defending argument may have a weaker support and an attacking argument may have a stronger support. By weakening a defending argument, we make it harder to attack it. Thus, a weakened defending argument means that there is less commitment by the defense (i.e., it is less difficult to put forward the defense case). This is because we weaken the knowledge in the knowledgebase for the defense. And as a complement, a strengthened attacking argument means that there is more commitment by the attackers (i.e., it is more difficult to put forward the attackers' case). Furthermore, if we consider the audience, then we can see that believability of the argument tree will be affected.

The justification for this definition of rationalization is that if the subject remains the same, and the support can be assumed, then it is reasonable to use the weaker argument and thereby avoid some undercuts.

Since shifting makes undercutting more difficult, this can help in applications that involve debate. If the defenders are weakened, then it is more difficult to find counterarguments.

Definition 5.6.1 Let T_1 and T_2 be argument trees from Δ_1 and Δ_2, respectively, where $\mathsf{Subject}(T_1) = \mathsf{Subject}(T_2)$. T_2 is a **shifting** of T_1 iff there is an injection (i.e., a one-to-one mapping) f from T_2 to T_1 such that

1. If A_2 is an undercut of A_1 in T_2, then $f(A_2)$ is an undercut of $f(A_1)$ in T_1 (Structural condition)
2. $\forall A \in \mathsf{Attackers}(T_2)$, $\mathsf{Support}(A) \vdash \mathsf{Support}(f(A))$ (Import condition)
3. $\forall A \in \mathsf{Defenders}(T_2)$, $\mathsf{Support}(f(A)) \vdash \mathsf{Support}(A)$ (Export condition)
4. $\forall A \in \mathsf{Attackers}(T_2)$, if $\mathsf{Support}(f(A)) \nvdash \mathsf{Support}(A)$, then $\mathsf{Degree}_C(A, \mathsf{Parent}(A)) \geq \mathsf{Degree}_C(f(A), \mathsf{Parent}(f(A)))$ (Relevancy condition)

We explain the conditions for shifting as follows: The structural condition ensures that each undercut in T_2 comes from a corresponding undercut in T_1 and so ensures that T_2 is a "substructure" of T_1; the import (respectively, export) condition ensures that each attacker in T_2 (respectively, defender in T_2) is inferentially stronger (respectively, weaker) than its corresponding argument in T_1; and the relevancy condition ensures that each attacker in T_2 that is not logically equivalent to its corresponding argument in T_1 has a higher or equal degree of undercut, and hence this ensures that no irrelevant assumptions are introduced.

Example 5.6.1 For T_1 and T_2 below, T_2 is a shifting of T_1:

$$\begin{array}{cc} T_1 & T_2 \\ \langle\{\alpha \wedge \beta\}, \alpha\rangle & \langle\{\alpha\}, \alpha\rangle \\ \uparrow & \uparrow \\ \langle\{\neg\alpha \vee \neg\beta\}, \Diamond\rangle & \langle\{\neg\alpha \vee \neg\beta, \beta\}, \Diamond\rangle \end{array}$$

Example 5.6.2 For T_1 and T_2 below, T_2 is a shifting of T_1:

$$\begin{array}{cc} T_1 & T_2 \\ \langle\{\alpha \wedge \beta\}, \alpha\rangle & \langle\{\alpha\}, \alpha\rangle \\ \uparrow & \uparrow \\ \langle\{\neg\alpha \vee \neg\beta\}, \Diamond\rangle & \langle\{\neg\alpha\}, \Diamond\rangle \end{array}$$

Example 5.6.3 For T_1 and T_2 below, T_2 is a shifting of T_1:

$$\begin{array}{cc} T_1 & T_2 \\ \langle\{\alpha \wedge \beta\}, \alpha\rangle & \langle\{\alpha\}, \alpha\rangle \\ \uparrow & \\ \langle\{\gamma, \gamma \rightarrow \neg\beta\}, \Diamond\rangle & \end{array}$$

Example 5.6.4 For T_1 and T_2 below, T_2 is not a shifting of T_1 because the irrelevant extra conjunct γ is added to the premise of the attacker, and so condition 4 of definition 5.6.1 is violated:

$$\begin{array}{cc} T_1 & T_2 \\ \langle\{\alpha \wedge \beta\}, \alpha\rangle & \langle\{\alpha \wedge \beta\}, \alpha\rangle \\ \uparrow & \uparrow \\ \langle\{\neg\beta\}, \Diamond\rangle & \langle\{\neg\beta \wedge \gamma\}, \Diamond\rangle \end{array}$$

A practical reasoning situation where an agent may wish to shift an argument tree is in the preparation of a political speech. By shifting, an argument tree may become more believable for the intended audience.

Furthermore, it may allow the politician to drop undercuts that are problematical. We return to the question of believability in the next chapter.

Another practical reasoning situation where shifting may be useful is in persuasion. If the defenders in an argument tree may be regarded as unpalatable for the intended audience, it may be better to weaken some of the premises of the defenders in order to make the argumentation more palatable.

Example 5.6.5 Consider the following argument used by a parent for the purpose of persuading a child to eat more healthy food, where b is "eat more broccoli," s is "eat more spinach," o is "eat more oranges," a is "eat more apples," and g is "grow up to be big and strong."

$$\langle\{b, s, o, a, b \wedge s \wedge o \wedge a \rightarrow g\}, g\rangle$$

Now we extend the original argument with an undercut, perhaps emanating from the child, as follows, where v is "like eating green vegetables":

$$\begin{array}{c}\langle\{b, s, o, a, b \wedge s \wedge o \wedge a \rightarrow g\}, g\rangle \\ \uparrow \\ \langle\{\neg v, \neg v \rightarrow \neg s \wedge \neg b\}, \diamond\rangle\end{array}$$

While the root argument seems to be a good argument from the point of view of realizing the goal g, if the intended recipient of the argument is likely to balk at the thought of eating more broccoli, spinach, oranges, and apples, as manifested by the undercut, then it may be better for the proponent to compromise and adopt a shift as follows:

$$\langle\{o, a, o \wedge a \rightarrow g\}, g\rangle$$

This rationalized argument is more difficult to undercut.

In the above example, we see that the shifted argument tree has lost the undercut. This is an illustration of a more general phenomenon that arises with shifting: By shifting, we may be able to change an initiating argument that is warranted to one that is unwarranted, or we may be able to change an initiating argument that is unwarranted to one that is warranted.

5.7 Discussion

Arguments can be compared on the basis of a range of intrinsic factors. In this chapter, we have considered how the tree structure can be analyzed to determine which arguments are defeated and which are undefeated, and we then used this to define how a root argument can be judged to be warranted. Next, we presented a framework for comparison based on degree of undercut. From analysis of intrinsic factors, the natural next step is rationalization of arguments.

Rationalizing argument trees is an important part of making argumentation a more manageable process for practical decision-support tasks. Even with a relatively modest knowledgebase, the number of arguments generated can be overwhelming. Therefore, it is imperative to adopt techniques, such as presented here, for editing the available information systematically.

Compression and condensation can allow undercuts to coalesce. This may be extended to handling first-order arguments. For example, if the root is a general statement that is being proposed, say $\forall x.(\alpha(x) \rightarrow \beta(x))$, then an undercut may be of the form $\alpha(a_1) \wedge \neg\beta(a_1)$. However, the degree of undercut is based on just one conflict. If there are many cases $a_1, \ldots, a_k$ such that for each $i \in \{1, \ldots, k\}$, we have $\alpha(a_i) \wedge \neg\beta(a_i)$; then these may coalesce to form a much stronger argument (i.e., an argument with a higher degree of undercut).

5.8 Bibliographic Notes

The definition for the judge function has been adopted from DeLP [GS04]. The notion of judging has been developed into a general framework for comparing and extending proposals for argumentation [Bow07]. The degree of undercut measure and rationalization based on pruning, compression, and condensation were first published in a paper by Besnard and Hunter [BH05]. Judgment criteria have been proposed for argument structures in [BH01], and this has been generalized into a framework for evaluating bipolar argumentation in [CL05a].

6 Considering the Audience

Whenever, for example, a speaker gives a talk, or an author writes an article, the intended audience of the talk or article is considered in advance of the proponent's delivering the talk or publishing the article. This is so that the proponent can tailor the presentation for the intended audience. For this, the proponent may have in mind some stereotypical member of the audience. We refer to a stereotypical member of the audience as a **representative agent**.

Effectively, a proponent has knowledge about such a representative agent to be used to decide how the proponent's argumentation is presented. The nature and source of knowledge about the representative agent varies according to the context. It may be detailed knowledge obtained via research of the audience (e.g., *when an advertising agency undertakes market research prior to developing an advertising campaign*), knowledge obtained by previous experience with the audience or with similar audiences (e.g., *when a university lecturer prepares a lecture for first-year undergraduates*), or knowledge obtained by reputation surrounding that audience (e.g., *when preparing a lecture for a job interview*), or it may just be presumptions, identified on the fly, about the audience (e.g., *when a customer makes a complaint to a shop assistant about a faulty product purchased in that shop*).

Obviously, there can be much uncertainty associated with the knowledge about an audience. This uncertainty may arise from incompleteness of knowledge about the audience, from misconceptions about the audience, and from heterogeneity within the audience. This last problem of heterogeneity is inevitable when the audience contains a number of agents. Moreover, trying to summarize, in the form of a stereotypical agent, knowledge about the group of agents in an audience will inevitably lead to some generalizations and simplifications of the knowledge about the individual agents in the audience.

Notwithstanding the potential difficulties of acquiring appropriate knowledge about the audience, we will explore how it can be captured and used for better practical argumentation. In this chapter, we will consider two approaches to representing and using such knowledge about the audience for extrinsic analysis: In section 6.1, we will consider "higher impact argumentation," and in section 6.2, we will consider "more convincing argumentation." For both these types of intrinsic factor, we will consider rationalization techniques.

6.1 Higher Impact Argumentation

Normally, any proponent of some arguments will want those arguments to have some impact on the intended audience: They will want the audience to notice the arguments and consider them in detail.

We see that arguments can have higher impact if they resonate with the audience. Furthermore, arguments resonate with the audience if they are related to the concerns of the audience. Consider a politician who wants to make a case for raising taxes. For example, arguments based on *increased taxes being needed to pay for improved health care* would resonate better with an audience of pensioners, whereas arguments based on *increased taxes being needed to pay for improved international transport infrastructure* would resonate better with an audience of business executives. The politician would select arguments depending on what audience is addressed.

To illustrate this idea further, consider reading an article in a current affairs magazine such as *The Economist* or *Newsweek*. Such an article has only a relatively small number of arguments. These arguments are quite a small subset of all possible arguments that either the writer or the reader could construct. In producing an article, a writer regards some arguments as having higher impact with the intended audience than others and prefers to use only the higher impact ones. In a sense, there is some filter on the arguments that are presented. This renders the argumentation more apposite for the intended audience. The need for such appositeness is reflected in diverse professions such as law, medicine, science, advertising, management, . . . , and of course in every-day life.

However, impact is not only a matter of addressing issues relevant to the intended audience; it also is about being comprehensible to the intended audience. The better the audience grasps an argument, the higher the argument's impact is. However, comprehending an argument involves a cost. Arguments that are more difficult to understand cost the audience more intellectual effort.

Aiming for higher impact argumentation raises two key questions: (1) What is impact here? (in other words, what does it mean and how do we calculate it in practice?) and (2) given an estimate of impact, how do we use it to be selective in constructing argument trees? In this section, we look at some of the issues surrounding these questions.

6.1.1 Resonance of Arguments

The impact of argumentation depends on what the audience regards as important. We adopt a simple way, called a "concernbase," of capturing what a representative agent for the intended audience thinks is important and then use this to rate how arguments resonate with this representative agent.

Definition 6.1.1 A **concernbase** is a map η from $\mathscr{L}$ to $[0,1]$. Each formula $\alpha \in \mathscr{L}$ such that $\eta(\alpha) > 0$ is a **concern**, which is denoted $\alpha \in \Theta_\eta$. If for all $\alpha \in \mathscr{L}$, $\eta(\alpha) \in \{0,1\}$, then η is called a **binary concernbase**.

Intuitively, $\eta(\alpha) < \eta(\beta)$ means that β is a *more important concern* than α.

A concern may represent something that an agent would like to be true in the world (i.e., a desideratum for the agent). There is no constraint that the concern has to be something that the agent can actually make true. It may be something unattainable, such as *there are no more wars* or *bread and milk are free for everyone*. Alternatively, a concern may represent something that an agent would like to be false in the world (i.e., the negation of a desideratum). For example, it may be *there are many street robberies of mobile phones*. However, a concern can also be something that the agent is neutral about but nonetheless regards as important. For example, in a World Cup football final, involving, say, Brazil and Italy, an agent may not have a preference for either team but may nonetheless be concerned by each of the outcomes (i.e., *the outcome that Brazil wins* or *the outcome that Italy wins*).

There is also no constraint that the concernbase for an agent be consistent (either explicitly or implicitly). An explicit inconsistency arises when $\Theta_\eta \vdash \bot$ holds. An illustration of an implicit inconsistency is given in the following example.

Example 6.1.1 Suppose an agent has the following desiderata as concerns.

p The government lowers tax revenue.

q The government does not make any spending cutbacks.

Implicitly this can be regarded as an inconsistency, unless lowering taxes is taken to be compatible with the absence of any cutbacks.

The concerns of an agent are not necessarily the same as the beliefs of an agent. For example, an agent may believe that *Ronaldo is a football player* while the agent does not care at all about that. In contrast, the agent may be concerned whether *he has won the lottery for which he has bought a ticket*, but he may believe that *he has not won the lottery*.

Note that a concern is not restricted to being an atom or a literal but can be a more complex formula.

Example 6.1.2 Consider an audience of disabled people who are concerned about the streetcar (tram) system in use in the city. They cannot access platforms because of stairs, and even if they could do so, this would not help them to get to their destination anyway, because current carriages are too narrow (both doors and the interior). Obviously, for such an audience, a concern is $p \wedge s$ (platforms are adapted, and streetcars (trams) are reconditioned), while neither p nor s is. In symbols, $\eta(p \wedge s) > 0$ but $\eta(p) = 0$ and $\eta(s) = 0$. That is, $p \wedge s \in \Theta_\eta$, but $p \notin \Theta_\eta$ and $s \notin \Theta_\eta$. The intuition here is that an agent may feel the conjunction $p \wedge s$ to be important while not caring at all about either conjunct alone.

There is no requirement in definition 6.1.1 that the importance of a concern α be related to the importance of any formula β that entails α: It is not the case in general that $\beta \vdash \alpha$ implies $\eta(\alpha) \leq \eta(\beta)$. Moreover, it is not the case in general that two logically equivalent formulae α and β are such that $\eta(\alpha) = \eta(\beta)$.

Example 6.1.3 Consider a family that has a history of lung cancer that has not been caused by smoking. For this audience, the following may be two concerns:

p having lung cancer

q smoking and having lung cancer

Furthermore, it would be reasonable for the weighting to be such that $\eta(p) > \eta(q)$, despite the fact that $q \vdash p$ holds.

Definition 6.1.1 is very liberal. As an example, it is possible to have η such that for some α, not only is α a concern but so is $\neg\alpha$. In symbols, $\alpha \in \Theta_\eta$ and $\neg\alpha \in \Theta_\eta$. As another example, it is possible to have η such that for some α and β, $\alpha \wedge \beta$ is a concern while $\beta \wedge \alpha$ is not. In symbols, $\alpha \wedge \beta \in \Theta_\eta$ and $\beta \wedge \alpha \notin \Theta_\eta$. Such concernbases are not ill-fated, but some care must be exercised when using one to assess the resonance of an argument (see the discussion in section 6.1.3 on the sales and income tax examples).

We now turn to the question of how a concern contributes to the impact of an argument. It is not enough that the concern simply occurs in the argument: This is a feature of rhetorics (e.g., politicians often mention proposals hypothetically, thereby giving the impression that they endorse these, whereas they never actually say so because they do not, and do not wish to, endorse the proposals). More naturally, concerns contribute to the impact of an argument if they are asserted, or, equally, refuted, by the argument, in which case these concerns are "echoed" by the argument.

Example 6.1.4 Consider an audience of pensioners living in the city. Assume they worry about the local level of noise pollution. A concern for the audience would then be formalized by p, denoting "the level of noise pollution is to be regulated." Hence, an argument asserting that the level of noise pollution is not to be regulated would resonate strongly with the audience—although not in a favorable way. As a formal example, $\langle\{q \wedge \neg p\}, \neg p\rangle$ would resonate strongly, but unfavorably, whatever q.

Thus, the impact of an argument is based on the set of concerns that the argument asserts or refutes (through its support). Such a set is accordingly characterized in the next definition.

Definition 6.1.2 The **echo** for an argument A with respect to a concernbase η is as follows:

$$\mathsf{Echo}(A, \eta) = \{\delta \in \Theta_\eta \mid \{\delta, \neg\delta\} \cap \mathsf{Cn}(\mathsf{Support}(A)) \neq \emptyset\}.$$

This means that $\delta \in \Theta_\eta$ is in the echo of $\langle\Phi, \alpha\rangle$ iff either $\Phi \vdash \delta$ or $\Phi \vdash \neg\delta$.

Example 6.1.5 Back to our audience of pensioners (so, $\Theta_\eta = \{p\}$ again), here is the echo for a few arguments:

$$\mathsf{Echo}(\langle\{p, q\}, \alpha\rangle, \eta) = \{p\}$$
$$\mathsf{Echo}(\langle\{\neg p, q\}, \beta\rangle, \eta) = \{p\}$$
$$\mathsf{Echo}(\langle\{p \vee q\}, \gamma\rangle, \eta) = \emptyset$$

In the third case, the pensioners get no echo from the argument because it takes no definite stand with regard to noise pollution (it only frames regulation of the level of noise pollution as an alternative to another proposition that is not among the audience's concerns).

There are various ways that we may use the echo of an argument to define the resonance of the argument. An obvious choice is the following weighted sum of the echo.

Definition 6.1.3 The **resonance as a weighted sum of the echo** is represented by the $\mathsf{Resonance}_w$ function as follows:

$$\mathsf{Resonance}_w(A, \eta) = \sum_{\alpha \in \mathsf{Echo}(A, \eta)} \eta(\alpha)$$

Example 6.1.6 Let $\Theta_\eta = \{\beta, \gamma\}$, where $\eta(\beta) = 1$ and $\eta(\gamma) = 0.5$. Let A_1 and A_2 be arguments as below, where A_2 is a canonical undercut for A_1:

$$A_1 = \langle\{\beta, \beta \to \alpha\}, \alpha\rangle$$

$$A_2 = \langle\{\neg(\beta \vee \gamma)\}, \neg(\beta \wedge (\beta \to \alpha))\rangle$$

Hence we get the following resonance values:

$$\mathsf{Resonance}_w(A_1, \eta) = 1.0$$

$$\mathsf{Resonance}_w(A_2, \eta) = 1.5$$

Thus, resonance of an argument, calculated as the weighted sum of the echo of the argument, is a simple and intuitive way of assessing an argument. We return to the question of what constitutes a good resonance function in section 6.1.3.

6.1.2 Analyzing Resonance of Arguments and Counterarguments

Now we want to consider resonance in the context of an argument tree. For this, we will use the following naive definition that is the sum of the resonance of the arguments in the tree.

Definition 6.1.4 The **resonance of the argument tree** T with respect to a concernbase η is as follows:

$$\mathsf{Resonance}_w(T, \eta) = \sum_{A \in \mathsf{Nodes}(T)} \mathsf{Resonance}_w(A, \eta)$$

When considering impact, we may also want to take the cost of the arguments into account. Here we will consider a very simple example of a measure of cost, in which we are assuming that each propositional symbol used in the support of an argument is of unit cost, and then in section 6.3, we return to the issue of measuring the cost of an argument.

Definition 6.1.5 For an argument A, $\mathsf{Cost}_s(A) = |\mathsf{Atoms}(\mathsf{Support}(A))|$.

The measure of cost given by Cost_s is $\log_2$ of the number of classical interpretations for the support of the argument. This gives an indication of the effort required by the intended audience to consider the argument.

Example 6.1.7 For the following argument A, we have $\mathsf{Cost}_s(A) = 4$:

$\langle \{\alpha \wedge \neg\delta, \beta \vee \delta, \alpha \wedge \beta \rightarrow \gamma\}, \gamma \vee \delta \rangle$

We aggregate the cost of each argument in an argument tree using the following notion of tree cost.

Definition 6.1.6 For an argument tree T, $\mathsf{TreeCost}_s(T) = \sum_{A \in \mathsf{Nodes}(T)} \mathsf{Cost}_s(A)$.

Using the definitions for the resonance of an argument tree and for the cost of an argument tree gives a two-dimensional analysis of impact for an argument tree. This analysis of impact can be used to identify, and perhaps prune, arguments that have little or no resonance and/or cost too much relative to their contribution to the resonance of the tree.

Now we consider an extended example in order to illustrate how impact can be improved for intended audiences. The example is based on an idea by a pressure group in the United Kingdom that wants the government to turn railway lines into dedicated roads for express buses and coaches: Each railway line would become a busway, and trains would be replaced by buses.

Example 6.1.8 Consider the proposition letters in figure 6.1. We use these propositional letters to give the following set of formulae Δ:

$$\left\{ \begin{array}{ccc} p_2 \wedge p_{10} \wedge p_{12} \wedge p_{14} \wedge p_{21} \wedge p_{24} \rightarrow p_1, & p_{20} \rightarrow \neg p_{11}, & p_{13} \rightarrow \neg p_6, \\ p_5 \wedge p_6 \wedge p_7 \rightarrow \neg p_1, & p_{22} \rightarrow p_{14}, & p_{19} \rightarrow \neg p_{21}, \\ p_3 \wedge p_9 \wedge p_{23} \rightarrow p_2, & p_{17} \rightarrow \neg p_{14}, & p_{15} \rightarrow \neg p_6, \\ p_{11} \rightarrow \neg p_4, & p_{22} \rightarrow p_{17}, & p_4 \rightarrow \neg p_{24}, \\ p_8 \rightarrow \neg p_5, & p_{18} \rightarrow p_{21}, & p_{16} \rightarrow \neg p_7 \end{array} \right\}$$

Let Δ be the union of the above set of implicational formulae together with the following set of propositions:

$\{p_3, p_4, p_5, p_6, p_7, p_8, p_9, p_{10}, p_{11}, p_{12}, p_{13}, p_{15}, p_{16}, p_{18}, p_{19}, p_{20}, p_{22}, p_{23}, p_{24}\}$

p_1	The government should turn railways into busways.
p_2	Busways require less government subsidy than railways.
p_3	Busways are cheaper to maintain than railways.
p_4	Buses are more polluting than trains.
p_5	Buses are less reliable than trains.
p_6	Buses are less popular than trains.
p_7	Buses are less comfortable than trains.
p_8	Busways are exclusively for scheduled buses.
p_9	Buses are cheaper to buy than trains.
p_{10}	It is cheap to convert a railway line into a busway.
p_{11}	Trains require electricity and so are just as polluting as buses.
p_{12}	Bus companies are more open to competition than train companies.
p_{13}	Bus tickets will be cheaper than the current train tickets.
p_{14}	There will be more variety of services.
p_{15}	There will be very cheap services for young people.
p_{16}	There will be luxury services for business executives.
p_{17}	There will be fewer services late at night and early in the morning.
p_{18}	Busways can have buses close together and so have more capacity than trains.
p_{19}	Stations cannot handle more passengers when converted for buses.
p_{20}	Pollution from buses is more difficult to manage than from power stations.
p_{21}	Busways can carry more people than railways.
p_{22}	There will be deregulation of services.
p_{23}	There will be investment opportunities for private enterprises.
p_{24}	Busways are a good environmental choice.

Figure 6.1
Proposition letters $p_1, \ldots, p_{24}$ for example 6.1.8 concerning proposals for turning railways into busways.

Let $\Phi \subset \Delta$ be the following set of formulae:

$$p_2 \wedge p_{10} \wedge p_{12} \wedge p_{14} \wedge p_{21} \wedge p_{24} \rightarrow p_1$$

$$p_3 \wedge p_9 \wedge p_{23} \rightarrow p_2$$

$$p_{22} \rightarrow p_{14}$$

$$p_{18} \rightarrow p_{21}$$

$$p_3, p_9, p_{10}, p_{12}, p_{18}, p_{22}, p_{23}, p_{24}$$

Using Δ, we can construct the following arguments:

$A_1 \quad \langle \Phi, p_1 \rangle$

$A_2 \quad \langle \{p_5, p_6, p_7, p_5 \wedge p_6 \wedge p_7 \rightarrow \neg p_1\}, \Diamond \rangle$

$A_3 \quad \langle \{p_{19}, p_{19} \rightarrow \neg p_{21}\}, \Diamond \rangle$

$A_4 \quad \langle\{p_{22}, p_{22} \to p_{17}, p_{17} \to \neg p_{14}\}, \diamond\rangle$

$A_5 \quad \langle\{p_4, p_4 \to \neg p_{24}\}, \diamond\rangle$

$A_6 \quad \langle\{p_{11}, p_{11} \to \neg p_4\}, \diamond\rangle$

$A_7 \quad \langle\{p_{20}, p_{20} \to \neg p_{11}\}, \diamond\rangle$

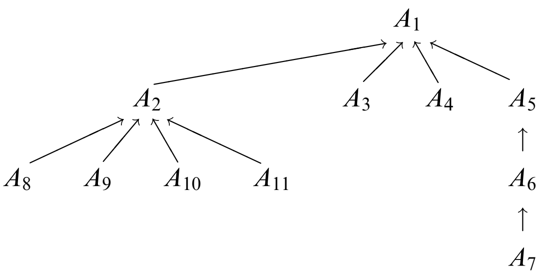
$A_8 \quad \langle\{p_8, p_8 \to \neg p_5\}, \diamond\rangle$

$A_9 \quad \langle\{p_{13}, p_{13} \to \neg p_6\}, \diamond\rangle$

$A_{10} \quad \langle\{p_{16}, p_{16} \to \neg p_7\}, \diamond\rangle$

$A_{11} \quad \langle\{p_{15}, p_{15} \to \neg p_6\}, \diamond\rangle$

These lead to the following argument tree, T_1. Let us suppose that T_1 represents all the key publicly available information on the proposal for turning railways into busways:



Clearly T_1 contains quite a lot of information, and not all of it will relate to the concerns of any particular audience.

Let us now consider an advocate for the project speaking to an audience of pensioners who live in London. We may suppose that their concerns can be captured by the following:

$\{p_5 \vee p_{13}, p_{24}\}$

Suppose we consider a binary concernbase; then, we obtain the following nonempty echo evaluations (i.e., we only consider the arguments with nonempty echo):

$\mathsf{Echo}(A_1, \eta) = \{p_{24}\}$

$\mathsf{Echo}(A_2, \eta) = \{p_5 \vee p_{13}\}$

$\mathsf{Echo}(A_5, \eta) = \{p_{24}\}$

$\mathsf{Echo}(A_8, \eta) = \{p_5 \vee p_{13}\}$

$\mathsf{Echo}(A_9, \eta) = \{p_5 \vee p_{13}\}$

On this basis, the following argument tree would be better than the original, since it has the same resonance as the original but lower cost if we use a measure of cost such as Cost_s (see definition 6.1.5):

$$\begin{tikzcd}
& & A_1 & \\
& A_2 \arrow[ur] & & A_5 \arrow[ul] \\
A_8 \arrow[ur] & A_9 \arrow[u] & &
\end{tikzcd}$$

Let us now consider an advocate for the project speaking to a group of business executives who may wish to be involved in the project as investors. We may suppose that their concerns can be captured by the following:

$\{p_6, p_{12}, p_{14}, p_{22}, p_{23}\}$

Assuming a binary concernbase, then we obtain the following nonempty echo evaluations (i.e., we only consider the arguments with nonempty echo):

$\mathsf{Echo}(A_1, \eta) = \{p_{12}, p_{14}, p_{22}, p_{23}\}$

$\mathsf{Echo}(A_2, \eta) = \{p_6\}$

$\mathsf{Echo}(A_4, \eta) = \{p_{14}, p_{22}\}$

$\mathsf{Echo}(A_9, \eta) = \{p_6\}$

$\mathsf{Echo}(A_{11}, \eta) = \{p_6\}$

On this basis, the following argument tree would be better than the original, since it has the same resonance as the original but lower cost if we use a measure of cost such as Cost_s:

$$\begin{tikzcd}
& & A_1 & \\
& A_2 \arrow[ur] & & A_4 \arrow[ul] \\
A_9 \arrow[ur] & A_{11} \arrow[u] & &
\end{tikzcd}$$

Finally, let us consider a protestor against the project who is speaking to a group of environmentalists. We may suppose that their concerns can be captured by the following:

$\{p_{20}, p_{21}, p_{24}\}$

Assuming a binary concernbase, then we obtain the following nonempty echo evaluations (i.e., we only consider the arguments with nonempty echo):

$$\text{Echo}(A_1, \eta) = \{p_{21}, p_{24}\}$$

$$\text{Echo}(A_3, \eta) = \{p_{21}\}$$

$$\text{Echo}(A_5, \eta) = \{p_{24}\}$$

$$\text{Echo}(A_7, \eta) = \{p_{20}\}$$

On this basis, the following argument tree would be better than the original, since it has the same resonance as the original but lower cost if we use a measure of cost such as Cost_s:

$$\begin{tikzcd}
& A_1 & \\
A_3 \arrow[ur] & & A_5 \arrow[ul] \\
& & A_6 \arrow[u] \\
& & A_7 \arrow[u]
\end{tikzcd}$$

The above example illustrates how we may improve the impact of argumentation, given a complete argument tree, and a concernbase, by selecting a subtree of an argument tree that has a good trade-off for resonance versus cost.

6.1.3 Defining Concernbases and Resonance Functions

We conclude our coverage of analyzing impact by discussing some issues with regard to the definitions for concernbases and for resonance functions.

We start by considering a potentially problematic phenomenon arising from definitions 6.1.1 and 6.1.2: Imagine that the audience is concerned that *the income tax is raised* and is also concerned that *the sales tax is raised*. In symbols, $\alpha \in \Theta_\eta$ and $\beta \in \Theta_\eta$. Unless $\alpha \vee \beta \in \Theta_\eta$ is explicitly specified, the echo for an argument whose support proves $\alpha \vee \beta$ (but neither α nor β) would be void—which is counterintuitive because the proposition that *either the sales tax or the income tax is raised* presumably resonates with the audience. Thus, we reiterate the recommendation that care needs to be taken with definition 6.1.1 to ensure the concernbase is defined appropriately for each application.

Definition 6.1.2 is purely logical and the overall approach could be criticized on pragmatic grounds as follows. Impact naturally occurs when the audience realizes that the argument asserts (or refutes) a concern. For a human audience, this may fail to happen if the corresponding inference is too complex: If it would take dozens of reasoning steps to follow how the argument asserts the concern, then humans are unlikely to notice that the concern is met. The case is even stronger when the concern only comes as an auxiliary assertion in the argument but is not the main claim of the argument (when the concern is the consequent of the argument, the audience is explicitly told that the concern is met, and that is presumably enough to make the audience realize that). At least in part, we address these issues when we take the cost of understanding into account (see section 6.3).

If η_1 and η_2 are concernbases such that $\mathsf{Cn}(\Theta_{\eta_1}) \subseteq \mathsf{Cn}(\Theta_{\eta_2})$, and A is an argument, then it is not necessarily the case that $\mathsf{Echo}(A, \eta_1) \subseteq \mathsf{Echo}(A, \eta_2)$, as illustrated in the next example.

Example 6.1.9 Let $\Theta_{\eta_1} = \{\alpha, \beta \wedge \gamma\}$ and $\Theta_{\eta_2} = \{\alpha \wedge \beta, \gamma\}$. Consider $A = \langle\{\alpha, \gamma\}, \alpha\rangle$. So, $\mathsf{Echo}(A, \eta_1) = \{\alpha\}$ and $\mathsf{Echo}(A, \eta_2) = \{\gamma\}$. That is, $\mathsf{Echo}(A, \eta_1) \not\subseteq \mathsf{Echo}(A, \eta_2)$.

The property invalidated by example 6.1.9 holds in the following weaker form.

Proposition 6.1.1 If $\mathsf{Cn}(\mathsf{Support}(A_1)) \subseteq \mathsf{Cn}(\mathsf{Support}(A_2))$ and $\Theta_{\eta_1} \subseteq \Theta_{\eta_2}$, then $\mathsf{Echo}(A_1, \eta_1) \subseteq \mathsf{Echo}(A_2, \eta_2)$.

The kernel of a function with codomain $[0, 1]$ is the set of all elements in the domain that have the value 0 (i.e., $\ker(\eta) = \{\alpha \in \mathcal{L} \mid \eta(\alpha) = 0\}$). Thus, the kernel of the concernbase function consists of all formulae that are not among the agent's concerns.

Corollary 6.1.1 If $\mathsf{Cn}(\mathsf{Support}(A_1)) \subseteq \mathsf{Cn}(\mathsf{Support}(A_2))$ and $\ker(\eta_2) \subseteq \ker(\eta_1)$, then $\mathsf{Echo}(A_1, \eta_1) \subseteq \mathsf{Echo}(A_2, \eta_2)$.

We now turn to a general definition for the resonance of an argument, which is meant to reflect (in general terms) the importance that the agent attaches to the argument. The definition for resonance as the weighted sum of the echo (definition 6.1.3) is an instance of this general definition.

Definition 6.1.7 Let Ω be the set of all arguments, and let Υ be the set of all concernbases. A measure of **resonance** of arguments with respect to concernbases is a function $\mathsf{Resonance} : (\Omega \times \Upsilon) \to [0, 1]$ such that

1. If $\mathsf{Echo}(A_1, \eta_1) \subseteq \mathsf{Echo}(A_2, \eta_2)$, then $\mathsf{Resonance}(A_1, \eta_1) \leq \mathsf{Resonance}(A_2, \eta_2)$.

2. $\text{Resonance}(A, \eta) = 0$ iff $\text{Echo}(A, \eta) = \emptyset$.

Given η, a resonating argument A is such that $\text{Resonance}(A, \eta) \neq 0$ (equivalently, $\text{Echo}(A, \eta) \neq \emptyset$). Stated otherwise, a resonating argument is such that its support implies either δ or $\neg\delta$ for some $\delta \in \Theta_\eta$.

Proposition 6.1.2 If $\text{Cn}(\text{Support}(A_1)) \subseteq \text{Cn}(\text{Support}(A_2))$ and $\Theta_{\eta_1} \subseteq \Theta_{\eta_2}$, then $\text{Resonance}(A_1, \eta_1) \leq \text{Resonance}(A_2, \eta_2)$.

An obvious candidate for resonance is a weighted sum of the echo, which we formalized earlier (definition 6.1.3).

The next example is a counterexample to a potential generalization of proposition 6.1.2 for resonance as weighted sum: If η_1 and η_2 are such that $\text{Cn}(\Theta_{\eta_1}) \subseteq \text{Cn}(\Theta_{\eta_2})$, then it is not necessarily the case that $\text{Resonance}_w(A, \eta_1) \leq \text{Resonance}_w(A, \eta_2)$.

Example 6.1.10 Let $\Theta_{\eta_1} = \{\alpha, \beta\}$ and $\Theta_{\eta_2} = \{\alpha \wedge \beta, \gamma\}$. Thus, $\text{Cn}(\Theta_{\eta_1}) \subseteq \text{Cn}(\Theta_{\eta_2})$. Take A to be the argument $\langle\{\beta\}, \beta\rangle$. Thus, $\text{Echo}(A, \eta_1) = \{\beta\}$ and $\text{Echo}(A, \eta_2) = \emptyset$. Hence, $\text{Resonance}_w(A, \eta_2) < \text{Resonance}_w(A, \eta_1)$.

There are various ways to regard a concernbase as being weaker than another. One way, given in the following proposition, implies that the weaker concernbase leads to a higher resonance for an argument.

Proposition 6.1.3 If there is an injection f from Θ_{η_1} to Θ_{η_2} such that $\{\delta\} \vdash f(\delta)$ and $\eta_1(\delta) \leq \eta_2(f(\delta))$ for all $\delta \in \Theta_{\eta_1}$, then $\text{Resonance}_w(A, \eta_1) \leq \text{Resonance}_w(A, \eta_2)$.

Increasing the strength of the weighting in a concernbase also increases resonance as shown in the next result.

Corollary 6.1.2 If $\eta_1 \leq \eta_2$, then $\text{Resonance}_w(A, \eta_1) \leq \text{Resonance}_w(A, \eta_2)$.

These observations support the case that Resonance_w is a simple but intuitive choice for a definition for a resonance function—though there are numerous other possibilities for the definition of Resonance. Nothing dogmatic dictates the choice of a particular Resonance function, and matters such as the characteristics and needs of a particular application should prevail.

In general, care should be taken if η and Resonance are to work well together. Indeed, consider an audience who are concerned about a proposal that a new stadium is to be built in their town. Of course, the most immediate way to capture this is with η_1 such that $\Theta_{\eta_1} = \{s\}$ (for the sake of simplicity, the audience is assumed not to have other concerns).

Surely, the concern of the audience is as well captured by η_2 such that $\Theta_{\eta_2} = \{\neg s\}$ or by η_3 such that $\Theta_{\eta_3} = \{s, \neg s\}$. Unfortunately, definition 6.1.7 does not preclude that the resonance of an argument in favor of building the new stadium be strictly stronger with respect to η_3 than with respect to η_1 (or η_2), which is a counterintuitive situation.

A related abstract case is with $\Theta_{\eta_1} = \{\alpha \wedge \beta\}$ and $\Theta_{\eta_2} = \{\alpha \wedge \beta, \beta \wedge \alpha\}$. Again, definition 6.1.7 does not preclude that the resonance of an argument whose support proves $\alpha \wedge \beta$ be strictly stronger with respect to η_2 than with respect to η_1.

In general, one should refrain from comparing resonance values issued from two different concernbases. In other words, it is better to avoid comparing how a given argument impacts different audiences. It is less problematic to compare how various arguments impact a given audience, but there are some difficult cases: Let $\eta(\alpha \wedge \beta \wedge \gamma) = \eta(\alpha \wedge \beta \wedge \delta)$ and $\delta \in \Theta_\eta$, but Θ_η contains none of $\alpha, \beta, \gamma, \alpha, \alpha \wedge \beta, \beta \wedge \gamma, \ldots$ Definition 6.1.7 tolerates the resonance of an argument proving $\alpha \wedge \beta \wedge \delta$ being strictly higher than the resonance of an argument proving $\alpha \wedge \beta \wedge \gamma$ (in a sense, resonance with respect to δ is counted twice).

6.2 More Convincing Argumentation

The goal of convincing the intended audience is often of paramount importance to a proponent when undertaking argumentation. This is normally the case among many professionals such as politicians, journalists, clinicians, scientists, administrators, and lawyers. In order to carry out many of their professional tasks, they must take an intended audience into account in attempting to assess how convincing the arguments may appear to a typical member of that intended audience.

Consider a politician who is giving a speech on a plan by the government to charge car drivers to be able to drive into the city. If the audience is a group of commuters who live in the city, then the politician would want to provide arguments that relate to what the audience is likely to believe—perhaps start by admitting that *new buses are much needed* and then continue by arguing that, *for the money for buying buses to be raised, charging car drivers is necessary*. In other words, the politician would use a purported belief of the intended audience (new buses are much needed) in order to make a claim in favor of the new tax.

In contrast, if the audience is a group of business executives, then the politician may start by saying that *companies lose money with their vehicles being caught in traffic jams in the city*, and the politician may later argue that *the cost of the charge to delivery trucks, for instance, would be more than offset by the savings made by these trucks no longer being stuck in traffic*.

The beliefs of these two audiences are unlikely to be the same. The business executives, for example, may live in a different city, and so they might not know whether or not there are enough buses or whether there is a lot of pollution in that city. And the commuters may be unaware that businesses have greater expenses to pay when their delivery vehicles are stuck in traffic. The beliefs of these two audiences may even be mutually contradictory. Therefore, the way the politician has to proceed is to be selective so that the arguments used are likely to be based on assumptions that are already believed by the audience and, if not, on assumptions that do not contradict the intended audience's beliefs.

This need to consider whether an argument is convincing is obviously reflected in everyday life as well. Consider, for example, that some friends, Jan, Fred, and Pierre, are discussing a plan to hold a barbecue on the weekend. Jan may put forward the argument that *it is a good time of year to hold it, and therefore they should proceed with arranging the barbecue*. Fred may say that *the astrological signs are not auspicious for a barbecue this weekend, and therefore they should not proceed with arranging it*. If Jan does not believe in astrology, then he is unlikely to believe in this counterargument. In contrast, Pierre may say that *the weather reports predict that there will be a storm over the weekend, and so they should not proceed with arranging it*.

If Jan believes in weather reports, then he may well believe this second counterargument, and as a result he may decide that his original argument is defeated. In both cases, Jan is the audience of the arguments by Fred and Pierre, and Jan is only convinced by the arguments that he finds believable.

From these examples, we see that one of the key requisites for making an argument convincing is to make it believable. In this section, we consider how we can capture the beliefs of a stereotypical member of the audience and use those beliefs to determine whether a given argument is believable and, thereby, in a sense, convincing. There are a number of additional dimensions for assessing how convincing an argument is, such as the authority and reliability of the proponent, but we will not consider these further in this chapter.

6.2.1 Empathy and Antipathy

Thus, to assess how convincing an argument is, we use a "beliefbase" that reflects the beliefs held by a typical member of the intended audience of the argument. We will assume that the beliefs of a stereotypical member of the audience are consistent. Argumentation for different intended audiences requires different beliefbases.

Definition 6.2.1 A **beliefbase** is a set of formulae $\Lambda \subseteq \mathscr{L}$ such that $\Lambda \nvdash \bot$.

Intuitively, the more an argument agrees with the audience's beliefbase, the more convincing the argument is. And, the more an argument disagrees with the audience's beliefbase, the less convincing the argument is. Thus, we regard this agreement/disagreement as one of the key dimensions of determining how convincing an argument is.

Technically, the audience's beliefbase is compared with the support of an argument, which amounts to comparing a pair of theories. To formalize this, we use the following definitions for antipathy and empathy.

The definition for the empathy for an argument is based on counting models. For this, we require the following definition for the degree of entailment. Given two sets of formulae Ψ and Φ, the degree of entailment of Ψ for Φ is the proportion of models of Ψ that are models of Φ. It captures, in a sense, how much Ψ implies $\bigwedge \Phi$.

Definition 6.2.2 Let $\Phi \subseteq \mathscr{L}$ and $\Psi \subseteq \mathscr{L}$ be such that $\Psi \nvdash \bot$. The **degree of entailment** of Ψ for Φ is as follows (where Models is defined in C.2.7):

$$\mathsf{Entailment}(\Psi, \Phi) = \frac{|\mathsf{Models}(\Phi \cup \Psi, \mathsf{Atoms}(\Phi \cup \Psi))|}{|\mathsf{Models}(\Psi, \mathsf{Atoms}(\Phi \cup \Psi))|}$$

Example 6.2.1 Parentheses are omitted for the sake of clarity:

$\mathsf{Entailment}(\alpha, \alpha \wedge \beta) = 1/2$

$\mathsf{Entailment}(\alpha, \alpha \wedge \beta \wedge \gamma) = 1/4$

$\mathsf{Entailment}(\alpha, \alpha \wedge \beta \wedge \gamma \wedge \delta) = 1/8$

$\mathsf{Entailment}(\alpha \wedge \beta, \alpha \vee \beta) = 1$

$\mathsf{Entailment}(\alpha \wedge \beta, \alpha \wedge \varepsilon) = 1/2$

$\mathsf{Entailment}(\alpha \wedge \beta \wedge \gamma, \alpha \wedge \varepsilon) = 1/2$

$\mathsf{Entailment}(\alpha \wedge \beta \wedge \gamma \wedge \delta, \alpha \wedge \varepsilon) = 1/2$

$\mathsf{Entailment}(\alpha \wedge \varepsilon, \alpha \wedge \beta \wedge \gamma \wedge \delta) = 1/8$

$\mathsf{Entailment}(\alpha \wedge \beta, \alpha \wedge \neg\beta) = 0$

In general, $\mathsf{Entailment}(\Phi, \Psi) \neq \mathsf{Entailment}(\Psi, \Phi)$.

Proposition 6.2.1 Let $\Phi \subseteq \mathscr{L}$ and $\Psi \subseteq \mathscr{L}$ be such that $\Phi \nvdash \bot$ and $\Psi \nvdash \bot$:

1. $0 \leq \mathsf{Entailment}(\Psi, \Phi) \leq 1$
2. $\mathsf{Entailment}(\Psi, \Phi) = 1$ iff for all $\varphi \in \Phi$, $\Psi \vdash \varphi$
3. If $\mathsf{Entailment}(\Psi, \Phi) = 1$ then $\mathsf{Entailment}(\Phi, \Psi) > 0$
4. $\mathsf{Entailment}(\Psi, \Phi) = 0$ iff $\Psi \cup \Phi \vdash \perp$
5. $\mathsf{Entailment}(\Psi, \Phi) = 0$ iff $\mathsf{Entailment}(\Phi, \Psi) = 0$

We can use the degree of entailment as a measure of the degree of empathy: Essentially, the audience with beliefs Λ will highly empathize with an argument A if the degree of entailment of Λ for the support of A is high.

Definition 6.2.3 The degree of **empathy** is represented by the Empathy function defined below:

$\mathsf{Empathy}(A, \Lambda) = \mathsf{Entailment}(\Lambda, \mathsf{Support}(A))$

The definition for the antipathy for an argument is specified using Dalal distance, which was introduced in chapter 5 (definition 5.2.3). Essentially, the antipathy for an argument is the minimum Hamming distance (i.e., the Dalal distance) between the nearest models of the beliefbase and the support of the argument.

Definition 6.2.4 The degree of **antipathy** is represented by the Antipathy function defined below:

$\mathsf{Antipathy}(A, \Lambda) = \mathsf{Conflict}(\mathsf{Support}(A), \Lambda, \mathsf{Atoms}(\mathsf{Support}(A) \cup \Lambda))$

Example 6.2.2 Let $\Delta = \{\alpha \vee \beta, \alpha \rightarrow \gamma, \neg\gamma, \neg\beta, \delta \leftrightarrow \beta\}$. Let $\Lambda = \{\alpha, \neg\beta\}$. Consider the following argument tree with root A and canonical undercut A':

$$\begin{array}{c} \langle\{\alpha \vee \beta, \neg\beta\}, \alpha \vee \neg\delta\rangle \\ \uparrow \\ \langle\{\alpha \rightarrow \gamma, \neg\gamma\}, \Diamond\rangle \end{array}$$

Then, $\mathsf{Empathy}(A, \Lambda) = 1$ and $\mathsf{Antipathy}(A', \Lambda) = 1/3$.

Example 6.2.3 Consider the following propositions used by a politician for making an argument to a group of pensioners that taxes need to rise:

t Tax increase is necessary.

b Better treatments for serious diseases are available.

c More money has to be spent on better treatments.

Thus, for the argument $A_1 = \langle \{b, b \to c, c \to t\}, t \rangle$, with beliefbase $\Lambda = \{b, b \to c\}$, we have $\mathsf{Empathy}(A_1, \Lambda) = 1/2$ and $\mathsf{Antipathy}(A_1, \Lambda) = 0$.

Now consider an alternative set of propositions that could be used by a politician for making an argument to a group of pensioners that taxes need to rise:

l There is a lack of cheap entertainment facilities for teenagers to relax.

m Current teenagers have a more stressful life than any previous generation.

n The government should provide cheap entertainment facilities for teenagers.

Thus, for the following additions to the beliefbase $\Lambda = \{\neg l, \neg m, \neg n\}$ and for the following argument

$$A_2 = \langle \{l, m, l \wedge m \to n, n \to t\}, t \rangle$$

we have $\mathsf{Empathy}(A_2, \Lambda) = 0$ and $\mathsf{Antipathy}(A_2, \Lambda) = 3/4$. Hence, the pensioners demonstrate a high degree of empathy and no antipathy for A_1, and a high degree of antipathy and no empathy for A_2.

The following results show basic properties for the empathy and antipathy measures and for interrelationships between them.

Proposition 6.2.2 Let A be an argument, and let Λ be a beliefbase:

1. $\mathsf{Empathy}(A, \Lambda) = 0$ iff $\mathsf{Support}(A) \cup \Lambda \vdash \perp$
2. $\mathsf{Empathy}(A, \Lambda) = 1$ iff for all $\alpha \in \mathsf{Support}(A)$, $\Lambda \vdash \alpha$

Proposition 6.2.3 Let A be an argument, and let Λ be a beliefbase:

1. $\mathsf{Antipathy}(A, \Lambda) = 0$ iff $\mathsf{Support}(A) \cup \Lambda \nvdash \perp$
2. $\mathsf{Antipathy}(A, \Lambda) = 1$ implies for all $\alpha \in \mathsf{Support}(A)$, $\Lambda \vdash \neg\alpha$

Proposition 6.2.4 Let A be an argument and Λ a beliefbase:

$$\mathsf{Empathy}(A, \Lambda) = 0 \quad \text{iff } \mathsf{Antipathy}(A, \Lambda) \neq 0$$

Proposition 6.2.5 Let $\langle \Phi, \alpha \rangle$ be an argument, let $\langle \Psi, \Diamond \rangle$ be a canonical undercut for it, and let Λ be a beliefbase:

If $\mathsf{Empathy}(\langle \Phi, \alpha \rangle, \Lambda) = 1$, then $\mathsf{Empathy}(\langle \Psi, \Diamond \rangle, \Lambda) = 0$

If $\mathsf{Antipathy}(\langle \Phi, \alpha \rangle, \Lambda) = 1$, then $\mathsf{Antipathy}(\langle \Psi, \Diamond \rangle, \Lambda) \neq 1$

Thus, given a beliefbase, the measures of empathy and antipathy provide an intuitive way of evaluating how convincing individual arguments are for the intended audience. Next, we consider how we can use these measures for argument trees.

6.2.2 More Convincing Argument Trees

Given a knowledgebase Δ, we may be able to generate more than one argument tree for a conclusion. Using a beliefbase, we can select one of these trees to optimize believability. We can also seek better argument trees by ignoring some of the formulae in Δ. In other words, we can delete formulae from Δ to give Δ' and then generate argument trees from Δ'. As a third alternative, we can improve believability by pruning some arguments from an argument tree. In these cases, we can use the following definitions for antipathy and empathy for trees where the Attackers and Defenders functions are defined in section 3.5.1.

Definition 6.2.5 The degree of **tree empathy** for T with respect to a beliefbase Λ, denoted by $\mathsf{TreeEmpathy}(T, \Lambda)$, is defined as follows:

$$\sum_{A \in \mathsf{Defenders}(T)} \mathsf{Empathy}(A, \Lambda) - \sum_{A \in \mathsf{Attackers}(T)} \mathsf{Empathy}(A, \Lambda)$$

Since it would be desirable to have the audience empathize with the arguments that are defenders and not empathize with the arguments that are attackers, we seek a positive value for $\mathsf{TreeEmpathy}(T, \Lambda)$. Ideally, we would want the audience to empathize completely with each argument in $\mathsf{Defenders}(T)$ and none in $\mathsf{Attackers}(T)$, in which case $\mathsf{TreeEmpathy}(T, \Lambda)$ is $|\mathsf{Defenders}(T)|$.

Definition 6.2.6 The degree of **tree antipathy** for T with respect to a beliefbase Λ, denoted by $\mathsf{TreeAntipathy}(T, \Lambda)$, is defined as follows:

$$\sum_{A \in \mathsf{Attackers}(T)} \mathsf{Antipathy}(A, \Lambda) - \sum_{A \in \mathsf{Defenders}(T)} \mathsf{Antipathy}(A, \Lambda)$$

Similarly, since it would be desirable to have the audience feel antipathy with the arguments that are attackers and not feel antipathy with the arguments that are defenders, we seek a positive value for $\mathsf{TreeAntipathy}(T, \Lambda)$. Ideally, we would want the audience to feel complete antipathy toward each argument in $\mathsf{Attackers}(T)$ and feel none toward $\mathsf{Defenders}(T)$, in which case $\mathsf{TreeAntipathy}(T, \Lambda)$ is $|\mathsf{Attackers}(T)|$.

Example 6.2.4 Consider $\Delta = \{\alpha \leftrightarrow \neg\delta, \beta, \beta \rightarrow \alpha, \gamma \wedge \neg\beta, \neg\gamma, \delta, \neg\delta\}$, giving the argument tree T_1 below:

$$\begin{tikzcd}[column sep=small]
& \langle \{\beta \to \alpha, \beta\}, \alpha \rangle & \\
\langle \{\gamma \wedge \neg\beta\}, \diamond \rangle \arrow[ur] & & \langle \{\delta, \alpha \leftrightarrow \neg\delta\}, \diamond \rangle \arrow[ul] \\
\langle \{\neg\gamma\}, \diamond \rangle \arrow[u] & & \langle \{\neg\delta\}, \diamond \rangle \arrow[u]
\end{tikzcd}$$

For $\Lambda = \{\delta, \neg\gamma\}$,

Argument	Empathy	Antipathy
$\langle \{\beta \to \alpha, \beta\}, \alpha \rangle$	0	0
$\langle \{\gamma \wedge \neg\beta\}, \diamond \rangle$	0	1/3
$\langle \{\delta, \alpha \leftrightarrow \neg\delta\}, \diamond \rangle$	1/2	0
$\langle \{\neg\gamma\}, \diamond \rangle$	1	0
$\langle \{\neg\delta\}, \diamond \rangle$	0	1/2

Hence, $\mathsf{TreeEmpathy}(T_1, \Lambda) = 1/2$ and $\mathsf{TreeAntipathy}(T_1, \Lambda) = -1/6$. Thus, the audience has more empathy for the defending arguments than the attacking arguments, but it also has more antipathy for the defending arguments than the attacking arguments.

Example 6.2.5 Consider $\Delta = \{\alpha \leftrightarrow \neg\delta, \beta, \beta \to \alpha, \gamma \wedge \neg\beta, \neg\gamma, \delta, \neg\delta\}$, giving the argument tree T_2 below:

$$\begin{tikzcd}[column sep=small]
& \langle \{\delta, \alpha \leftrightarrow \neg\delta\}, \neg\alpha \rangle & \\
\langle \{\beta, \beta \to \alpha\}, \diamond \rangle \arrow[ur] & & \langle \{\neg\delta\}, \diamond \rangle \arrow[ul] \\
\langle \{\gamma \wedge \neg\beta\}, \diamond \rangle \arrow[u] & & \\
\langle \{\neg\gamma\}, \diamond \rangle \arrow[u] & &
\end{tikzcd}$$

For $\Lambda = \{\delta, \neg\gamma\}$,

Argument	Empathy	Antipathy
$\langle \{\delta, \alpha \leftrightarrow \neg\delta\}, \neg\alpha \rangle$	1/2	0
$\langle \{\beta, \beta \to \alpha\}, \diamond \rangle$	0	0
$\langle \{\neg\delta\}, \diamond \rangle$	0	1/2
$\langle \{\gamma \wedge \neg\beta\}, \diamond \rangle$	0	1/3
$\langle \{\neg\gamma\}, \diamond \rangle$	1	0

Hence, $\mathsf{TreeEmpathy}(T_2, \Lambda) = -1/2$ and $\mathsf{TreeAntipathy}(T_2, \Lambda) = 1/6$. Thus, the audience has less empathy for the defending arguments than the attacking arguments, but it also has less antipathy for the defending arguments than the attacking arguments.

In the next example, we illustrate how tree empathy and tree antipathy behave in two related practical reasoning scenarios.

Example 6.2.6 Consider the following propositions:

n It is a nice day.

b We should have a barbecue.

r The weather forecast is rain.

They can be used for the following argument tree T_3:

$$\begin{array}{c} \langle \{n, n \to b\}, b \rangle \\ \uparrow \\ \langle \{r, r \to \neg n\}, \diamond \rangle \end{array}$$

For $\Lambda = \{n \to b, r, r \to \neg n\}$, we get the following, which summarizes that the empathy for defenders and antipathy for attackers are both negative, hence indicating the case for b is not convincing:

$$\begin{aligned} &\mathsf{TreeEmpathy}(T_3, \Lambda) \\ &\quad = \mathsf{Empathy}(\langle \{n, n \to b\}, b \rangle, \Lambda) - \mathsf{Empathy}(\langle \{r, r \to \neg n\}, \diamond \rangle, \Lambda) \\ &\quad = -1 \end{aligned}$$

$$\begin{aligned} &\mathsf{TreeAntipathy}(T_3, \Lambda) \\ &\quad = \mathsf{Antipathy}(\langle \{r, r \to \neg n\}, \diamond \rangle, \Lambda) - \mathsf{Antipathy}(\langle \{n, n \to b\}, b \rangle, \Lambda) \\ &\quad = -1/3 \end{aligned}$$

Now consider the additional proposition:

s The astrological forecast says it is a bad day for a barbecue.

Using this additional proposition, we can present the following argument tree T_4:

$$\begin{array}{c} \langle \{n, n \to b\}, b \rangle \\ \uparrow \\ \langle \{s, s \to \neg n\}, \diamond \rangle \end{array}$$

For $\Lambda' = \{n, n \to b, \neg s\}$, we get the following, which summarizes that the empathy for defenders and antipathy for attackers are both positive, hence indicating the case for b is convincing:

$$\begin{aligned}
&\mathsf{TreeEmpathy}(T_4, \Lambda')\\
&\quad = \mathsf{Empathy}(\langle\{n, n \to b\}, b\rangle, \Lambda') - \mathsf{Empathy}(\langle\{s, s \to \neg n\}, \diamond\rangle, \Lambda')\\
&\quad = 1
\end{aligned}$$

$$\begin{aligned}
&\mathsf{TreeAntipathy}(T_4, \Lambda')\\
&\quad = \mathsf{Antipathy}(\langle\{s, s \to \neg n\}, \diamond\rangle, \Lambda') - \mathsf{Antipathy}(\langle\{n, n \to b\}, b\rangle, \Lambda')\\
&\quad = 2/3
\end{aligned}$$

Improving believability involves increasing both tree empathy and tree antipathy. However, there may be a trade-off between these two parameters. In other words, it may be the case that increasing tree empathy causes a decrease in tree antipathy, and increasing tree antipathy causes a decrease in tree empathy.

6.3 Cost of Understanding Arguments

In order to assess arguments for an intended audience, we have already suggested that we may need to consider the cost of understanding arguments. When a recipient of an argument considers it, there is a cost involved in understanding it. The cost can reflect the work required if the recipient checks the correctness of the argument (consistency, entailment, and minimality) as well as the work required in relating the argument (both the support and the claim) to the agent's own knowledge. This cost depends on the agent and on the argument.

A number of factors may contribute to the cost of understanding an argument for an agent. These may include the amount of information involved in understanding the argument (e.g., many complex statements vs. a couple of simple statements). Some of these factors can also be considered in terms of "the cost of deduction." For example, some agents are equally happy to use modus tollens as they are to use modus ponens; others find contrapositive reasoning more difficult, and therefore more costly, to understand. Similarly, the use of double negation in argumentation can increase the cost of understanding. Thus, again, we stress that the cost of understanding depends on the recipient as well as the argument.

How arguments relate to the recipient's knowledge is essential. A key factor here is the choice of statements. For example, *you are a liar* and *you are a perpetrator of terminological inexactitudes* are equivalent statements in some contexts, but for some agents the first has a lower cost of understanding than the second. As an aside, in the House of Commons, which is the main chamber of the United Kingdom parliament, it is not allowed for a member to call another a liar. This has led to an interesting range of alternatives being developed for the phrase *a liar*, such as being *economical with the truth*. As another illustration, consider the following example, where different phrases can be used for the same concept.

Example 6.3.1 Consider the following statements:

p	I am on holiday.	p'	Je suis en vacances.
q	I am happy.	q'	Je suis heureux.

Intuitively, we can regard the following arguments as being equivalent in most respects. But for an agent who is fluent in English, but not fluent in French, the cost of understanding for the second argument is higher than for the first argument.

$\langle \{p, p \to q\}, q \rangle$

$\langle \{p', p' \to q'\}, q' \rangle$

Another kind of issue arising is the use of technical language. In an argument about the need for adding salt when making bread, one can use either of the following two propositions: *add* 2 *ml of salt* and *add half a teaspoon of salt*. Here some audiences prefer the more technical language, whereas others prefer the more informal language. If the less preferred language is used for the audience, then the cost of understanding may be higher for that audience.

Example 6.3.2 I am back in Europe after a trip to Chicago, where I bought a cookbook, from which I follow the instructions to prepare a chocolate cake. At some point, the recipe says

Spread into a lightly greased $8'' \times 12'' \times 2''$ pan and bake in preheated 350°F oven

which is the same as saying

Spread into a lightly greased 20 cm $\times$ 30 cm $\times$ 5 cm pan and bake in preheated 180°C oven

but the cost of understanding the former (using inches and Fahrenheit degrees) is likely to be higher if I am used to international measures.

In order to take into account the cost of understanding, we assume that we can define a cost function for a representative agent. As there are a range of ways of estimating cost, we provide a general definition to be instantiated by particular cost functions. In this way, cost is a parameter in our framework. An instantiation of a cost function is given in definition 6.1.5.

Definition 6.3.1 A **cost function** is a mapping $\mathsf{Cost} : \Lambda \to O$, where $\langle O, \leq \rangle$ is a poset such that if $\Phi \subseteq \Psi$, then $\mathsf{Cost}(\langle \Phi, \gamma \rangle) \leq \mathsf{Cost}(\langle \Psi, \gamma \rangle)$.

The reason for the above definition is that we intend to compare the cost of arguments whose claim is the same because we want to use the measure as part of being selective. For example, most audiences presumably find that understanding $\langle \emptyset, \alpha \rangle$, where α is Pierce's law (i.e., $(\alpha \to (\beta \to \alpha)) \to \alpha$), is harder than understanding $\langle \{\alpha\}, \alpha \rangle$.

While we present general instances of Cost, it is desirable to define these functions for a particular audience. We may wish to impose some constraints on these functions such as the following.

Definition 6.3.2 A cost function is **homogeneous** if it satisfies the condition below:

$$\mathsf{Cost}(\langle \Phi, \alpha \rangle) - \mathsf{Cost}(\langle \Psi, \alpha \rangle) = \mathsf{Cost}(\langle \Phi, \beta \rangle) - \mathsf{Cost}(\langle \Psi, \beta \rangle)$$

The idea underlying the previous definition is that the cost of understanding one support rather than another should not differ more when considering one claim rather than another.

We leave the notion of a cost function general so that it can be developed for particular applications. Together with the measure of resonance, and/or the measures of empathy/antipathy, it can be used to compare the trade-off of increasing impact versus increasing cost or the trade-off of increasing believability versus increasing cost.

6.4 Discussion

With the first proposal for extrinsic analysis presented in this chapter, arguments can be analyzed to select a subset that would have higher impact for an audience. This analysis could be useful in a variety of professional domains such as politics and law. Already, for some high profile U.S. court cases, lawyers are commissioning research based on focus groups for shaping the presentation of cases.

Taking the audience into account also raises the possibility of choosing the audience to suit the arguments. In other words, once we start evaluating the arguments with respect to audiences, we can select an audience for which a given set of arguments will have substantial impact. We see this in invited audiences for speeches by senior politicians during election campaigns. In this case, the significant impact on the audience looks good for the television coverage. Another example of selecting audiences comes from jury selections that are allowed in certain trials in the United States.

The proposal for evaluating believability of arguments, as presented in our second proposal for extrinsic analysis, is another way for argumentation systems to interact intelligently with users. Given a beliefbase that reflects the intended audience, the values for empathy/antipathy provide an intuitive ranking over arguments. This may then be used to optimize the presentation of argumentation for audiences. It may also be used to define judge functions that capture important criteria for deciding when an initiating argument is warranted.

The framework for convincing arguments can be used for manipulating audiences. Consider an argument tree with two nodes. Suppose the root is A_1 and the undercut is A_2. Now imagine a politician who gives this tree knowing that the audience will have some empathy for A_1 and a high degree of antipathy for A_2. The politician could present this information in such a way that he is feigning honesty by presenting a undercut (i.e., A_2) to an argument that he wants the audience to accept (i.e., A_1). With this strategy, the audience may think the politician is being honest with them because he is admitting a shortcoming (i.e., A_2) to his proposal (i.e., A_1).

However, this approach is not just about manipulating audiences. Consider another example where a scientist wants to present a root argument A_3 as a new piece of scientific knowledge resulting from laboratory experiments but wants, for the sake of completeness, to also present some exceptional data that negates the new finding. Suppose this exceptional data is represented by A_4. This would be an undercut to A_3. Now also suppose the scientist and the audience have a high degree of empathy with A_3 and a high degree of antipathy with A_4. Here there is no intention to manipulate the audience. It is just that the scientist wants to clearly present any doubts that may exist in the new scientific knowledge.

6.5 Bibliographic Notes

In the philosophical study of rhetoric and argumentation, the importance of taking the audience into account when presenting arguments is well documented, as expounded by Perelman [Per82] (see also [Wal89, CC92, HB05]). Yet it is only recently that consideration has been given to bringing the audience into account in formal frameworks for argumentation. The first attempt to formalize the notion of impact is [Hun04b], and the first attempt to formalize the notion of believability is [Hun04a].

7 Algorithms for Argumentation

In this chapter, we consider algorithms for constructing arguments and argument trees. Let us start by considering the construction of individual arguments. If Δ is a knowledgebase, and we are interested in a claim α, how can we find an argument $\langle \Phi, \alpha \rangle$ where $\Phi \subseteq \Delta$? Deciding whether a set of propositional classical formulae is consistent is an NP-complete decision problem, and deciding whether a set of propositional classical formulae entails a given formula is a co-NP-complete decision problem [GJ79]. However, if we consider the problem as an abduction problem, where we seek the existence of a minimal subset of a set of formulae that implies the consequent, then the problem is in the second level of the polynomial hierarchy [EG95]. Even worse, deciding whether a set of first-order classical formulae is consistent is an undecidable decision problem [BBJ02]. Thus, even finding the basic units of argumentation is computationally challenging.

In this chapter, we start by introducing some basic algorithms for argumentation (section 7.1). These simple algorithms are for finding supports for an argument, for constructing an argument tree for a given subject, and for constructing an argument tree for a given root argument. These algorithms are naive in the sense that no techniques or strategies are considered for addressing the computational complexity and undecidability issues arising from using classical propositional or first-order logic.

Then, to address the computational viability problems, we propose three approaches: (1) compilation of a knowledgebase, based on minimal inconsistent subsets of the knowledgebase (section 7.2); (2) construction of contours of a knowledgebase, a form of lemma generation (section 7.3); and (3) construction of approximate argument trees composed of approximate arguments, where each approximate argument is obtained as a relaxation of one or more of the entailment, consistency, and minimality conditions required for an argument (section 7.4).

7.1 Basic Algorithms

In this section, we provide some simple algorithms for argumentation that are not intended to address the computational viability issues inherent in coherence-based reasoning. The algorithms offer some naive solutions for generating arguments, argument trees, and argument structures that will work for small knowledgebases. For these, we require the following subsidiary functions, where Δ is a knowledgebase, Φ is a subset of Δ, α is a formula, and Θ is a subset of $\wp(\Delta)$:

- $\texttt{Subsets}(\Delta, C)$, which returns the set of subsets of Δ that have a cardinality C;
- $\texttt{NotSuperSet}(\Phi, \Theta)$, which is a Boolean function that is true if there is no member of Θ that is a subset of Φ;
- $\texttt{Entails}(\Phi, \alpha)$, which is a Boolean function that is true if $\Phi \vdash \alpha$;
- $\texttt{Consistent}(\Phi)$, which is a Boolean function that is true if $\Phi \nvdash \bot$.

Now we start with an algorithm for finding the supports for an argument.

Definition 7.1.1 Given a set of formulae Δ, and a formula α, the $\texttt{GenerateSupports}$ algorithm generates $\Theta \subseteq \wp(\Delta)$, which is the set of all the supports for arguments for the claim α:

```
GenerateSupports(Δ, α)
    Θ = ∅
    C = 1
    while C ≤ |Δ|
        Ω = Subsets(Δ, C)
        while Ω ≠ ∅
            let Φ be an arbitrary member of Ω
            if NotSuperSet(Φ, Θ) & Entails(Φ, α) & Consistent(Φ)
            then Θ = Θ ∪ {Φ}
            Ω = Ω\{Φ}
        C = C + 1
    return Θ
```

The $\texttt{GenerateSupports}$ algorithm works by considering each subset of Δ as a possible support for an argument for α. The search is started with the subsets of smallest cardinality, and then with each cycle of the outer while loop, the cardinality of the subsets is increased by one. The inner while loop checks each of the subsets Φ of a given cardinality. The if statement checks whether Φ is not a superset of a support that has already been identified, that Φ entails the consequent α, and that Φ is consistent. If all these conditions are met, then Φ is added to the set of supports Θ that is eventually returned by the algorithm. Thus, if Φ is in Θ, then $\langle \Phi, \alpha \rangle$ is an argument.

Now we consider an algorithm, `GenerateATree`, for constructing an argument tree for a subject. The `GenerateATree` algorithm undertakes a nonrecursive depth-first search based on a stack. This means that the `GenerateSupports` function can be used to find all the canonical undercuts to an argument before the next stage of depth-first search. The algorithm assumes the normal push and pop operations for a stack (see e.g., [CCLR091]).

Definition 7.1.2 Given a set of formulae Δ, and a formula α, the `GenerateATree` algorithm finds an argument tree for α represented by a pair (N, A) where N is the set of nodes and A is the set of arcs for the argument tree. For this, we assume a subsidiary function `BranchSupport`$(\langle \Psi, \beta \rangle, A)$ that returns the union of the supports for the arguments on the branch from the argument $\langle \Psi, \beta \rangle$ to the root:

`GenerateATree`(Δ, α)
    $N = \emptyset$
    $A = \emptyset$
    let S be an empty stack
    let $\langle \Phi, \alpha \rangle$ be s.t. $\Phi \in$ `GenerateSupports`(Δ, α)
    push $\langle \Phi, \alpha \rangle$ onto S
    while S is nonempty
        let $\langle \Psi, \beta \rangle$ be the top of the stack S
        pop $\langle \Psi, \beta \rangle$ from S
        for each $\langle \Gamma, \neg(\bigwedge \Psi) \rangle$ s.t. $\Gamma \in$ `GenerateSupports`$(\Delta, \neg(\bigwedge \Psi))$
            if $\Gamma \not\subseteq$ `BranchSupport`$(\langle \Psi, \beta \rangle, A)$
                then push $\langle \Gamma, \neg(\bigwedge \Psi) \rangle$ onto S
                    and $A = A \cup \{(\langle \Gamma, \neg(\bigwedge \Psi) \rangle, \langle \Psi, \beta \rangle)\}$
                    and $N = N \cup \{\langle \Gamma, \neg(\bigwedge \Psi) \rangle\}$
    return (N, A)

This tree is arbitrarily chosen from the argument trees for α that can be constructed from Δ, since the root is an arbitrary choice from the possible roots available from `GenerateSupports`(Δ, α).

The next algorithm, `GenerateTheTree`, is a variation on the `GenerateATree` algorithm. This variation starts with a given argument for the root of the argument tree. It constructs the argument tree for a subject by again undertaking a nonrecursive depth-first search based on a stack.

Definition 7.1.3 Given a set of formulae Δ, and an argument $\langle\Phi, \alpha\rangle$, the following algorithm `GenerateTheTree` finds the argument tree with $\langle\Phi, \alpha\rangle$ as the root. For this, we assume a subsidiary function `BranchSupport`$(\langle\Psi, \beta\rangle, A)$ that returns the union of the supports for the arguments on the branch from the argument $\langle\Psi, \beta\rangle$ to the root:

`GenerateTheTree`(Δ, Φ, α)
  $N = \emptyset$
  $A = \emptyset$
  let S be an empty stack
  push $\langle\Phi, \alpha\rangle$ onto S
  while S is nonempty
    let $\langle\Psi, \beta\rangle$ be the top of the stack S
    pop $\langle\Psi, \beta\rangle$ from S
    for each $\langle\Gamma, \neg(\bigwedge\Psi)\rangle$ s.t. $\Gamma \in$ `GenerateSupports`$(\Delta, \neg(\bigwedge\Psi))$
      if $\Gamma \not\subseteq$ `BranchSupport`$(\langle\Psi, \beta\rangle, A)$
        then push $\langle\Gamma, \neg(\bigwedge\Psi)\rangle$ onto S
          and $A = A \cup \{(\langle\Gamma, \neg(\bigwedge\Psi)\rangle, \langle\Psi, \beta\rangle)\}$
          and $N = N \cup \{\langle\Gamma, \neg(\bigwedge\Psi)\rangle\}$
  return (N, A)

These naive algorithms could be adapted to take a fourth parameter n that would be a number to indicate the maximum depth of the tree to be built. In this way, the algorithms would not construct an argument tree with a leaf having any depth greater than n.

7.2 Argument Compilation

Argument compilation is a process that involves finding all the minimal inconsistent subsets of Δ and then forming a hypergraph from this collection of subsets. While argument compilation is expensive, once Δ has been compiled, it can be queried relatively cheaply.

We proceed as follows: We provide some propositions concerning minimal inconsistent subsets of a knowledgebase that we use to motivate the use of compilations (section 7.2.1), we present our definition for compilation together with a theorem relating compilations and argumentation (section 7.2.2), we present the algorithm for generating arguments from a compilation and follow this with illustrative examples and completeness and correctness theorems (section 7.2.3), and we present an algorithm for generating a compilation from a knowledgebase (section 7.2.4).

7.2.1 Minimal Inconsistent Subsets

If Δ is consistent, then there are no counterarguments; otherwise the minimal inconsistent subsets of a knowledgebase Δ summarize the conflicts that arise in Δ.

Definition 7.2.1 For a knowledgebase Δ, the set of **minimal inconsistent subsets** of Δ, denoted $\mathsf{MinIncon}(\Delta)$, is defined as follows:

$$\mathsf{MinIncon}(\Delta) = \{X \subseteq \Delta \mid X \vdash \bot \text{ and } \forall Y \subset X, Y \nvdash \bot\}$$

We are interested in the minimal inconsistent subsets of Δ because of the following propositions.

Proposition 7.2.1 If $\langle \Psi, \diamond \rangle$ is a canonical undercut, then there is an $X \in \mathsf{MinIncon}(\Delta)$, such that $\Psi \subset X$.

However, if $\langle \Phi, \alpha \rangle$ is the root of an argument tree, it is not necessarily the case that there is an $X \in \mathsf{MinIncon}(\Delta)$, such that $\Phi \subset X$.

Example 7.2.1 Let $\Delta = \{\alpha, \neg\alpha, \beta, \neg\beta\}$. Hence the minimal inconsistent subsets are $X_1 = \{\alpha, \neg\alpha\}$ and $X_2 = \{\beta, \neg\beta\}$. Now consider the argument at the root of an argument tree $\langle \Phi, \alpha \wedge \beta \rangle$ where $\Phi = \{\alpha, \beta\}$. Hence, $\Phi \not\subset X_1$ and $\Phi \not\subset X_2$.

The following result shows how we can construct our canonical undercuts from the set of minimal inconsistent subsets of Δ.

Proposition 7.2.2 Let $\langle \Phi, \alpha \rangle$ be an argument. For all $X \in \mathsf{MinIncon}(\Delta)$, if Φ is a nonempty subset of X, then $\langle X \backslash \Phi, \diamond \rangle$ is a canonical undercut for $\langle \Phi, \alpha \rangle$.

Example 7.2.2 Let $\mathsf{MinIncon}(\Delta) = \{\{\beta, \neg\beta \vee \neg\gamma, \gamma\}\}$. Consider the argument $\langle \{\neg\beta \vee \neg\gamma\}, \neg(\beta \wedge \gamma) \rangle$. Clearly, $\{\neg\beta \vee \neg\gamma\}$ is a nonempty subset of $\{\beta, \neg\beta \vee \neg\gamma, \gamma\}$. Hence, $\langle \{\beta, \neg\beta \vee \neg\gamma, \gamma\} \backslash \{\neg\beta \vee \neg\gamma\}, \diamond \rangle$ is a canonical undercut of $\langle \{\neg\beta \vee \neg\gamma\}, \neg(\beta \wedge \gamma) \rangle$.

We generalize proposition 7.2.2 as follows.

Proposition 7.2.3 Let $\langle \Phi, \alpha \rangle$ be an argument. For all $X \in \mathsf{MinIncon}(\Delta)$, if $\Phi \cap X \neq \emptyset$, and there is no $Y \in \mathsf{MinIncon}(\Delta)$, such that $(Y \backslash \Phi) \subset (X \backslash \Phi)$, then $\langle X \backslash \Phi, \diamond \rangle$ is a canonical undercut for $\langle \Phi, \alpha \rangle$.

Example 7.2.3 Let $\mathsf{MinIncon}(\Delta) = \{X_1, X_2\}$ where $X_1 = \{\alpha, \neg\alpha\}$ and $X_2 = \{\beta, \neg\beta\}$. Now consider the argument $\langle \Phi, \alpha \wedge \beta \rangle$ where $\Phi = \{\alpha, \beta\}$. Here $\langle X_1 \backslash \Phi, \diamond \rangle$ and $\langle X_2 \backslash \Phi, \diamond \rangle$ are each a canonical undercut for $\langle \Phi, \alpha \rangle$.

We get the converse of proposition 7.2.2 as follows.

Proposition 7.2.4 If $\langle \Psi, \diamond \rangle$ is a canonical undercut for $\langle \Phi, \alpha \rangle$, then there is an X such that $X \in \mathsf{MinIncon}(\Delta)$ and $\Psi = X \backslash \Phi$.

Example 7.2.4 Let $\Delta = \{\beta, \beta \to \alpha, \beta \to \neg\alpha\}$. There is only one minimal inconsistent subset, which is $\{\beta, \beta \to \alpha, \beta \to \neg\alpha\}$. Now consider the argument $\langle \{\beta, \beta \to \alpha\}, \alpha \rangle$. There is only one canonical undercut, which is $\langle \{\beta \to \neg\alpha\}, \diamond \rangle$. Furthermore, we have the following holding between the support of the argument, the support of the undercut, and the minimal inconsistent subset:

$$\{\beta \to \neg\alpha\} = \{\beta, \beta \to \alpha, \beta \to \neg\alpha\} \backslash \{\beta, \beta \to \alpha\}$$

Minimal inconsistent subsets of a knowledgebase are not necessarily disjoint. Consider $X, Y \in \mathsf{MinIncon}(\Delta)$ where $X \cap Y \neq \emptyset$. Furthermore, if the support of an argument $\langle \Phi, \alpha \rangle$ is a subset of X, and $Y \cap \Phi$ is nonempty, and there is no $Z \in \mathsf{MinIncon}(\Delta)$ such that $(Z \backslash \Phi) \subset (Y \backslash \Phi)$, then $\langle Y \backslash \Phi, \diamond \rangle$ is a canonical undercut for $\langle \Phi, \alpha \rangle$ as captured in the following result.

Proposition 7.2.5 Let $\langle \Phi, \alpha \rangle$ be an argument. For all $X, Y \in \mathsf{MinIncon}(\Delta)$, if $X \cap Y \neq \emptyset$, and $\Phi \subseteq X$, and $Y \cap \Phi \neq \emptyset$, and there is no $Z \in \mathsf{MinIncon}(\Delta)$, such that $(Z \backslash \Phi) \subset (Y \backslash \Phi)$, then $\langle Y \backslash \Phi, \diamond \rangle$ is a canonical undercut for $\langle \Phi, \alpha \rangle$.

Example 7.2.5 Let $X_1, X_2 \subset \Delta$ where $X_1 = \{\alpha, \alpha \to \neg\beta, \beta\}$ and $X_2 = \{\alpha, \neg\alpha\}$. So $X_1, X_2 \in \mathsf{MinIncon}(\Delta)$ and $X_1 \cap X_2 \neq \emptyset$. Now consider the argument $\langle \Phi, \neg\beta \rangle$ where $\Phi = \{\alpha, \alpha \to \neg\beta\}$. So $\Phi \subset X_1$ and $\Phi \cap X_2 \neq \emptyset$. Furthermore, $\langle X_2 \backslash \Phi, \diamond \rangle$ is an undercut for $\langle \Phi, \diamond \rangle$.

Example 7.2.6 Let $X_1, X_2, X_3 \subset \Delta$ where

$$X_1 = \{\neg\alpha, \neg\beta, \alpha \vee \beta\}$$

$$X_2 = \{\neg\alpha, \alpha \vee \neg\beta \vee \neg\gamma, \gamma, \beta\}$$

$$X_3 = \{\beta, \neg\beta\}$$

Let $\mathsf{MinIncon}(\Delta) = \{\{X_1, X_2, X_3\}\}$. Now consider the argument $\langle \{\alpha \vee \beta\}, \alpha \vee \beta \rangle$. Let $\Phi = X_1 \backslash \{\alpha \vee \beta\} = \{\neg\alpha, \neg\beta\}$. Thus, $\langle \Phi, \diamond \rangle$ is a canonical undercut for $\langle\{\alpha \vee \beta\}, \alpha \vee \beta\rangle$. Furthermore, $\Phi \cap X_2 \neq \emptyset$ and $\Phi \cap X_3 \neq \emptyset$. However, $\langle X_2 \backslash \Phi, \diamond\rangle$ is not a canonical undercut for $\langle \Phi, \diamond\rangle$ because $(X_3 \backslash \Phi) \subset (X_2 \backslash \Phi)$, whereas $\langle X_3 \backslash \Phi, \diamond\rangle$ is a canonical undercut for $\langle \Phi, \diamond\rangle$.

We also see that every canonical undercut for a canonical undercut can be found using these overlapping minimal inconsistent subsets of a knowledgebase.

Proposition 7.2.6 Let $\langle \Phi, \diamond\rangle$ be a canonical undercut. If $\langle \Psi, \diamond\rangle$ is a canonical undercut of $\langle \Phi, \diamond\rangle$, then $\Psi \subset Y$ for some $Y \in \mathsf{MinIncon}(\Delta)$ such that $X \cap Y \neq \emptyset$ for every $X \in \mathsf{MinIncon}(\Delta)$ satisfying $\Phi \subset X$.

These results indicate that for any knowledgebase Δ, if we have $\mathsf{MinIncon}(\Delta)$, we can find all undercuts to any argument without recourse to a consequence relation or a satisfiability check.

7.2.2 Compilation of a Knowledgebase

Argument compilation is a process that involves finding all the minimal inconsistent subsets of Δ and then forming a hypergraph from this collection of subsets. Given a knowledgebase Δ, the function $\mathsf{Compilation}(\Delta)$ gives the hypergraph that can then be used for efficiently building argument trees.

Definition 7.2.2 Let $\mathsf{Compilation}(\Delta) = (N, A)$ where $N = \mathsf{MinIncon}(\Delta)$ and

$$A = \{(X, Y) \mid X, Y \in N \text{ and } X \neq Y \text{ and } X \cap Y \neq \emptyset\}$$

A graph is bidirectional if for each arc (X, Y) in the graph, (Y, X) is also in the graph. Clearly, for all Δ, $\mathsf{Compilation}(\Delta)$ is a bidirectional hypergraph, though $\mathsf{Compilation}(\Delta)$ may be disjoint. Consider, for example, $\Delta = \{\alpha, \neg\alpha, \beta, \neg\beta\}$.

Since the hypergraphs are bidirectional, we will represent them in the examples by just giving one of (X, Y) and (Y, X) for each of the arcs.

Example 7.2.7 Let $\Delta = \{\delta \wedge \neg\alpha, \alpha, \neg\alpha, \alpha \vee \beta, \neg\beta\}$. Hence $\mathsf{MinIncon}(\Delta) = \{X_1, X_2, X_3, X_4\}$, where $X_1 = \{\delta \wedge \neg\alpha, \alpha\}$, $X_2 = \{\alpha, \neg\alpha\}$, $X_3 = \{\neg\alpha, \alpha \vee \beta, \neg\beta\}$, and $X_4 = \{\delta \wedge \neg\alpha, \alpha \vee \beta, \neg\beta\}$. Thus, the nodes and arcs of the hypergraph are such that $N = \{X_1, X_2, X_3, X_4\}$ and $A = \{(X_1, X_2), (X_2, X_3), (X_3, X_4), (X_1, X_4)\}$.

The following result shows that an argument tree is composed from subgraphs of the compilation.

Theorem 7.2.1 Let $\mathsf{Compilation}(\Delta) = (N, A)$. If $\langle \Phi, \alpha \rangle$ is the root of a complete argument tree T and $\langle \Phi, \alpha \rangle$ is the parent node of $\langle \Psi, \diamond \rangle$, then the subtree of T rooted at $\langle \Psi, \diamond \rangle$ is isomorphic to a subgraph of (N, A).

Corollary 7.2.1 Let $\mathsf{Compilation}(\Delta) = (N, A)$. If $\langle \Psi, \diamond \rangle$ is a canonical undercut for $\langle \Phi, \diamond \rangle$, and $\langle \Phi, \diamond \rangle$ is itself a canonical undercut, then there is an $(X, Y) \in A$, such that $\Phi \subset X$ and $\Psi \subset Y$.

Thus, by walking over $\mathsf{Compilation}(\Delta)$, we can construct argument trees from Δ. If we have an argument for the root of the argument tree, then we can find undercuts and, by recursion, undercuts to undercuts, by following the arcs of the compilation. Moreover, we will argue in the next section that the cost of using the compilation is much lower than that of constructing arguments directly from Δ.

7.2.3 Generating Arguments from a Compilation

While argument compilation is expensive, once $\mathsf{Compilation}(\Delta)$ has been computed, it can be used relatively cheaply. Assume that we want to construct a complete argument tree from Δ with the argument $\langle \Phi, \alpha \rangle$ as its root. Now we want to use the support of this argument together with the hypergraph (i.e., $\mathsf{Compilation}(\Delta)$) to systematically build the argument tree.

The way we proceed is to find which nodes in the hypergraph have a nonempty set intersection with Φ. Suppose X is a node in the hypergraph and $\Phi \cap X \neq \emptyset$, then $X \backslash \Phi$ is the support of a counterargument to $\langle \Phi, \alpha \rangle$. Indeed, if there is no node Y such that $\Phi \cap Y \neq \emptyset$ and $(Y \backslash \Phi) \subset (X \backslash \Phi)$, then $\langle X \backslash \Phi, \diamond \rangle$ is a canonical undercut of $\langle \Phi, \alpha \rangle$ (according to proposition 7.2.3). We now proceed by recursion, looking for canonical undercuts to the canonical undercuts. For this, we use the arcs of the hypergraph. Suppose $\langle \Gamma_j, \diamond \rangle$ is a canonical undercut in the tree. Let $\langle \Gamma_i, \diamond \rangle$ be the parent of $\langle \Gamma_j, \diamond \rangle$ (i.e., $\langle \Gamma_j, \diamond \rangle$ is a canonical undercut of $\langle \Gamma_i, \diamond \rangle$). From this, we know there is a node in the hypergraph X such that $\Gamma_j = X \backslash \Gamma_i$ (cf. proposition 7.2.4). Hence, $\Gamma_j \subset X$. Moreover, if there is an arc in the hypergraph (X, Y) and $Y \cap \Gamma_i \neq \emptyset$, and there is no arc (X, Z) in the hypergraph such that $Z \backslash \Phi \subset Y \backslash \Phi$, then $\langle Y \backslash \Gamma_j, \diamond \rangle$ is a canonical undercut to $\langle \Gamma_j, \diamond \rangle$ (according to proposition 7.2.5).

Proposition 7.2.6 justifies that we pick *one* X, such that X is a superset of Φ, and then we can find all the other canonical undercuts of $\langle \Phi, \diamond \rangle$ by just following the arcs in $\mathsf{Compilation}(\Delta)$. By keeping track of which arcs we have traversed in the hypergraph, we can efficiently build a complete argument tree. The `Undercuts` algorithm, presented in definition 7.2.3, assumes that the argument for the root of the argument tree is given as input. Then the algorithm determines where this root argument is located in the hypergraph and undertakes a depth-first search identifying canonical undercuts by recursion. Thus, for a knowledgebase Δ, the `Undercuts` algorithm together with subsidiary algorithms only use $\mathsf{Compilation}(\Delta)$.

Definition 7.2.3 The algorithm `Undercuts` takes the root of an argument tree $\langle\Phi, \alpha\rangle$ and the hypergraph $\mathsf{Compilation}(\Delta) = (N, A)$ and returns the nodes of the argument tree. The first add statement puts the undercuts to the root into the solution, and the second add statement uses the algorithm `Subcuts` to find undercuts to undercuts by recursion and puts them into the solution:

```
Undercuts(⟨Φ, α⟩, N, A)
    FirstLevelSets = FirstLevel(⟨Φ, α⟩, N, A)
    Output = ∅
    while FirstLevelSets ≠ ∅
        remove X from FirstLevelSets
        add ⟨X\Φ, ◇⟩ is an undercut of ⟨Φ, α⟩ to Output
        add Subcuts(X\Φ, N\{X}, A, X, Φ) to Output
    return Output
```

The algorithm `FirstLevel` takes the root of an argument tree $\langle\Phi, \alpha\rangle$ and the hypergraph $\mathsf{Compilation}(\Delta) = (N, A)$ and returns the subset of N that can be used for constructing the undercuts to the root:

```
FirstLevel(⟨Φ, α⟩, N, A)
    Candidates = {X ∈ N | Φ ∩ X ≠ ∅}
    Output = ∅
    while Candidates ≠ ∅
        remove X from Candidates
        if there is no Y ∈ Candidates ∪ Output s.t. (Y\Φ) ⊂ (X\Φ)
        then add X to Output
    return Output
```

The algorithm `Subcuts` is a depth-first search algorithm that takes the support of an argument, Γ, the set of available hypernodes M in the depth-first search, the current hypernode X, and the current union of the supports for the arguments used on the branch up to the root S.

Subcuts(Γ, M, A, X, S)
    Output $= \emptyset$
    for all $Y \in M$
        if $(X, Y) \in A$ and $Y \not\subseteq S$ and $Y \cap \Gamma \neq \emptyset$
                and there is no $(X, Z) \in A$ such that $(Z \backslash \Gamma) \subset (Y \backslash \Gamma)$
        then add $\langle Y \backslash \Gamma, \diamond \rangle$ is an undercut of $\langle \Gamma, \diamond \rangle$ to Output
        and add Subcuts$(Y \backslash \Gamma, M \backslash \{Y\}, A, Y, S \cup Y)$ to Output
    return Output

We illustrate the key features of the Undercuts algorithm in example 7.2.8, which shows how the FirstLevel algorithm finds the nodes in N that intersect the root of the support, how these are then used to generate the undercuts to the root, and how the Subcuts algorithm finds the undercuts to the undercuts using the arcs in the graph.

Note, in the FirstLevel algorithm, that for each candidate node X, there is the following check (on the sixth line of the algorithm):

there is no $Y \in$ Candidates $\cup$ Output
        such that $(Y \backslash \Phi) \subset (X \backslash \Phi)$

This is to ensure that the support for each undercut is minimal: The support of the root can involve formulae from more than one minimally inconsistent set of formulae and so it is not the case that every node X with a nonempty intersection with Φ is such that $X \backslash \Phi$ is minimal (as illustrated in example 7.2.9).

Note that in Subcuts we check that $Y \cap \Gamma \neq \emptyset$ because we want to ensure that $Y \backslash \Gamma$ is inconsistent with Γ rather than with the support of the parent of $\langle \Gamma, \diamond \rangle$ (which is illustrated in example 7.2.10). Also, in Subcuts, we check that $Y \not\subseteq S$, because according to the definition of an argument tree, no undercut on a branch can have a support that is a subset of the union of the support of the ancestors.

Example 7.2.8 Consider $\Delta = \{\alpha, \neg\alpha, \beta \wedge \neg\beta, \gamma, \gamma \rightarrow \delta, \neg\delta, \neg\gamma \vee \delta\}$, which has the following compilation (N, A), where $N = \{X_1, X_2, X_3, X_4\}$ and $A = \{(X_4, X_3)\}$:

$X_1 = \{\alpha, \neg\alpha\}$

$X_2 = \{\beta \wedge \neg\beta\}$

$X_3 = \{\gamma, \gamma \rightarrow \delta, \neg\delta\}$

$X_4 = \{\gamma, \neg\gamma \vee \delta, \neg\delta\}$

Now suppose we have the following argument as the root of the argument tree:

$\langle \{\gamma, \gamma \rightarrow \delta\}, \delta \vee \beta \rangle$

We have support for one canonical undercut of the root (i.e., `FirstLevelSets` $= \{X_3\}$):

$X_3 \backslash \{\gamma, \gamma \rightarrow \delta\} = \{\neg\delta\}$

Also we have (X_4, X_3) as an arc, and so we have an undercut to the undercut:

$X_4 \backslash \{\neg\delta\} = \{\gamma, \neg\gamma \vee \delta\}$

There are no more arcs to consider, and so the net result is the following argument tree. The elements of N that are used are given adjacent to the arrow in the following:

$$\begin{array}{c} \langle \{\gamma, \gamma \rightarrow \delta\}, \delta \vee \beta \rangle \\ \uparrow X_3 \\ \langle \{\neg\delta\}, \diamond \rangle \\ \uparrow X_4 \\ \langle \{\gamma, \neg\gamma \vee \delta\}, \diamond \rangle \end{array}$$

In this example, we see that not all nodes are used in the construction of this tree. Furthermore, one node will never be used in any argument tree (i.e., X_2).

Now suppose we have the following argument as the root of the argument tree:

$\langle \{\gamma, \gamma \rightarrow \delta, \alpha\}, \delta \wedge \alpha \rangle$

We have supports for two canonical undercuts of the root (i.e., `FirstLevelSets` $= \{X_1, X_3\}$):

$X_1 \backslash \{\gamma, \gamma \rightarrow \delta, \alpha\} = \{\neg\alpha\}$

$X_3 \backslash \{\gamma, \gamma \rightarrow \delta, \alpha\} = \{\neg\delta\}$

Also we have (X_4, X_3) as an arc, and so we have an undercut to the second of the undercuts:

$$X_4 \backslash \{\neg\delta\} = \{\gamma, \neg\gamma \vee \delta\}$$

There are no more arcs to consider, and so the net result is the following argument tree:

$$\begin{tikzcd}
& \langle\{\gamma, \gamma \rightarrow \delta, \alpha\}, \delta \wedge \alpha\rangle & \\
\langle\{\neg\alpha\}, \Diamond\rangle \arrow[ur, "X_1"] & & \langle\{\neg\delta\}, \Diamond\rangle \arrow[ul, "X_3"'] \\
& & \langle\{\gamma, \neg\gamma \vee \delta\}, \Diamond\rangle \arrow[u, "X_4"]
\end{tikzcd}$$

Example 7.2.9 Let $\Delta = \{\alpha, \beta, \neg\beta, \neg\alpha \vee \beta\}$. Thus, $N = \{X_1, X_2\}$, where $(X_1, X_2) \in A$, for X_1 and X_2 as follows:

$$X_1 = \{\alpha, \neg\beta, \neg\alpha \vee \beta\}$$

$$X_2 = \{\beta, \neg\beta\}$$

Now consider the root of an argument tree $\langle\Phi, \alpha \wedge \beta\rangle$, where $\Phi = \{\alpha, \beta\}$. Here we have $X_1 \cap \Phi \neq \emptyset$ and $X_2 \cap \Phi \neq \emptyset$. Thus, both are candidates for forming supports for undercuts for the root argument:

$$X_1 \backslash \Phi = \{\neg\beta, \neg\alpha \vee \beta\}$$

$$X_2 \backslash \Phi = \{\neg\beta\}$$

However, from the above, we see that we have the following:

$$X_2 \backslash \Phi \subseteq X_1 \backslash \Phi$$

Hence, X_1 fails the condition on line 6 of the `FirstLevel` algorithm, whereas X_2 succeeds. Therefore, only X_2 is used to construct an undercut to the root. Also we have (X_1, X_2) as an arc, and so we have an undercut to the undercut. The net result is the following argument tree:

$$\begin{array}{c} \langle\{\alpha,\beta\},\alpha\wedge\beta\rangle \\ \uparrow \\ \langle\{\neg\beta\},\diamond\rangle \\ \uparrow \\ \langle\{\alpha,\neg\alpha\vee\beta\},\diamond\rangle \end{array}$$

Example 7.2.10 Let $\Delta = \{\gamma\wedge\alpha,\beta,\beta\to\neg\alpha,\neg\gamma\}$. Hence, $N = \{X_1, X_2\}$, where $(X_1, X_2) \in A$, for X_1 and X_2 as follows:

$$X_1 = \{\gamma\wedge\alpha,\beta,\beta\to\neg\alpha\}$$

$$X_2 = \{\gamma\wedge\alpha,\neg\gamma\}$$

For this, we get the following argument tree with root $\langle\{\gamma\wedge\alpha\},\alpha\rangle$:

$$\begin{tikzcd} & \langle\{\gamma\wedge\alpha\},\alpha\rangle & \\ \langle\{\beta,\beta\to\neg\alpha\},\diamond\rangle \arrow[ur, "X_1"] & & \langle\{\neg\gamma\},\diamond\rangle \arrow[ul, "X_2"'] \end{tikzcd}$$

Note, even though we have (X_1, X_2) in A, we do not get an undercut for $\langle\{\beta,\beta\to\neg\alpha\},\diamond\rangle$ using X_2 because of the following, which fails the condition on line 3 of the `Subcuts` algorithm:

$$\{\beta,\beta\to\neg\alpha\}\cap X_2 = \emptyset$$

For the same reason, we do not get an undercut for $\langle\{\neg\gamma\},\diamond\rangle$ using X_1.

Example 7.2.11 As another illustration, continuing example 7.2.7, suppose $\langle\{\alpha\vee\beta\},\alpha\vee\beta\rangle$ is the root of the tree. We have supports for two canonical undercuts of the root, which are as follows:

$$X_3\backslash\{\alpha\vee\beta\} = \{\neg\alpha,\neg\beta\}$$

$$X_4\backslash\{\alpha\vee\beta\} = \{\delta\wedge\neg\alpha,\neg\beta\}$$

We have (X_3, X_2), (X_3, X_4), and (X_4, X_1) as arcs, and so we have undercuts to undercuts as follows:

$$X_2\backslash\{\neg\alpha,\neg\beta\} = \{\alpha\}$$

$$X_4\backslash\{\neg\alpha,\neg\beta\} = \{\delta\wedge\neg\alpha,\alpha\vee\beta\}$$

$$X_1\backslash\{\delta\wedge\neg\alpha,\neg\beta\} = \{\alpha\}$$

$$X_3\backslash\{\delta\wedge\neg\alpha,\neg\beta\} = \{\neg\alpha,\alpha\vee\beta\}$$

Finally, for the above undercuts, we have (X_4, X_1), (X_2, X_1), and (X_3, X_2) as arcs, and so we have undercuts to undercuts as follows:

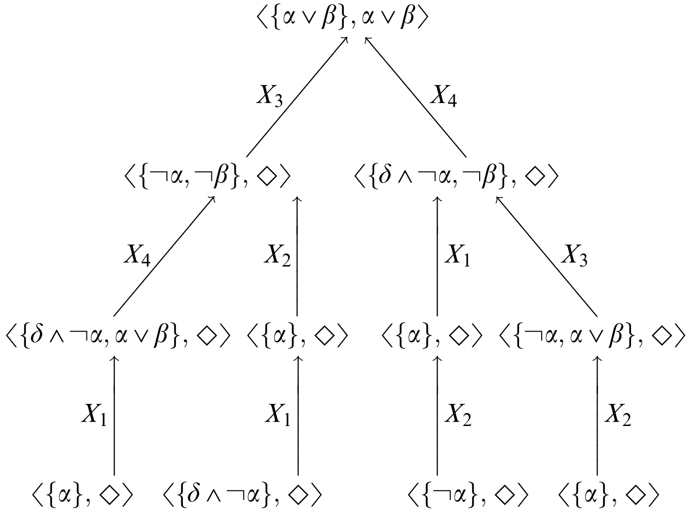
$$X_1 \backslash \{\delta \wedge \neg\alpha, \alpha \vee \beta\} = \{\alpha\}$$

$$X_1 \backslash \{\alpha\} = \{\delta \wedge \neg\alpha\}$$

$$X_2 \backslash \{\alpha\} = \{\neg\alpha\}$$

$$X_2 \backslash \{\neg\alpha, \alpha \vee \beta\} = \{\alpha\}$$

The net result is the following argument tree labeled with the element from N used for finding each undercut:



Example 7.2.12 This example represents an extreme case of inconsistency in a knowledgebase. Let $\Delta = \{\alpha \wedge \beta, \alpha \wedge \neg\beta, \neg\alpha \wedge \beta, \neg\alpha \wedge \neg\beta\}$, let $N = \{X_1, X_2, X_3, X_4, X_5, X_6\}$, and let $A = \{(X_1, X_2), (X_1, X_3), (X_1, X_4), (X_1, X_5), (X_2, X_3), (X_2, X_4), (X_2, X_6), (X_3, X_5), (X_3, X_6), (X_4, X_5), (X_4, X_6), (X_5, X_6)\}$, which is illustrated in figure 7.1, and where

$$X_1 = \{\alpha \wedge \beta, \alpha \wedge \neg\beta\} \qquad X_2 = \{\alpha \wedge \beta, \neg\alpha \wedge \beta\}$$

$$X_3 = \{\alpha \wedge \beta, \neg\alpha \wedge \neg\beta\} \qquad X_4 = \{\alpha \wedge \neg\beta, \neg\alpha \wedge \beta\}$$

$$X_5 = \{\alpha \wedge \neg\beta, \neg\alpha \wedge \neg\beta\} \qquad X_6 = \{\neg\alpha \wedge \beta, \neg\alpha \wedge \neg\beta\}$$

Let

$$A_1 = \langle\{\alpha \wedge \beta\}, \alpha \wedge \beta\rangle$$

$$A_2 = \langle\{\alpha \wedge \neg\beta\}, \alpha \wedge \neg\beta\rangle$$

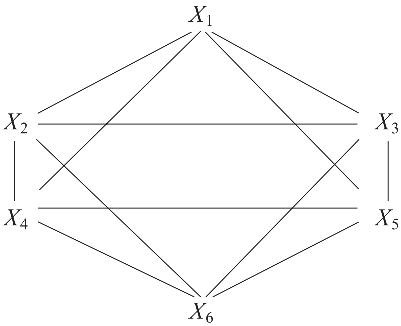
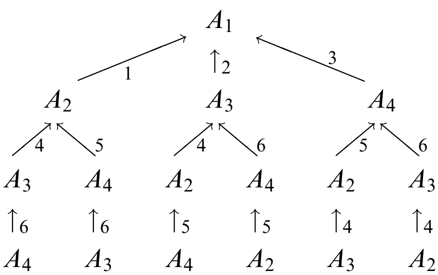


Figure 7.1
The compilation for example 7.2.12 with nodes $X_1, \ldots, X_6$.

$$A_3 = \langle \{\neg\alpha \wedge \beta\}, \neg\alpha \wedge \beta \rangle$$

$$A_4 = \langle \{\neg\alpha \wedge \neg\beta\}, \neg\alpha \wedge \neg\beta \rangle$$

Hence, we get the following argument tree. Each arc is labeled with the index i of the set $X_i \in N$ used to construct the undercut:



Assuming some knowledgebase Δ, the following completeness result shows that for any argument, we get all the canonical undercuts for it using the compilation of Δ.

Theorem 7.2.2 For a knowledgebase Δ, let Compilation$(\Delta) = (N, A)$ and let $\langle \Gamma, \alpha \rangle$ be an argument. For all $X \in N$ such that $\Gamma \subseteq X$, if $\langle \Psi, \diamond \rangle$ is an undercut for $\langle \Gamma, \alpha \rangle$ then

- either $\Psi = X \setminus \Gamma$
- or $\Psi = Y \setminus \Gamma$ for some $Y \in N$ such that $(X, Y) \in A$ and $Y \cap \Gamma \neq \emptyset$

For an argument $\langle \Phi, \alpha \rangle$, the `Undercuts` algorithm is correct in the sense that it constructs a complete argument tree with $\langle \Phi, \alpha \rangle$ as the root.

This is captured in theorem 7.2.3, and it is based on the completeness result in theorem 7.2.2 and the soundness results in propositions 7.2.2 and 7.2.5.

Theorem 7.2.3 For a knowledgebase Δ, a hypergraph $\mathsf{Compilation}(\Delta) = (N, A)$, and an argument $\langle \Phi, \alpha \rangle$, such that $\Phi \subseteq \Delta$, the algorithm `Undercuts`$(\langle \Phi, \alpha \rangle, N, A)$ returns the canonical undercuts for a complete argument tree T where the root of T is $\langle \Phi, \alpha \rangle$.

We can consider the `Undercuts` algorithm, with the subsidiary algorithms `FirstLevel` and `Subcuts`, as constructing a depth-first search tree, annotating each node (except the root) of the tree with an undercut and annotating each arc of the search tree with an arc from the hypergraph. In this way, the depth-first search tree is isomorphic with an argument tree rooted at $\langle \Phi, \alpha \rangle$.

While the `Undercuts` algorithm, with the subsidiary algorithms `FirstLevel` and `Subcuts`, construct a depth-first search tree from the compilation, they do not backtrack. Furthermore, in order to construct an argument tree, these algorithms only use constant time operations and tests, such as set membership, set union, and set intersection. In contrast, an algorithm for constructing an argument tree directly from a knowledgebase Δ has to search $\wp(\Delta)$ to find undercuts, and for each element $\Gamma \in \wp(\Delta)$, if an undercut with claim ψ is required, then the algorithm has to undertake the expensive tests of entailment (i.e., does $\Gamma \vdash \psi$ hold?), consistency (i.e., does $\Gamma \vdash \perp$ hold?), and minimality (i.e., does $\Gamma \cup \{\gamma_i\} \vdash \psi$ hold for each $\gamma_i \in \Gamma$?).

7.2.4 Constructing a Compilation

A compilation of a knowledgebase can be constructed by the `GenerateCompilation` algorithm, which in turn uses the `GenerateMinIncons` algorithm. These are defined below using the following subsidiary functions: `Subsets`(Δ, C), which returns the set of subsets of Δ that have cardinality C; `NotSuperSet`(Φ, Θ), which is a Boolean function that is true if there is no member of Θ that is a subset of Φ; and `Inconsistent`(Φ), which is a Boolean function that is true if $\Phi \vdash \perp$.

The `GenerateCompilation` algorithm takes the set of minimal inconsistent subsets of Δ and then seeks all pairs of minimal inconsistent subsets with nonempty intersections, hence giving the set of arcs of the graph.

Definition 7.2.4 Given a set of formulae Δ, the `GenerateCompilation` algorithm finds the compilation for Δ:

```
GenerateCompilation(Δ)
    let N = GenerateMinIncons(Δ)
    let M = N
    let A = ∅
    while M ≠ ∅
        remove an arbitrary element X from M
        let Temp = M
        while Temp ≠ ∅
            remove an arbitrary element Y from Temp
            if X ∩ Y ≠ ∅, then A = A ∪ {(X, Y)}
    return (N, A)
```

The `GenerateMinIncons` algorithm works by considering each subset of Δ as a possible minimal inconsistent subset of Δ. The search is started with the subsets of smallest cardinality, and then with each cycle of the outer while loop, the cardinality of the subsets is increased by one. The inner while loop checks each of the subsets Φ of a given cardinality. The if statement checks whether Φ is not a superset of a minimal inconsistent subset that has already been identified, and that Φ is inconsistent. If these conditions are met, then Φ is added to the set of minimal inconsistent subsets Θ that is eventually returned by the algorithm.

Definition 7.2.5 Given a set of formulae Δ, the `GenerateMinIncons` algorithm returns $\Theta \subseteq \wp(\Delta)$, which is the set of all the minimal inconsistent subsets of Δ:

```
GenerateMinIncons(Δ)
    let Θ = ∅
    let C = 1
    while C ≤ |Δ|
        Ω = Subsets(Δ, C)
        while Ω ≠ ∅
            let Φ be an arbitrary member of Ω
            if NotSuperSet(Φ, Θ) & Inconsistent(Φ)
            then Θ = Θ ∪ {Φ}
            Ω = Ω\{Φ}
        C = C + 1
    return Θ
```

We now turn to the complexity of constructing a compilation. We start by considering the potential size of $\mathsf{MinIncon}(\Delta)$. Let $\mathsf{Combinations}(x, y)$

be the number of combinations of subsets of cardinality y that can be formed from a set of cardinality x. In other words, $\text{Combinations}(x, y) = x!/(x-y)!y!$. Also, let $\lfloor z \rfloor$ be the greatest integer less than or equal to z.

Proposition 7.2.7 Let $n = |\Delta|$ and $m = \lfloor n/2 \rfloor$. If $\Delta \vdash \bot$, then $1 \leq |\text{MinIncon}(\Delta)| \leq \text{Combinations}(n, m)$.

While the above result shows that the cardinality of $\text{MinIncon}(\Delta)$ may be an issue, in practice it is likely this cardinality will be significantly lower than the upper limit and so be of an acceptable and manageable size.

Proposition 7.2.8 If $|\Delta| = n$, and $m = \lfloor n/2 \rfloor$, and $k = \text{Combinations}(n, m)$, and $\text{Compilation}(\Delta) = (N, A)$, then $|N| \leq k$ and $|A| \leq (k \times (k-1))$.

From this, the expensive part of compilation is the call `GenerateMinIncons`(Δ), which in the worst case involves 2^n tests where $|\Delta| = n$ and each test is of the following expensive form:

`NotSuperSet`(Φ, Θ) & `Inconsistent`(Φ)

In contrast, the algorithms for using the compilation are relatively inexpensive, involving only constant cost operations and tests such as set membership, set intersection, and set union. Hence, constructing a compilation is expensive, whereas the cost of using it is relatively inexpensive.

7.3 Argument Contouring

For constructing arguments, we need a minimal set of formulae that proves the claim. An automated theorem prover (ATP) may use a "goal-directed" approach, bringing in extra premises when required, but they are not guaranteed to be minimal. For example, suppose we have a knowledgebase $\{\alpha, \alpha \wedge \beta\}$ for proving $\alpha \wedge \beta$. The ATP may start with the premise α, and then, to prove β, a second premise is required, which would be $\alpha \wedge \beta$, and so the net result is $\{\alpha, \alpha \wedge \beta\} \vdash \alpha \wedge \beta$, which does not involve a minimal set of premises. In addition, an ATP is not guaranteed to use a consistent set of premises, since by classical logic it is valid to prove anything from an inconsistency.

Thus, if we seek arguments for a particular claim δ, we need to post queries to an ATP to ensure that a particular set of premises entails δ, that the set of premises is minimal for this, and that it is consistent. Therefore, finding arguments for a claim α involves considering subsets Φ of Δ and testing them with the ATP to ascertain whether $\Phi \vdash \alpha$ and

$\Phi \nvdash \perp$ hold. For $\Phi \subseteq \Delta$, and a formula α, let $\Phi?\alpha$ denote a call (a query) to an ATP. If Φ classically entails α, then we get the answer $\Phi \vdash \alpha$; otherwise we get the answer $\Phi \nvdash \alpha$. In this way, we do not give the whole of Δ to the ATP. Rather we call it with particular subsets of Δ. Thus, for example, if we want to know if $\langle \Phi, \alpha \rangle$ is an argument, then we have a series of calls $\Phi?\alpha$, $\Phi?\perp$, $\Phi\backslash\{\phi_1\}?\alpha, \dots, \Phi\backslash\{\phi_k\}?\alpha$, where $\Phi = \{\phi_1, \dots, \phi_k\}$. The first call is to ensure that $\Phi \vdash \alpha$, the second call is to ensure that $\Phi \nvdash \perp$, and the remaining calls are to ensure that there is no subset Φ' of Φ such that $\Phi' \vdash \alpha$.

We can now summarize all the queries that are posted to the ATP for finding all the arguments for α from the knowledgebase Δ. For this, we use the set $\mathsf{NaiveQuerying}(\Delta, \alpha)$ defined next.

Definition 7.3.1 For a knowledgebase Δ and a formula α,

$$\mathsf{NaiveQuerying}(\Delta, \alpha) = \{(\Phi?\alpha) \mid \Phi \subseteq \Delta\} \cup \{(\Phi?\perp) \mid \Phi \subseteq \Delta\}$$

Proposition 7.3.1 For Δ and α, if $|\Delta| = n$, then $|\mathsf{NaiveQuerying}(\Delta, \alpha)| = 2^{n+1}$.

Clearly, by using all the queries in $\mathsf{NaiveQuerying}(\Delta, \alpha)$, we are taking no account of any of the results that we have gained at any intermediate stage. In other words, we are not being intelligent. Yet if we want to harness automated reasoning, we need principled means for intelligently querying an ATP in order to search a knowledgebase for arguments.

When we think of the more difficult problem of not just finding all arguments for α, but then finding all undercuts to these arguments and, by recursion, undercuts to these undercuts, then there appears to be even more need for an intelligent strategy. During the course of building an argument tree, there will be repeated attempts to ask the same query, and there will be repeated attempts to ask a query with the same support set. However, importantly, each time a particular subset is tested for a particular claim, we gain more information about Δ. Therefore, instead of taking the naive approach of always throwing the result of querying away, we see that this information can be collated about subsets to help guide the search for further arguments and counterarguments. For example, if we know that a particular subset Φ is such that $\Phi \nvdash \alpha$, and we are looking for an argument with claim $\alpha \wedge \beta$, then we can infer that $\Phi \nvdash \alpha \wedge \beta$, and there is no need to test Φ for this claim (i.e., it is not useful to make the call $\Phi?\alpha \wedge \beta$).

Thus, as we undertake more tests of Δ for various subsets of Δ and for various claims, we build up a picture of Δ. We can consider these being stored as contours on the (cartographical) map of $\wp(\Delta)$. We formalize this in section 7.3.1.

To make our presentation more concise, from now on, we refer to each subset of Δ by a binary number. If we suppose Δ has cardinality n, then we adopt the arbitrary enumeration $\langle \alpha_1, \ldots, \alpha_n \rangle$ of Δ, presented in chapter 3. Using this, we can then represent any subset Φ of Δ by an n-digit binary number of the form $d_1, \ldots, d_n$. For the ith formula (i.e., α_i) in $\langle \alpha_1, \ldots, \alpha_n \rangle$, if α_i is in Φ, then the ith digit (i.e., d_i) in $d_1, \ldots, d_n$ is 1, and if α_i is not in Φ, then the ith digit (i.e., d_i) in $d_1, \ldots, d_n$ is 0. For example, for the enumeration $\langle \alpha, \beta \wedge \neg\gamma, \gamma \vee \varepsilon \rangle$, 000 is $\{\ \}$, 100 is $\{\alpha\}$, 010 is $\{\beta \wedge \neg\gamma\}$, 001 is $\{\gamma \vee \varepsilon\}$, 110 is $\{\alpha, \beta \wedge \neg\gamma\}$, 111 is $\{\alpha, \beta \wedge \neg\gamma, \gamma \vee \varepsilon\}$, and so forth.

Since we will be considering the power set of a knowledgebase, the following definition of the MaxWidth function will be useful for us: If n is $|\Delta|$ and m is $\lfloor n/2 \rfloor$ (i.e., m is the greatest integer less than or equal to $n/2$), then $\mathsf{MaxWidth}(n)$ is $n!/(n-m)!m!$. In other words, of all cardinalities for subsets of Δ, the most numerous are those of cardinality $\lfloor n/2 \rfloor$, and so the MaxWidth function gives the number of subsets of Δ of cardinality $\lfloor n/2 \rfloor$.

7.3.1 Framework for Contouring

We start with the ideal situation where we have substantial information about the knowledgebase Δ.

Definition 7.3.2 The **contour** for α is $\mathsf{C}(\alpha) = \{\Phi \subseteq \Delta \mid \Phi \vdash \alpha$ and there is no $\Psi \subset \Phi$ such that $\Psi \vdash \alpha\}$, the **uppercontour** for α is $\mathsf{U}(\alpha) = \{\Phi \subseteq \Delta \mid \Phi \vdash \alpha\}$, and the **lowercontour** for α is $\mathsf{L}(\alpha) = \{\Phi \subseteq \Delta \mid \Phi \nvdash \alpha\}$.

Clearly, $\mathsf{C}(\alpha) \subseteq \mathsf{U}(\alpha)$, and $\mathsf{L}(\alpha) \cup \mathsf{U}(\alpha)$ is $\wp(\Delta)$, and $\mathsf{L}(\alpha) \cap \mathsf{U}(\alpha)$ is $\emptyset$. Furthermore, it is possible that any set in $\mathsf{L}(\alpha)$, $\mathsf{C}(\alpha)$, or $\mathsf{U}(\alpha)$ is inconsistent. Obviously, all the sets in $\mathsf{U}(\perp)$ are inconsistent, and none in $\mathsf{L}(\perp)$ are inconsistent.

Proposition 7.3.2 For any $\Phi \subseteq \Delta$, $\Phi \in \mathsf{C}(\alpha)$ iff (1) for all $\Psi \in \mathsf{U}(\alpha)$, $\Psi \not\subset \Phi$, and (2) for all $\Psi \in \mathsf{L}(\alpha)$, $\Phi \not\subseteq \Psi$.

Example 7.3.1 For $\langle \alpha \wedge \neg\alpha, \beta, \neg\beta \rangle$, $\mathsf{U}(\perp) = \{100, 110, 101, 011, 111\}$, $\mathsf{C}(\perp) = \{100, 011\}$, $\mathsf{L}(\perp) = \{000, 010, 001\}$, $\mathsf{U}(\alpha \vee \beta) = \{100, 010, 101, 110, 011, 111\}$, $\mathsf{C}(\alpha \vee \beta) = \{100, 010\}$, and $\mathsf{L}(\alpha \vee \beta) = \{000, 001\}$.

We can use contours directly to generate arguments. For this, we obtain $\mathsf{L}(\perp)$ from $\mathsf{C}(\perp)$ as follows: $\mathsf{L}(\perp) = \{\Phi \subseteq \Delta \mid$ if $\Psi \in \mathsf{C}(\perp)$, then $\Psi \not\subseteq \Phi\}$.

Proposition 7.3.3 For any Φ and α, $\Phi \in \mathsf{C}(\alpha) \cap \mathsf{L}(\perp)$ iff $\langle \Phi, \alpha \rangle$ is an argument.

This means that if we have $\mathsf{C}(\alpha)$ and $\mathsf{C}(\perp)$, we have all the knowledge we require to generate all arguments for α. By this, we mean that we do not need to use the ATP.

However, we may be in a position where we have some γ for which we want to generate arguments but for which we do not have $\mathsf{C}(\gamma)$. In general, we cannot expect to have a contour for every possible claim for which we may wish to construct an argument. To address this, we will require an ATP, but we can focus our search for the arguments, so that we can reduce the number of calls that we make to the ATP. For this, we can identify Boolean constituents of γ for which we do have the contours. We start by considering conjunction.

Definition 7.3.3 For any α, β, the **contour conjunction** operator, denoted $\oplus$, is defined as follows: $\mathsf{C}(\alpha) \oplus \mathsf{C}(\beta) = \{\Phi \cup \Psi \mid \Phi \in \mathsf{C}(\alpha) \text{ and } \Psi \in \mathsf{C}(\beta)\}$.

The set $\mathsf{C}(\alpha) \oplus \mathsf{C}(\beta)$ is a superset of $\mathsf{C}(\alpha \wedge \beta)$.

Example 7.3.2 Consider $\langle \alpha, \beta, \alpha \vee \beta \to \gamma, \alpha \vee \beta \to \delta \rangle$. Thus, $\mathsf{C}(\gamma) = \{1010, 0110\}$ and $\mathsf{C}(\delta) = \{1001, 0101\}$. Hence, $\mathsf{C}(\gamma) \oplus \mathsf{C}(\delta) = \{1011, 0111, 1111\}$. For comparison, $\mathsf{C}(\gamma \wedge \delta) = \{1011, 0111\}$.

From the following containment result, we see that if we are looking for an argument for $\alpha \wedge \beta$, we take the minimal sets (by using the $\subseteq$ relation) in $\mathsf{C}(\alpha) \oplus \mathsf{C}(\beta)$ that are consistent.

Proposition 7.3.4 For any α, β, $\mathsf{C}(\alpha \wedge \beta) \subseteq (\mathsf{C}(\alpha) \oplus \mathsf{C}(\beta)) \subseteq \mathsf{U}(\alpha \wedge \beta)$.

In a similar way to conjunction, we now consider disjunction.

Definition 7.3.4 For any α, β, the **contour disjunction** operator, denoted $\otimes$, is defined as follows: $\mathsf{C}(\alpha) \otimes \mathsf{C}(\beta) = \{\Phi \subseteq \Psi \mid \Psi \in \mathsf{C}(\alpha) \text{ or } \Psi \in \mathsf{C}(\beta)\}$.

Example 7.3.3 Consider $\langle \alpha, \beta, \alpha \vee \beta \to \gamma, \alpha \vee \beta \to \delta \rangle$. So $\mathsf{C}(\gamma) = \{1010, 0110\}$ and $\mathsf{C}(\delta) = \{1001, 0101\}$. Hence, $\mathsf{C}(\gamma) \otimes \mathsf{C}(\delta) = \{0000, 1000, 0100, 0010, 0001, 1010, 0110, 1001, 0101\}$. Note that $\mathsf{C}(\gamma \vee \delta) = \{1010, 0110, 1001, 0101\}$.

Example 7.3.4 Consider $\langle \alpha \vee \beta \vee \gamma, \neg\beta, \neg\gamma, \neg\neg\beta \rangle$. Thus, $\mathsf{C}(\alpha) = \{1110\}$ and $\mathsf{C}(\beta) = \{0001\}$. Hence, $\mathsf{C}(\alpha) \otimes \mathsf{C}(\beta) = \{0000, 1000, 0100, 0010, 0001, 1100, 1010, 0110, 1110\}$. Note that $\mathsf{C}(\alpha \vee \beta) = \{0001, 1010\}$.

From following containment result, we see that if we are looking for an argument for $\alpha \vee \beta$, we look in the set $\mathsf{C}(\alpha) \otimes \mathsf{C}(\beta)$: By querying the ATP, we seek the minimal elements Φ of $\mathsf{C}(\alpha) \otimes \mathsf{C}(\beta)$ for which $\Phi \vdash \alpha \vee \beta$ holds. However, with the disjunction contour, we are not guaranteed to find all the arguments for $\alpha \vee \beta$.

Proposition 7.3.5 For any α, β, $\mathsf{C}(\alpha) \cup \mathsf{C}(\beta) \subseteq \mathsf{U}(\alpha \vee \beta)$.

Now we consider an operator to facilitate the use of a contour $\mathsf{C}(\alpha)$ to find arguments with the claim $\neg\alpha$.

Definition 7.3.5 For any $X \subseteq \wp(\Delta)$, the **shift** operator, denoted $\div$, is defined as follows: $\div X = \{\Phi \subseteq \Delta \mid \text{for all } \Psi \in X, \Phi \not\subseteq \Psi \text{ and } \Psi \not\subseteq \Phi\}$.

Example 7.3.5 Consider $\langle \alpha \vee \beta, \neg\alpha, \neg\beta, \neg\gamma \wedge \delta, \neg\delta \wedge \beta \rangle$. Thus, $\mathsf{C}(\beta)$ is $\{11000, 00001\}$. Hence, $\div\mathsf{C}(\beta)$ is $\{10100, 10010, 10110, 01100, 01010, 01110, 00100, 00010, 00110\}$. By comparison, $\mathsf{C}(\neg\beta)$ is $\{00100, 00011\}$.

While $\div$ is, in a weak sense, a kind of complementation operator, properties such as $\div(\div\mathsf{C}(\alpha)) = \mathsf{C}(\alpha)$ do not hold. However, we do have the following useful property, which shows how given $\mathsf{C}(\alpha)$, we can use $\div\mathsf{C}(\alpha)$ to focus our search for an argument for $\neg\alpha$. Essentially, it says that for any $\Phi \subseteq \Delta$, if $\Phi \vdash \neg\alpha$ and $\Phi \nvdash \bot$, then Φ is not a subset or a superset of any $\Psi \subseteq \Delta$ such that $\Psi \vdash \alpha$ and $\Psi \nvdash \bot$.

Proposition 7.3.6 For any α, $(\mathsf{C}(\neg\alpha) \cap \mathsf{L}(\bot)) \subseteq (\div\mathsf{C}(\alpha) \cap \mathsf{L}(\bot))$.

We conclude this section by considering undercuts. For any undercut, the claim is the negation of the support of its parent, and the support of the parent is some subset of Δ. Suppose the support of the parent is $\{\delta_1, \ldots, \delta_k\}$, so the claim of any undercut is $\neg(\delta_1 \wedge \cdots \wedge \delta_k)$. This claim is equivalent to $\neg\delta_1 \vee \cdots \vee \neg\delta_k$. Thus, if we have contours for each of $\neg\delta_1, \ldots, \neg\delta_k$, we can focus our search for any these undercut in the space delineated by $\mathsf{C}(\neg\delta_1) \otimes \cdots \otimes \mathsf{C}(\neg\delta_k)$. Hence, we may consider keeping a contour for the negation of each element in Δ in order to be able to construct canonical undercuts from contours.

7.3.2 Partial Contours

Let Π be a set of answers to queries to the ATP. In other words, for a series of queries $(\Phi_1?\alpha_1), \ldots, (\Phi_n?\alpha_n)$, we have obtained a set of answers $\Pi = \{\pi_1, \ldots, \pi_n\}$, where for each $\pi_i \in \Pi$, the answer is either of the form $\Phi_i \vdash \alpha_i$ or of the form $\Phi_i \nvdash \alpha_i$. Using Π, we want to generate a partial contour $\mathsf{C}(\Pi, \alpha)$ for α. We will define it so that $\mathsf{C}(\Pi, \alpha)$ is a subset of $\mathsf{C}(\alpha)$ calculated on the basis of Π.

Definition 7.3.6 For Π and α, the **partial uppercontour**, denoted $\mathsf{U}(\Pi, \alpha)$, and the **partial lowercontour**, denoted $\mathsf{L}(\Pi, \alpha)$, are defined as follows:

$\mathsf{U}(\Pi, \alpha) = \{\Phi \subseteq \Delta \mid \Psi \subseteq \Phi \text{ and } (\Psi \vdash \alpha) \in \Pi\}$
$\mathsf{L}(\Pi, \alpha) = \{\Phi \subseteq \Delta \mid \Phi \subseteq \Psi \text{ and } (\Psi \nvdash \alpha) \in \Pi\}$

For $\Pi \subseteq \Pi'$, $\mathsf{U}(\Pi, \alpha) \subseteq \mathsf{U}(\Pi', \alpha)$, and $\mathsf{L}(\Pi, \alpha) \subseteq \mathsf{L}(\Pi', \alpha)$.

Definition 7.3.7 For Π, the **partial contour** for $\mathsf{C}(\Pi, \alpha)$, is defined as follows:

$\mathsf{C}(\Pi, \alpha) = \{\Phi \in \mathsf{U}(\Pi, \alpha) \mid \text{ for all } \phi \in \Phi, (\Phi \backslash \{\phi\}) \in \mathsf{L}(\Pi, \alpha)\}$

Thus, from Π, we construct the partial uppercontour and the partial lowercontour for some α, and then from these we can construct the partial contour for α.

Example 7.3.6 For $\langle \alpha, \neg\alpha, \beta \rangle$, let $\Pi = \{(111 \vdash \bot), (101 \nvdash \bot), (011 \nvdash \bot), (100 \nvdash \bot), (110 \vdash \bot), (010 \nvdash \bot)\}$. Hence, $\mathsf{C}(\Pi, \bot) = \{110\}$.

We use partial contours in the same way as we use full contours. The only proviso is that with partial contours, we have incomplete information about the knowledgebase. Thus, for example, since $\mathsf{C}(\Pi, \alpha) \subseteq \mathsf{C}(\alpha)$, we are not guaranteed to obtain all arguments for α from just using $\mathsf{C}(\Pi, \alpha)$. However, for any $\Phi \in \mathsf{C}(\Pi, \alpha)$, we know that Φ or any superset of Φ implies α, and any proper subset does not imply Φ. Therefore, any element in $\mathsf{C}(\Pi, \alpha)$ can have a profound effect on searching for arguments. We use the shift operator (definition 7.3.5) to formalize this next.

Proposition 7.3.7 If $\langle \Phi, \alpha \rangle$ is an argument, then $\Phi \in (\mathsf{C}(\Pi, \alpha) \cup \div\mathsf{C}(\Pi, \alpha))$.

Proposition 7.3.8 If $\Phi \in \mathsf{C}(\Pi, \alpha) \cap \mathsf{L}(\bot)$, then $\langle \Phi, \alpha \rangle$ is an argument.

Now we turn to the question of what the bounds are on the number of queries (i.e., size of Π) to build a contour. In the worst case, for any α, to ensure $\mathsf{C}(\Pi, \alpha) = \mathsf{C}(\alpha)$, it is necessary for Π to have 2^n queries where $\Pi = \{(\Phi?\alpha) \mid \Phi \subseteq \Delta\}$. However, we can take a more intelligent approach to decrease Π. For example, if we generate Π dynamically, once we have enough information from Π to add a particular $\Phi \subseteq \Delta$ to a particular contour, we have no need to seek supersets or subsets of Φ for that contour. Furthermore, we can envisage that the contours are built over time, as a by-product of using a knowledgebase, so each time the ATP is queried, the answer to the query is added to Π. As items are added to Π, the contours are constructed incrementally. We can therefore think of contours as being formed by a kind of lemma generation.

Finally, if we use partial contours, we do not even need to assume that the knowledgebase is fixed, since every partial contour of a knowledgebase Δ is a partial contour of a knowledgebase $\Delta \cup \Delta'$ for any Δ'.

7.3.3 Storing Contours

So far we have made a case for the value of using contours. Even in the worst case, they offer an improvement over naive searching for arguments and counterarguments. By maintaining appropriate contours, the numbers of calls to the ATP is always decreased. The downside to maintaining contours, or even partial contours, is the amount of information that has to be kept about the knowledgebase over time.

Proposition 7.3.9 For any α, if $|\Delta| = n$, then $1 \leq \mathsf{C}(\alpha) \leq \mathsf{MaxWidth}(n)$.

Thus, for example, if the cardinality of our knowledgebase is 10, we may, in the worst case, have a cardinality for $\mathsf{C}(\bot)$ of 252. This is not surprising, given that $\mathsf{C}(\bot)$ is the set of minimally inconsistent subsets of Δ. Of course, this is a worst case scenario, and it would indicate a remarkably inconsistent knowledgebase for which it would be difficult to imagine arising in the real world. Nonetheless, it raises the question of whether we could compromise on the information we retain from Π and yet still reduce the number of queries to the ATP. We address this next.

For any α, as the cardinality of $\mathsf{C}(\alpha)$ increases towards $\mathsf{MaxWidth}(n)$, where $|\Delta| = n$, the average size of the elements of $\mathsf{C}(\alpha)$ approaches $\lfloor n/2 \rfloor$. This suggests that a simple solution is that when a contour or partial contour is being constructed, and it appears to be too large to be manageable, it may be better to erase it and construct the elements of the contour as and when required, and furthermore, if it is sufficiently large, it is relatively straightforward to find them because they are likely to be subsets of Δ of cardinality of approximately $\lfloor n/2 \rfloor$.

As another alternative to storing all the precise information we have about the contours, given some Π, we propose using outline contours. To do this, we require the subsidiary notion of a k-**partition**: For a set $\Lambda = \{\Phi_1, \ldots, \Phi_n\}$, where $k \leq n$, a k-partition is a partitioning of Λ into k disjoint subsets of Λ such that the partition with the most members has at most one member more than the partition with the fewest members. Thus, a k-partition is as close as possible to a partitioning with partitions of equal size. Because a k-partition is a partitioning, each element of Λ is in exactly one partition. We also require the subsidiary notion of a **serial union**: For a set $\Lambda = \{\Phi_1, \ldots, \Phi_n\}$, where $k \leq n$, suppose $(\Lambda_1, \ldots, \Lambda_k)$ is a k-partitioning of Λ. Then the serial union of $(\Lambda_1, \ldots, \Lambda_k)$ is $(\cup(\Lambda_1), \ldots, \cup(\Lambda_k))$.

Definition 7.3.8 For a set of answers Π, a formula α, and a threshold $k \in \mathbb{N}$, an **outline contour** $\mathsf{C}(\Pi, \alpha, k)$ is defined as follows: If $|\mathsf{C}(\Pi, \alpha)| < k$, then $\mathsf{C}(\Pi, \alpha, k)$ is $\mathsf{C}(\Pi, \alpha)$; otherwise let $\mathsf{C}(\Pi, \alpha, k)$ be the serial union of a k-partitioning of $\mathsf{C}(\Pi, \alpha)$.

Thus, if $\mathsf{C}(\Pi, \alpha, k)$ is $\mathsf{C}(\Pi, \alpha)$, then we are keeping the explicit information we have about the contour. However, if $|\mathsf{C}(\Pi, \alpha)| > k$, then we merge sets in $\mathsf{C}(\Pi, \alpha)$ so that we never have more than k sets. In any case, we can recover the contours from each outline contour.

Example 7.3.7 For $\langle \alpha, \neg\alpha, \beta, \neg\beta, \gamma, \neg\gamma \rangle$, $\mathsf{C}(\perp) = \{110000, 001100, 000011\}$. Let Π be such that $\mathsf{C}(\Pi, \perp) = \mathsf{C}(\perp)$, and let $k = 2$. Thus, there are three possibilities for $\mathsf{C}(\Pi, \perp, k)$. One of them is $\{111100, 000011\}$.

Proposition 7.3.10 Let Π be such that $\mathsf{C}(\Pi, \alpha) = \mathsf{C}(\alpha)$. For any k, if $\langle \Phi, \alpha \rangle$ is an argument, then there is a $\Psi \in \mathsf{C}(\Pi, \alpha, k)$, such that $\Phi \subseteq \Psi$.

Thus, outline contours involve more queries to the ATP than contours, but less space is required to store them.

7.4 Approximate Arguments

We now turn to the potential for approximation when building argument trees. As we suggested in section 7.3, if we seek arguments for a particular claim δ, we need to post queries to an ATP to ensure that a set of premises entails δ, that the set of premises is minimal for this, and that it is consistent.

Thus, for example, if we want to know if $\langle \Phi, \alpha \rangle$ is an argument, then we have a series of calls $\Phi?\alpha$, $\Phi?\perp$, $\Phi\backslash\{\phi_1\}?\alpha, \ldots, \Phi\backslash\{\phi_k\}?\alpha$, where $\Phi = \{\phi_1, \ldots, \phi_k\}$. The first call is to ensure that $\Phi \vdash \alpha$ holds, the second call is to ensure that $\Phi \nvdash \perp$ holds, and the remaining calls are to ensure that there is no subset Φ' of Φ such that $\Phi' \vdash \alpha$ holds. Now if $\Phi \vdash \alpha$ holds, but some of the further calls fail (i.e., Φ is not minimal or it is inconsistent), we still have an "approximate argument." Rather than throwing this away, we can treat it as an intermediate finding and use it as part of an "approximate argument tree" that we can build with fewer calls to the ATP than building a complete argument tree, and this approximate argument tree can then be refined, as required, with the aim of getting closer to a complete argument tree. We formalize this next.

7.4.1 Framework for Approximation

An approximate argument is a pair $\langle \Phi, \alpha \rangle$ where $\Phi \subseteq \mathscr{L}$ and $\alpha \in \mathscr{L}$. This is a very general definition. It does not assume that Φ is consistent, or that it entails α, or that it is even a subset of the knowledgebase Δ.

To focus our presentation in this chapter, we will restrict consideration to a particular class of approximate arguments, namely, entailments. If $\Phi \subseteq \Delta$ and $\Phi \vdash \alpha$, then $\langle \Phi, \alpha \rangle$ is an **entailment**. Furthermore, we will consider some subclasses of entailments defined as follows: If $\langle \Phi, \alpha \rangle$ is an entailment, and there is no $\Phi' \subset \Phi$ such that $\Phi' \vdash \alpha$, then $\langle \Phi, \alpha \rangle$ is a **miniment**; and if $\langle \Phi, \alpha \rangle$ is an entailment, and $\Phi \nvdash \bot$, then $\langle \Phi, \alpha \rangle$ is an **altoment**. Each of these kinds of entailment is defined as a relaxation of the definition for an argument: The support of an entailment implies the consequent, but neither an entailment nor an altoment has a support that is necessarily a minimal set of assumptions for implying the consequent and neither an entailment nor a miniment is necessarily a consistent set of assumptions for implying the consequent.

Example 7.4.1 Let $\Delta = \{\alpha, \neg\alpha \vee \beta, \gamma, \delta, \neg\beta, \neg\gamma\}$. Thus, entailments for β include A_1, A_2, A_3, A_4, and A_5, of which A_1, A_3, and A_5 are altoments, A_2 and A_5 are miniments, and A_5 is a argument:

$$A_1 = \langle \{\alpha, \neg\alpha \vee \beta, \gamma, \delta\}, \beta \rangle$$

$$A_2 = \langle \{\gamma, \neg\gamma\}, \beta \rangle$$

$$A_3 = \langle \{\alpha, \neg\alpha \vee \beta, \gamma\}, \beta \rangle$$

$$A_4 = \langle \{\alpha, \neg\alpha \vee \beta, \gamma, \neg\gamma\}, \beta \rangle$$

$$A_5 = \langle \{\alpha, \neg\alpha \vee \beta\}, \beta \rangle$$

An altoment is a "potentially overweight proof" in the sense that there are more assumptions in the support than required for the consequent, and a miniment is a "minimal proof" in the sense that if any formula is removed from the support, there will be insufficient assumptions in the support for the consequent to be entailed.

Some simple observations that we can make concerning entailments include the following: (1) If $\langle \Gamma, \alpha \rangle$ is an altoment, then there is a $\Phi \subseteq \Gamma$ such that $\langle \Phi, \alpha \rangle$ is an argument; (2) if $\langle \Gamma, \alpha \rangle$ is an entailment, then there is a $\Phi \subseteq \Gamma$ such that $\langle \Phi, \alpha \rangle$ is a miniment; and (3) if $\langle \Phi, \alpha \rangle$ is a miniment, and $\langle \Phi, \alpha \rangle$ is an altoment, then $\langle \Phi, \alpha \rangle$ is an argument.

An **approximate undercut** for an approximate argument $\langle \Phi, \alpha \rangle$ is an approximate argument $\langle \Psi, \beta \rangle$ such that $\beta \vdash \neg(\wedge\Phi)$.

An **approximate tree** is a tree T where each node is an approximate argument from Δ. There are various kinds of approximate tree. Here we define three particular kinds: (1) An **entailment tree** for α is an approximate tree where each node is an entailment and the root is for α, (2) an

altoment tree for α is an approximate tree where each node is an altoment and the root is for α, and (3) an **annotated tree** for α is an approximate tree where each node is a argument and the root is for α. For these trees, we do not impose that the children of a node are approximate undercuts, but in the way we construct them, we will aim for this.

Example 7.4.2 In the following, T_1 is an entailment tree, T_2 is an altoment tree, and T_3 is a argument tree:

$$\begin{tikzcd}[column sep=small] & T_1 \quad \langle\{\beta, \neg\beta, \delta, \delta \vee \beta, (\beta \vee \delta) \rightarrow \alpha\}, \alpha\rangle & \\ \langle\{\phi, \neg\delta, \delta \vee \neg\beta\}, \neg\beta\rangle \arrow[ur] & & \langle\{\phi, \neg\phi\}, \neg\delta\rangle \arrow[ul] \end{tikzcd}$$

$$\begin{tikzcd}[column sep=small] & T_2 \quad \langle\{\beta, \delta, \delta \vee \beta, (\beta \vee \delta) \rightarrow \alpha\}, \alpha\rangle & \\ \langle\{\phi, \neg\delta, \delta \vee \neg\beta\}, \neg\beta\rangle \arrow[ur] & & \langle\{\phi, \neg\beta, \neg\delta \vee \beta\}, \neg\delta\rangle \arrow[ul] \end{tikzcd}$$

$$\begin{tikzcd} T_3 \quad \langle\{\beta, \beta \rightarrow \alpha\}, \alpha\rangle \\ \langle\{\gamma, \gamma \rightarrow \neg\beta\}, \neg(\beta \wedge (\beta \rightarrow \alpha))\rangle \arrow[u] \end{tikzcd}$$

Obviously, all argument trees are annotated trees, all annotated trees are altoment trees, and all altoment trees are entailment trees.

The following definition of refinement is a relationship that holds between some altoment trees. It holds when each altoment in T_2 uses the same number of or fewer assumptions than its corresponding altoment in T_1 and its claim is weaker or the same as its corresponding altoment in T_1. For this, let $\mathsf{Support}(\langle\Phi, \alpha\rangle)$ be Φ, and let $\mathsf{Claim}(\langle\Phi, \alpha\rangle)$ be α, where $\langle\Phi, \alpha\rangle$ is an entailment.

Definition 7.4.1 Let T_1 and T_2 be altoment trees. T_2 is a **refinement** of T_1 iff there is a bijection f from the nodes of T_1 to the nodes of T_2 such that for all nodes A in T_1, $\mathsf{Support}(f(A)) \subseteq \mathsf{Support}(A)$ and $\mathsf{Claim}(A) \vdash \mathsf{Claim}(f(A))$. We call f the **refinement function**.

Proposition 7.4.1 If T_2 is a refinement of T_1 with refinement function f, then for all $\langle\Phi_1, \alpha_1\rangle \in T_1$, $f(\langle\Phi_1, \alpha_1\rangle)$ is more conservative than $\langle\Phi_1, \alpha_1\rangle$.

Refinement is useful because we can build a tree using altoments and then refine those altoments as part of a process of obtaining a better approximate tree and, if required, a complete argument tree. We consider this process more generally for entailments in the next section.

7.4.2 Constructing Approximate Trees

To render the construction, and improvement, of approximate trees implementable, we define the **revision steps** that can be undertaken on a tree T as follows, where T' is the result of the revision step, and all entailments come from the knowledgebase Δ.

1. T' is obtained by **resupport** from T by taking an entailment $\langle \Phi, \alpha \rangle$ in T and removing one formula, say, ϕ, such that $\langle \Phi \backslash \{\phi\}, \alpha \rangle$ is an entailment.

2. T' is obtained by **reconsequent** from T by replacing an entailment $\langle \Phi, \alpha \rangle$ in T with entailment $\langle \Phi, \alpha' \rangle$ where $\langle \Psi, \beta \rangle$ is the parent of $\langle \Phi, \alpha \rangle$ in T and $\langle \Psi, \beta \rangle$ is the parent of $\langle \Phi, \alpha' \rangle$ in T' and $\alpha \not\equiv \alpha'$ and $\alpha' \equiv \neg(\psi_1 \wedge \cdots \wedge \psi_n)$ and $\Psi = \{\psi_1, \ldots, \psi_n\}$.

3. T' is obtained by **expansion** from T by taking an entailment $\langle \Phi, \alpha \rangle$ in T and adding an entailment $\langle \Psi, \beta \rangle$ such that $\langle \Psi, \beta \rangle$ is an approximate undercut of $\langle \Phi, \alpha \rangle$ and it has not been shown that $\Psi \vdash \bot$ holds and, if $\langle \Psi, \beta \rangle$ has ancestor nodes $\langle \Phi_1, \beta_1 \rangle, \ldots, \langle \Phi_n, \beta_n \rangle$, then Ψ is not a subset of $\Phi_1 \cup \cdots \cup \Phi_n$.

4. T' is obtained by **contraction** from T by removing an entailment $\langle \Psi, \beta \rangle$ (and all its offspring) such that $\langle \Psi, \beta \rangle$ is a miniment and $\Psi \vdash \bot$.

5. T' is obtained by **deflation** from T by removing an entailment $\langle \Psi, \beta \rangle$ (and all its offspring) such that $\langle \Psi, \beta \rangle$ is a child of $\langle \Phi, \alpha \rangle$ and $\Psi \cup \Phi \nvdash \bot$.

We explain the revision steps as follows: Resupport weakens the support of an entailment to remove an unnecessary premise, reconsequent strengthens the claim of an entailment so that it negates the conjunction of its parent's support, expansion adds a new approximate undercut to the tree (assuming that it has not been shown to have an inconsistent support), contraction removes a node that has become an inconsistent miniment (after previously being subject to resupport), and deflation removes a node with a support that is consistent with the support of its parent (after previously one or other being subject to resupport). Illustrations of using revision steps are given in examples 7.4.3 to 7.4.5 and figure 7.2.

Example 7.4.3 Each of the following three trees is an altoment tree. Furthermore, T_2 is a resupport of T_1 and T_3 is an expansion of T_2:

$$\langle\{\alpha, \alpha \to \beta, \delta\}, \alpha \wedge \beta\rangle$$

(3)
$$\begin{array}{c} \langle\{\alpha, \alpha \to \beta, \delta\}, \alpha \wedge \beta\rangle \\ \uparrow \\ \langle\{\alpha, \neg\alpha \vee \neg\beta\}, \neg\beta\rangle \end{array}$$

(3)
$$\begin{array}{ccc} & \langle\{\alpha, \alpha \to \beta, \delta\}, \alpha \wedge \beta\rangle & \\ \nearrow & & \nwarrow \\ \langle\{\alpha, \neg\alpha \vee \neg\beta\}, \neg\beta\rangle & & \langle\{\neg\alpha \vee \neg\delta, \delta\}, \neg\alpha\rangle \end{array}$$

(1)
$$\begin{array}{ccc} & \langle\{\alpha, \alpha \to \beta\}, \alpha \wedge \beta\rangle & \\ \nearrow & & \nwarrow \\ \langle\{\alpha, \neg\alpha \vee \neg\beta\}, \neg\beta\rangle & & \langle\{\neg\alpha \vee \neg\delta, \delta\}, \neg\alpha\rangle \end{array}$$

(2)
$$\begin{array}{ccc} & \langle\{\alpha, \alpha \to \beta\}, \alpha \wedge \beta\rangle & \\ \nearrow & & \nwarrow \\ \langle\{\alpha, \neg\alpha \vee \neg\beta\}, \neg(\alpha \wedge (\alpha \to \beta))\rangle & & \langle\{\neg\alpha \vee \neg\delta, \delta\}, \neg\alpha\rangle \end{array}$$

(1)
$$\begin{array}{ccc} & \langle\{\alpha, \alpha \to \beta\}, \alpha \wedge \beta\rangle & \\ \nearrow & & \nwarrow \\ \langle\{\neg\alpha \vee \neg\beta\}, \neg(\alpha \wedge (\alpha \to \beta))\rangle & & \langle\{\neg\alpha \vee \neg\delta, \delta\}, \neg\alpha\rangle \end{array}$$

(2)
$$\begin{array}{ccc} & \langle\{\alpha, \alpha \to \beta\}, \alpha \wedge \beta\rangle & \\ \nearrow & & \nwarrow \\ \langle\{\neg\alpha \vee \neg\beta\}, \neg(\alpha \wedge (\alpha \to \beta))\rangle & & \langle\{\neg\alpha \vee \neg\delta, \delta\}, \neg(\alpha \wedge (\alpha \to \beta))\rangle \end{array}$$

Figure 7.2
Let $\Delta = \{\alpha, \alpha \to \beta, \delta, \gamma, \neg\alpha \vee \neg\beta, \beta\}$. For trees $T_1, \ldots, T_7$ in the diagram, T_{i+1} is obtained from T_i by the revision step given in parentheses (i.e., 1 denotes resupport, 2 denotes reconsequent, and 3 denotes expansion).

$$T_1 \quad \begin{array}{c} \langle\{\alpha, \alpha \to \beta, \neg\gamma, \delta\}, \beta\rangle \\ \uparrow \\ \langle\{\gamma, \gamma \to \neg\alpha, \sigma\}, \neg\alpha\rangle \end{array} \qquad T_2 \quad \begin{array}{c} \langle\{\alpha, \alpha \to \beta, \delta\}, \beta\rangle \\ \uparrow \\ \langle\{\gamma, \gamma \to \neg\alpha, \sigma\}, \neg\alpha\rangle \end{array}$$

$$T_3 \quad \begin{array}{c} \langle\{\alpha, \alpha \to \beta, \delta\}, \beta\rangle \\ \uparrow \\ \langle\{\gamma, \gamma \to \neg\alpha, \sigma\}, \neg\alpha\rangle \\ \uparrow \\ \langle\{\neg\gamma\}, \neg\gamma\rangle \end{array}$$

Example 7.4.4 Each of the following trees is an altoment tree. Furthermore, T_2 is a contraction of T_1:

$$T_1 \quad \begin{array}{c} \langle\{\alpha, \alpha \to \beta\}, \beta\rangle \\ \uparrow \\ \langle\{\gamma, \neg\gamma\}, \neg(\alpha \wedge (\alpha \to \beta))\rangle \end{array} \qquad T_2 \quad \langle\{\alpha, \alpha \to \beta\}, \beta\rangle$$

Example 7.4.5 Each of the following trees is an altoment tree. Furthermore, T_2 is a resupport of T_1 and T_3 is a deflation of T_2:

$$T_1 \quad \begin{array}{c} \langle\{\delta, \alpha, \delta \wedge \gamma, \gamma \to \beta\}, \beta\rangle \\ \uparrow \\ \langle\{\neg\alpha\}, \neg(\delta \wedge \alpha \wedge (\delta \wedge \gamma) \wedge (\gamma \to \beta))\rangle \end{array}$$

$$T_2 \quad \begin{array}{c} \langle\{\delta, \delta \wedge \gamma, \gamma \to \beta\}, \beta\rangle \\ \uparrow \\ \langle\{\neg\alpha\}, \neg(\delta \wedge \alpha \wedge (\delta \wedge \gamma) \wedge (\gamma \to \beta))\rangle \end{array}$$

$$T_3 \quad \langle\{\delta, \delta \wedge \gamma, \gamma \to \beta\}, \beta\rangle$$

The next result shows that we can obtain a complete argument tree by some finite sequence of revision steps.

Proposition 7.4.2 If T_n is a complete argument tree for α, then there is a sequence of approximate trees for α, $\langle T_1, \ldots, T_n\rangle$, such that T_1 is an altoment tree with just a root node, and for each i, where $i < n$, T_{i+1} is obtained by a revision step from T_i.

Starting with an altoment tree for α that contains one node, and then using a sequence of revision steps to obtain a complete argument tree, does not necessarily offer any computational advantages over constructing a complete argument tree directly (by finding an argument for the root; then finding canonical undercuts to the root; and then, by recursion, finding canonical undercuts to the canonical undercuts). To revise an approximate tree involves calls to the ATP, and so the more we revise an approximate tree, the less there is an efficiency advantage over constructing a complete argument tree.

The real benefit of approximate argumentation is that an intermediate tree (in the sequence in proposition 7.4.2) tends to be more informative (than a partially constructed argument tree), since it tends to have more nodes and, thereby, better indicates the range of potential conflicts arising in Δ. Thus, in comparison with an argument tree, an approximate tree is less cautious (it compromises on the correctness of the arguments used); is less expensive (it uses fewer calls to an ATP, and each call made would also be made to construct each argument in the corresponding argument tree); and, in a sense, is more informative (in reporting on potential conflicts in Δ).

To obtain an entailment tree, we use the following algorithm, which has an upper limit on the number of revision steps used.

Definition 7.4.2 The algorithm $\texttt{Generate}(\Delta, \Phi, \alpha, \lambda)$ returns an entailment tree that is constructed in a fixed number of cycles (delineated by the number λ), for a knowledgebase Δ, and an altoment $\langle \Phi, \alpha \rangle$ for the root of the entailment tree that is returned:

$\texttt{Generate}(\Delta, \Phi, \alpha, \lambda)$
    let T be the node $\langle \Phi, \alpha \rangle$
    let $\texttt{counter} = 0$
    while $\texttt{counter} \leq \lambda$ and there is
            a revision step T' of T
        let $T = T'$
        let $\texttt{counter} = \texttt{counter} + 1$
    return T

The aim of using the above algorithm is to start with an entailment for α as the root and then incrementally revise this entailment tree to get as close as possible to a complete argument tree within an acceptable number of iterations. The following shows that if we run the $\texttt{Generate}(\Delta, \Phi, \alpha, \lambda)$ algorithm for sufficient time, it will eventually result in a complete argument tree for α.

Proposition 7.4.3 Let $\langle T_1, \ldots, T_n \rangle$ be a sequence of entailment trees for α such that T_1 is an altoment tree with just a root node, and for each i, where $1 \leq i < n$, T_{i+1} is obtained by a revision step from T_i. If T_n is such that no further revision steps are possible on it, then T_n is a complete argument tree for α.

Corollary 7.4.1 For all propositional knowledgebases Δ and altoment $\langle \Phi, \alpha \rangle$, there is a $\lambda \in \mathbb{N}$ such that $\texttt{Generate}(\Delta, \Phi, \alpha, \lambda)$ returns an altoment tree T for α, and T is a complete argument tree for α.

We can improve the algorithm by incorporating selection criteria to prefer some revision steps over others. For example, we may prefer (1) an expansion that adds a node higher up the tree rather than lower down the tree, (2) resupport steps on nodes with large supports rather than those with small supports, and (3) as many expansion steps as possible in order to get an impression of as many as possible of the conflicts that potentially exist. By adopting selection criteria, we aim to have more meaningful (from a user's perspective) approximate trees returned given the threshold for the number of revision steps undertaken by the algorithm.

7.5 Discussion

In this chapter, we have provided some basic algorithms for argumentation and then attempted to address the computational viability problems by proposing compilation techniques, contouring, and approximate argumentation.

Much progress has been made on developing formalisms for argumentation. Some algorithms for argumentation have been developed (e.g., [KT99, CDM01, BG02, GS04]). However, relatively little progress has been made in developing techniques for overcoming the computational challenges of constructing arguments. Even though there are proposals for making DeLP more efficient by storing noninstantiated arguments in a database, again for reducing the number of calls to an ATP [CCS05], and by the use of prunning strategies [CSG05], there does appear to be a pressing need to look more widely for approaches for improving the efficiency of argumentation, and, in particular, there is a need for techniques appropriate for argumentation based on classical logic.

In the section on compilation, we have presented a solution to the inefficiency problem that involves compiling a knowledgebase Δ based on the set of minimal inconsistent subsets of Δ and then generating arguments from the compilation. While generating a compilation is expensive, generating arguments from a compilation is relatively inexpensive. Then, in the section on contouring, we have identified the need to manage the querying of an ATP when constructing arguments and counterarguments, introduced contours as a way of representing information about a knowledgebase obtained by querying with an ATP, and illustrated how using contours can be more efficient than naively using the knowledgebase with the ATP to search for arguments. Finally, in the section on approximation, we have presented a framework for approximate arguments that can serve as useful intermediate results when undertaking argumentation using an ATP. Finding approximate arguments requires fewer calls to an ATP, but it involves compromising the correctness of arguments.

7.6 Bibliographic Notes

Proof procedures and algorithms have been developed for finding preferred arguments from a knowledgebase following, for example, Dung's preferred semantics (see, e.g., [PS97, KT99, CDM01, DNT02, DMT02, DKT05, Vre06]). However, these techniques and analyses do not offer any ways of ameliorating the computational complexity inherent in finding arguments and counterarguments, even though it is a significant source of computational inefficiency. In this chapter, we have reviewed some approaches, namely, compilation [BH06], contouring [Hun06b], and approximate argumentation [Hun06a], that are intended to address these shortcomings.

Another possible approach, which may ameliorate the cost of entailment, is to use approximate entailment: Proposed in [Lev84], and developed in [SC95], classical entailment is approximated by two sequences of entailment relations. The first is sound but not complete, and the second is complete but not sound. Both sequences converge to classical entailment. For a set of propositional formulae Δ, a formula α, and an approximate entailment relation $\models_i$, the decision of whether $\Delta \models_i \alpha$ holds or $\Delta \not\models_i \alpha$ holds can be computed in polynomial time. Approximate entailment has been developed for anytime coherence reasoning [Kor01, Kor02]. However, the approach still needs to be further developed and evaluated for finding arguments and counterarguments in argumentation. This would need to start with a conceptualization of the notions of argument and counteragument derived using approximate entailment and approximate coherence.

Finally, another approximation approach is cost-bounded argumentation proposed for a form of probabilistic argumentation [Hae01]. For this, the cost is a function of the premises used for each argument, and this is justified in terms of probabilities associated with the formulae. In effect, the cost function biases which arguments are constructed in preference to others. This proposal may be adaptable for nonprobabilistic argumentation.

8 Comparison with Related Approaches

In this book, we have drawn out a number of the key elements of argumentation from a logical perspective. At the end of each chapter, we have indicated the literature that has led to the material covered in the chapter. However, we have left a more comprehensive comparison with other proposals in the literature until this chapter. We do not aim to be exhaustive here in our coverage of the literature. Rather, we review some of the key proposals and use this coverage to consider how our proposal, presented in this book, relates to the established literature. For general reviews of formalisms for argumentation, see [FKEG93, GK98, VLG98, PV02, CML00, CRL00].

Since there are many proposals for formalisms for logic-based argumentation, we consider two key dimensions that we use to characterize the diversity of these proposals. These dimensions are the underlying logic and the form of global warrant. We conceptualize these in general terms below and expand on them during the course of the chapter:

Underlying logic Most proposals for logic-based argumentation assume an underlying logic for generating an argument. For example, in our proposal in this book, an argument is a pair $\langle \Phi, \alpha \rangle$ where Φ is a minimal consistent subset of the knowledgebase that entails α, and for which the underlying logic used for consistency and entailment is classical logic. A number of other approaches to argumentation use classical logic for the underlying logic. Alternatives include various defeasible logics and default logic.

Warrant criteria Most proposals for logic-based argumentation incorporate criteria for deciding whether an argument is in some sense acceptable (believable, undefeated, etc.) given the other arguments that can be generated from the knowledgebase. Thus, for example, in our proposal in this book, we provide some judgment criteria that define when an argument is warranted. Informally, there are two classes of warrant criteria. The first is the fixpoint class, which have been characterized in abstract argumentation (chapter 2) graphs, and which we will see instantiated with logical arguments in this chapter. The second class is the recursive class, starting with the proposal by Pollock (see below) and adapted in numerous proposals, including our definition for judgment (definition 5.1.1).

In this chapter, we will consider four types of underlying logic for argumentation. In section 8.1, we consider classical logic as the underlying logic; in section 8.2, we consider various defeasible logics as the underlying logic; in section 8.3, we consider some inference rule systems as the underlying logic; and then in section 8.4, we consider various default logics as the underlying logic. We then switch to reviewing some tools developed for argumentation and briefly consider some application areas for argumentation in section 8.6.

Note that a number of terms, such as "argument," "undercut," "warranted," etc., are used by different authors in different ways; the definitions for terms are local to each proposal under consideration and so normally are used only in the subsection in which they are defined.

8.1 Argumentation Based on Classical Logic

In this section, we consider a series of proposals of logic-based argumentation where the underlying logic is classical logic. We start, in section 8.1.1, with the highly influential proposals by Pollock, who was a pioneer in the use of formal logic for conceptualizing argumentation in terms of arguments and counterarguments. Then, in section 8.1.2, we consider a number of proposals for reasoning with consistent subsets of a knowledgebase, and in section 8.1.3, developments of these that incorporate preference information. These proposals offer means for some simple comparisons between arguments. Next, in section 8.1.4, we review some developments, by Cayrol, of coherence reasoning for argumentation that instantiates the abstract argumentation notions of acceptability by Dung with logical arguments.

8.1.1 Argumentation as a Form of Defeasible Reasoning

The formalization of defeasible reasoning has its origins in philosophy, and it was originally developed for reasoning problems similar to those addressed by nonmonotonic logics in artificial intelligence. In [Pol87, Pol92], Pollock conceptualizes the notions of arguments, rebutting defeaters, and undercutting defeaters in terms of formal logic. Arguments can then be defined as chains of reasons leading to a conclusion with consideration of potential defeaters at each step. Different types of argument occur depending on the nature of the reasons and defeaters. This has provided a starting point for numerous proposals for logic-based argumentation.

Pollock aims at an argument-based approach to characterizing defeasible reasoning. Defeasibility arises from the fact that in the presence of further information, we may wish to withdraw an argument. For example, if a student has a runny nose and a sore throat, then we can infer that the student has a cold. However, if we also learn that the student has a temperature above 39 °C, then we may withdraw the inference that the student has a cold and instead infer that the student has flu. This kind of reasoning is a form of nonmonotonic reasoning [MH69, BDK97, Ant97].

Definition 8.1.1 An **argument** is a tuple of the form $\langle \Phi, \alpha \rangle$, where Φ is a set of premises, and α is a formula. In this review of the framework, we restrict consideration to arguments where $\Phi \vdash \alpha$ holds, which are called deductive arguments.

Definition 8.1.2 An argument $\langle \Psi, \beta \rangle$ **rebuts** an argument $\langle \Phi, \alpha \rangle$ iff $\vdash \beta \leftrightarrow \neg\alpha$. An argument $\langle \Psi, \beta \rangle$ **undercuts** an argument $\langle \Phi, \alpha \rangle$ iff $\vdash \beta \leftrightarrow \neg(\wedge\Phi)$. An argument A_1 **attacks** an argument A_2 iff A_1 rebuts A_2 or A_1 undercuts A_2.

For the above definition of attacks, Pollock uses the term "defeats." We substitute the term "attacks" to simplify the comparison with other approaches. In addition, "defeats" sounds decisive whereas "attacks" is neutral as to the outcome after taking the arguments into account, and so we believe that "attacks" is a better term.

In order to get to a characterization of what follows from a set of arguments (in other words, the warranted propositions that are justified by a set of arguments), we need to identify the undefeated arguments. When considering whether an argument A in a set of arguments $\mathcal{A}$ is undefeated, we need to consider whether any other arguments attack A, and then by recursion consider whether any arguments attack that attacker, and so on. Thus, the process proceeds in "levels," where at level 0, no argument is considered as provisionally defeated; at level 1, those arguments that are attacked are regarded as provisionally defeated; and at level 2, those argument that are attacked by arguments that have been provisionally defeated at level 1 are "reinstated," and so on.

The subset $\uparrow_i(\mathscr{A})$ of $\mathscr{A}$ denotes the arguments that are provisionally undefeated at level i, and the subset $\downarrow_i(\mathscr{A})$ of $\mathscr{A}$ denotes the arguments that are provisionally defeated at level i. Thus, $\uparrow_i(\mathscr{A})$ and $\downarrow_i(\mathscr{A})$ are disjoint, and their union is $\mathscr{A}$. The membership at any level i is given by the following definition.

Definition 8.1.3 Given a set of arguments $\mathscr{A}$, $\uparrow_0(\mathscr{A}) = \mathscr{A}$, and for $i > 0$,

$$\uparrow_{i+1}(\mathscr{A}) = \{X \in \mathscr{A} \mid X \text{ is not attacked by any } Y \in \uparrow_i(\mathscr{A})\}$$

For all i, let $\downarrow_i(\mathscr{A}) = \mathscr{A} \backslash \uparrow_i(\mathscr{A})$.

Definition 8.1.4 An argument X is **defeated** in $\mathscr{A}$ iff there is a level k such that for all $n \geq k$, $X \in \downarrow_n(\mathscr{A})$. An argument X is **undefeated** in $\mathscr{A}$ iff there is a level k such that for all $n \geq k$, $X \in \uparrow_n(\mathscr{A})$. Given an argument $\langle \Phi, \alpha \rangle \in \mathscr{A}$, the proposition α is **warranted** iff $\langle \Phi, \alpha \rangle$ is undefeated in $\mathscr{A}$.

Example 8.1.1 Consider the arguments $\mathscr{A} = \{A_1, A_2, A_3, A_4\}$, where A_4 attacks A_3, A_3 attacks A_2, and A_2 attacks A_1.

$$\begin{array}{ll}
\uparrow_0(\mathscr{A}) = \{A_1, A_2, A_3, A_4\} & \downarrow_0(\mathscr{A}) = \{\ \} \\
\uparrow_1(\mathscr{A}) = \{A_4\} & \downarrow_1(\mathscr{A}) = \{A_1, A_2, A_3\} \\
\uparrow_2(\mathscr{A}) = \{A_1, A_2, A_4\} & \downarrow_2(\mathscr{A}) = \{A_3\} \\
\uparrow_3(\mathscr{A}) = \{A_2, A_4\} & \downarrow_3(\mathscr{A}) = \{A_1, A_3\}
\end{array}$$

Since every level greater than 3 is the same as level 3, A_4 and A_2 are undefeated, and A_3 and A_1 are defeated.

Example 8.1.2 Consider the arguments $\mathscr{A} = \{A_1, A_2, A_3\}$, where A_2 attacks A_1 and A_2 attacks A_3 and A_3 attacks A_2:

$$\begin{array}{ll}
\uparrow_0(\mathscr{A}) = \{A_1, A_2, A_3\} & \downarrow_0(\mathscr{A}) = \{\ \} \\
\uparrow_1(\mathscr{A}) = \{\ \} & \downarrow_1(\mathscr{A}) = \{A_1, A_2, A_3\} \\
\uparrow_2(\mathscr{A}) = \{A_1, A_2, A_3\} & \downarrow_2(\mathscr{A}) = \{\ \} \\
\vdots & \vdots
\end{array}$$

Since every even level is the same as level 0, and every odd level is the same as level 1, none of the arguments are defeated nor undefeated.

An alternative, but equivalent, way of obtaining the undefeated arguments from a set of arguments is to use the following notion of a defeat graph together with the marking scheme (defined below).

Definition 8.1.5 Let $\mathscr{A}$ be a set of arguments. A **defeat graph** for $\mathscr{A}$ is a graph where each argument in $\mathscr{A}$ is a node in the graph and the arcs are such that for $X, Y \in \mathscr{A}$, (X, Y) is an arc iff X attacks Y. We describe X as a child of Y in the graph.

Definition 8.1.6 (Marked defeat graph) Each node in a defeat graph is marked with one of D (defeated), U (undefeated), and P (provisionally defeated). Given a defeat graph for $\mathscr{A}$, the nodes as marked as follows: (1) A node is marked as U iff it has no children or all its children are marked as D, (2) a node is marked as D iff it has at least one child that is marked as U, and (3) any remaining node is marked as P.

Example 8.1.3 Consider the arguments $\mathscr{A} = \{A_1, A_2, A_3, A_4\}$; the following is a defeat graph where A_4 attacks A_3, A_3 attacks A_2, and A_2 attacks A_1:

$$A_4 \to A_3 \to A_2 \to A_1$$

In this graph, A_4 and A_2 are marked as U, and A_3 and A_1 are marked as D.

Example 8.1.4 For the arguments $\mathscr{A} = \{A_1, A_2, A_3\}$, the following is a defeat graph where A_2 attacks A_1, A_2 attacks A_3, and A_3 attacks A_2, and hence all nodes are marked as P.

$$\begin{array}{c} A_1 \\ \uparrow \\ A_2 \\ \uparrow\downarrow \\ A_3 \end{array}$$

From this short review of Pollock's approach to characterizing defeasible reasoning using arguments, it can be seen that a number of key ideas in argumentation theory were first formalized by him. The notion of argument in the format $\langle \Phi, \alpha \rangle$ has been used by most other logic-based approaches, though normally with the additional constraints that Φ is consistent and minimal for entailing α. His definitions for rebut and undercut have also been used in most other proposals, though there are various adaptations of the notion of undercut, such as the definition we have used for our proposal in this book. His definition for warrant, a recursive type of warrant, has also been used and adapted by a number of other proposals, such as within the framework by Simari and Loui that we discuss in section 8.2.2, and it is adapted by us for our judge function (definition 5.1.1).

His research program has encompassed a wider range of goals in cognitive science as well as artificial intelligence. This has led him to characterize other kinds of arguments within his framework, including inductive arguments and statistical arguments. These different kinds of argument are yet to be considered within subsequent logic-based proposals for argumentation, including ours.

Note that while this proposal was intended to capture aspects of defeasible reasoning, the underlying logic is classical logic. Later we will consider defeasible logics which incorporate a special connective, and corresponding proof theory, for defeasible implication. The idea of defeasible logics also came about to capture aspects of defeasible reasoning.

8.1.2 Coherence Reasoning as a Simple Form of Argumentation

One of the most obvious strategies for reasoning with an inconsistent knowledgebase is to reason with consistent subsets of the knowledgebase. We call this coherence reasoning. This is closely related to the approach of removing information from the knowledgebase that is causing an inconsistency. Here, we explore some of the issues relating these approaches in the context of classical proof theory.

In coherence reasoning, an argument is based on a consistent subset of an inconsistent set of formulae. Further constraints can be imposed on the consistent subset for it to be an allowed argument. This range of further constraints gives us a range of approaches to argumentation.

Definition 8.1.7 Let Δ be a knowledgebase, let $\vdash$ be the classical consequence relation, let $\mathsf{Con}(\Delta) = \{\Pi \subseteq \Delta | \Pi \nvdash \perp\}$ and $\mathsf{Inc}(\Delta) = \{\Pi \subseteq \Delta | \Pi \vdash \perp\}$:

$$\mathsf{MaxCon}(\Delta) = \{\Pi \in \mathsf{Con}(\Delta) | \forall \Phi \in \mathsf{Con}(\Delta) \Pi \not\subset \Phi\}$$

$$\mathsf{MinIncon}(\Delta) = \{\Pi \in \mathsf{Inc}(\Delta) | \forall \Phi \in \mathsf{Inc}(\Delta) \Phi \not\subset \Pi\}$$

$$\mathsf{Free}(\Delta) = \bigcap \mathsf{MaxCon}(\Delta)$$

Hence $\mathsf{MaxCon}(\Delta)$ is the set of maximally consistent subsets of Δ; $\mathsf{MinIncon}(\Delta)$ is the set of minimally inconsistent subsets of Δ; and $\mathsf{Free}(\Delta)$

is the set of information that all maximally consistent subsets of Δ have in common. We also have the following relationship:

$$\bigcap \mathsf{MaxCon}(\Delta) = \Delta - \bigcup \mathsf{MinIncon}(\Delta)$$

We can consider a maximally consistent subset of a knowledgebase as a "coherent" view on the knowledgebase. For this reason, the set $\mathsf{MaxCon}(\Delta)$ is important in many of the definitions presented in the rest of this section. Furthermore, we consider $\mathsf{Free}(\Delta)$, which is equal to $\bigcap \mathsf{MaxCon}(\Delta)$, as capturing all the "uncontroversial" information in Δ. In contrast, we consider the set $\bigcup \mathsf{MinIncon}(\Delta)$ as capturing all the inconsistent data in Δ.

Example 8.1.5 Let $\Delta = \{\alpha, \neg\alpha, \alpha \to \beta, \neg\alpha \to \beta, \gamma\}$. This gives two maximally consistent subsets, $\Phi_1 = \{\alpha, \alpha \to \beta, \neg\alpha \to \beta, \gamma\}$ and $\Phi_2 = \{\neg\alpha, \alpha \to \beta, \neg\alpha \to \beta, \gamma\}$. From this $\bigcap \mathsf{MaxCon}(\Delta) = \{\alpha \to \beta, \neg\alpha \to \beta, \gamma\}$, and there exists a single minimally inconsistent subset $\Psi = \{\alpha, \neg\alpha\}$.

A problem with using inferences from consistent subsets of an inconsistent knowledgebase is that we do not know whether the premises used to derive the inference conflict with the other formulae in the knowledgebase. To handle this problem, we can adopt the notion of an argument from a knowledgebase and a notion of acceptability of an argument. An argument is a subset of the database, together with an inference from that subset. Using the notion of acceptability, the set of all arguments can be partitioned into sets of (arguments of) different degrees of acceptability. This can then be used to define a class of consequence relations (see, e.g., [BDP93, EGH95]).

Definition 8.1.8 Let Δ be a knowledgebase. An **argument** from Δ is a pair, $\langle \Pi, \phi \rangle$, such that $\Pi \subseteq \Delta$ and $\Pi \vdash \phi$. An argument is consistent if Π is consistent. We denote the set of arguments from Δ as $\mathsf{An}(\Delta)$, where $\mathsf{An}(\Delta) = \{\langle \Pi, \phi \rangle \mid \Pi \subseteq \Delta \wedge \Pi \vdash \phi\}$. Γ is an argument set of Δ iff $\Gamma \subseteq \mathsf{An}(\Delta)$.

Definition 8.1.9 Let Δ be a knowledgebase. Let $\langle \Pi, \phi \rangle$ and $\langle \Theta, \psi \rangle$ be any arguments constructed from Δ. If $\vdash \phi \leftrightarrow \neg\psi$, then $\langle \Pi, \phi \rangle$ is a **rebutting defeater** of $\langle \Theta, \psi \rangle$. If $\gamma \in \Theta$ and $\vdash \phi \leftrightarrow \neg\gamma$, then $\langle \Pi, \phi \rangle$ is an **undercutting defeater** of $\langle \Theta, \psi \rangle$.

Rebutting defeat, as defined here, is a symmetrical relation. One way of changing this is by use of priorities, such as in systems based on explicit representation of preference (e.g., [Bre89, CRS93, BDP95]) or as in systems based on specificity (e.g., [Poo88]).

For a knowledgebase Δ, an argumentative structure is any set of subsets of $\mathsf{An}(\Delta)$. The intention behind the definition for an argumentative structure is that different subsets of $\mathsf{An}(\Delta)$ have different degrees of acceptability. Below, we present one particular argumentative structure, denoted A^*, and then explain how the definition captures notions of acceptability.

Definition 8.1.10 The following sets constitute the argumentative structure A^*, where Δ is a knowledgebase and $\mathsf{A}\exists(\Delta) = \{\langle\Pi, \phi\rangle \mid \Pi \in \mathsf{Con}(\Delta) \wedge \Pi \vdash \phi\}$:

$$\mathsf{AT}(\Delta) = \{\langle\emptyset, \phi\rangle \mid \emptyset \vdash \phi\}$$

$$\mathsf{AF}(\Delta) = \{\langle\Pi, \phi\rangle \in \mathsf{A}\exists(\Delta) \mid \Pi \subseteq \mathsf{Free}(\Delta)\}$$

$$\mathsf{AB}(\Delta) = \{\langle\Pi, \phi\rangle \in \mathsf{A}\exists(\Delta) \mid (\forall\Phi \in \mathsf{MaxCon}(\Delta), \psi \in \Pi\Phi \vdash \psi)\}$$

$$\begin{aligned}\mathsf{ARU}(\Delta) = \{\langle\Pi, \phi\rangle \in \mathsf{A}\exists(\Delta) \mid (\forall\Phi \in \mathsf{MaxCon}(\Delta)\Phi \nvdash \neg\phi)\\ \text{and } (\forall\Phi \in \mathsf{MaxCon}(\Delta), \psi \in \Pi\Phi \nvdash \neg\psi)\}\end{aligned}$$

$$\mathsf{AU}(\Delta) = \{\langle\Pi, \phi\rangle \in \mathsf{A}\exists(\Delta) \mid (\forall\Phi \in \mathsf{MaxCon}(\Delta), \psi \in \Pi\Phi \nvdash \neg\psi)\}$$

$$\mathsf{A}\forall(\Delta) = \{\langle\Pi, \phi\rangle \in \mathsf{A}\exists(\Delta) \mid (\forall\Phi \in \mathsf{MaxCon}(\Delta)\Phi \vdash \phi)\}$$

$$\mathsf{AR}(\Delta) = \{\langle\Pi, \phi\rangle \in \mathsf{A}\exists(\Delta) \mid (\forall\Phi \in \mathsf{MaxCon}(\Delta)\Phi \nvdash \neg\phi)\}$$

The naming conventions for the argument sets are motivated as follows. T is for the tautological arguments—that is, those that follow from the empty set of premises. F is for the free arguments (due to Benferhat et al. [BDP93]), which are the arguments that follow from the data that are free of inconsistencies. B is for the backed arguments—that is, those for which all the premises follow from all the maximally consistent subsets of the data. RU is for the arguments that are not subject to either rebutting or undercutting. U is for the arguments that are not subject to undercutting. $\forall$ is for the universal arguments (essentially due to Manor and Rescher [MR70], where it was called inevitable arguments), which are the arguments that follow from all maximally consistent subsets of the data. R is for the arguments that are not subject to rebutting. $\exists$ is for existential arguments (essentially due to Manor and Rescher [MR70]), which are the arguments with consistent premises.

The definitions for $\mathsf{A}\exists$, AF, AT should be clear. We therefore focus on the remainder. AR allows an argument $\langle\Pi, \phi\rangle$ only if there is no maximally consistent subset that gives $\neg\phi$. AU allows an argument $\langle\Pi, \phi\rangle$ only if, for all items ψ in Π, there is no maximally consistent subset that gives $\neg\psi$. ARU combines the conditions of the AR and AU. Notice that AR

$$\begin{array}{c} & \mathsf{AT}(\Delta) \\ & | \\ \mathsf{AF}(\Delta) = \mathsf{AB}(\Delta) = & \mathsf{ARU}(\Delta) & = \mathsf{AU}(\Delta) \\ & | \\ & \mathsf{A}\forall(\Delta) \\ & | \\ & \mathsf{AR}(\Delta) \\ & | \\ & \mathsf{A}\exists(\Delta) \\ & | \\ & \mathsf{An}(\Delta) \end{array}$$

Figure 8.1
Partial order on A* induced by $\subseteq$. Sets lower in the diagram are subsets of sets higher in the diagram.

and A$\forall$ have very similar definitions, with the only difference being "$\Phi \nvdash \neg\phi$" in AR versus "$\Phi \vdash \phi$" in A$\forall$. A similar remark applies to AU and AB. Therefore A$\forall$ and AB are strengthenings of AR and AU, respectively (i.e., "$\nvdash \neg\phi$" replaced with "$\vdash \phi$"). We summarize the relationship between these sets in figure 8.1. The main features to notice are that A* is a linear structure and that there is an equivalence of AF, AB, ARU, and AU.

Example 8.1.6 Consider $\Delta = \{\alpha, \neg\alpha\}$. Then, $\langle\{\alpha, \neg\alpha\}, \alpha \wedge \neg\alpha\rangle \in \mathsf{An}(\Delta)$, $\langle\{\alpha\}, \alpha\rangle \in \mathsf{A}\exists(\Delta)$, $\langle\{\alpha\}, \alpha \vee \beta\rangle \in \mathsf{AR}(\Delta)$, if $\beta \nvdash \alpha$ (notice that if β were replaced by a formula γ such that $\gamma \vdash \alpha$, then $\neg\alpha \vdash \neg\gamma$; hence, $\neg\alpha \vdash \neg\alpha \wedge \neg\gamma$, which is equivalent with $\neg\alpha \vdash \neg(\alpha \vee \beta)$, so that the argument would not be in $AR(\Delta)$), $\langle\{\ \}, \alpha \vee \neg\alpha\rangle \in \mathsf{A}\forall(\Delta)$. Furthermore, $\mathsf{A}\forall(\Delta) = \mathsf{AF}(\Delta) = \mathsf{AB}(\Delta) = \mathsf{ARU}(\Delta) = \mathsf{AU}(\Delta) = \mathsf{AT}(\Delta)$.

Example 8.1.7 As another example, consider $\Delta = \{\neg\alpha \wedge \beta, \alpha \wedge \beta\}$. Then, for $\Pi = \{\alpha \wedge \beta\}$, $\langle\Pi, \beta\rangle \in \mathsf{A}\exists(\Delta)$, $\langle\Pi, \beta\rangle \in \mathsf{AR}(\Delta)$, and $\langle\Pi, \beta\rangle \in \mathsf{A}\forall(\Delta)$. However, there is no $\Pi \subseteq \Delta$ such that $\langle\Pi, \beta\rangle \in \mathsf{AU}(\Delta)$, $\langle\Pi, \beta\rangle \in \mathsf{ARU}(\Delta)$, $\langle\Pi, \beta\rangle \in \mathsf{AB}(\Delta)$, or $\langle\Pi, \beta\rangle \in \mathsf{AF}(\Delta)$.

The types of argument given in definition 8.1.10 can be extended with further kinds of argument such as lex arguments [BDP93]. These are based on the largest of the maximal consistent subsets of the knowledgebase. The set of lex arguments is given as

$$\mathsf{AL}(\Delta) = \{(\Pi, \phi) | \Pi \in \mathsf{LEX}(\Delta) \wedge \Pi \vdash \phi\}$$

where

$\mathsf{LEX}(\Delta) = \{\Gamma \in \mathsf{MaxCon}(\Delta) \mid \forall \Gamma' \in \mathsf{MaxCon}(\Delta) |\Gamma| \geq |\Gamma'|\}$

The set of lex arguments does not necessarily coincide with that of universal arguments, nor is one a subset of the other in general.

8.1.3 Coherence Reasoning with Preferences

An extension of coherence reasoning is the introduction of preference orderings over formulae. In [Bre89], a knowledgebase Δ is a tuple $(\Delta_1, \ldots, \Delta_n)$ where each Δ_i is a set of classical formulae. Information in Δ_i is preferred to (or more certain than) information in Δ_j if $i < j$. Given a knowledgebase Δ, a preferred subtheory $\Phi \subseteq \Delta$ is obtained using the following definition:

$\Phi = \Phi_1 \cup \cdots \cup \Phi_n$ is a **preferred subtheory** of Δ
iff $\forall k\ (1 \leq k \leq n)\ \Phi_1 \cup \cdots \cup \Phi_k$
is a maximal consistent subset of $\Delta_1 \cup \cdots \cup \Delta_k$

Thus, to obtain a preferred subtheory of Δ, we have to start with any maximal consistent subset of Δ_1, add as many formulae from Δ_2 as consistently can be added, and continue the process with $\Delta_3, \ldots, \Delta_n$. Reasoning is then done using classical logic with a preferred subtheory, and hence arguments can be constructed from each preferred subtheory.

In [BDP95], another approach to reasoning from an inconsistent knowledgebase is proposed, where each knowledgebase Δ is partitioned into a sequence of disjoint subsets $\Delta_1, \ldots, \Delta_n$, and Δ_i is regarded as more certain than Δ_j if $i < j$ holds. From these subsets, the following sets can be formed for $i\ (1 \leq i \leq n)\ \Delta^i = \Delta_1 \cup \cdots \cup \Delta_i$. We can take $\mathsf{Free}(\Delta)$ as the intersection of all the maximally consistent subsets of Δ. An inference α follows from Δ iff there is a positive integer such that α is a classical inference from $\mathsf{Free}(\Delta^i)$. This integer provides a qualification on the inferences.

In [CRS93], preferences over individual formulae in a knowledgebase are used to generate preferences over subsets of the knowledgebase. For this, a partial preordering relation (i.e., a reflexive and transitive relation) is used to represent the relative preferences that someone may have over a set of formulae Δ. For example, someone may prefer a formula that is recognized as being more certain. Given a knowledgebase Δ, and a partial preordering relation $\geq$ over Δ, for $x, y \in \Delta$, $x \geq y$ denotes that x is more preferable than, or equally preferable to, y. A preference relation over $\wp(\Delta)$ is obtained from $(\Delta, \geq)$ using either the democratism principle or the elitism principle, defined as follows:

Democratism Let Φ and Ψ be two nonempty subsets of Δ. Φ is democratically preferred to Ψ, denoted $\Phi >_{demo} \Psi$, iff for any $\psi \in \Psi \backslash \Phi$, there is a $\phi \in \Phi \backslash \Psi$ such that $\phi \geq \psi$ and $\psi \not\geq \phi$.

Elitism Let Φ and Ψ be two nonempty subsets of Δ. Φ is elitistically preferred to Ψ, denoted $\Phi >_{elite} \Psi$, iff for all $\phi \in \Phi \backslash \Psi$, there is a $\psi \in \Psi \backslash \Phi$, $\phi \geq \psi$ and $\psi \not\geq \phi$.

By analogy with voting theory, democratism prefers that anything removed from a set is replaced by something better, and elitism prefers that anything kept in a set must be better than something removed. Democratism prefers maximal subsets while elitism prefers minimal subsets. Furthermore, the preferred subtheories of Δ (using the definition of [Bre89]) are democratically preferred subsets of a knowledgebase Δ. Further preference relations that fit into this approach are presented in [ACL96].

8.1.4 Preference-Based Argumentation Frameworks

Classical logic can be used to instantiate the abstract argumentation proposed by Dung (see chapter 2). In [AC02], Amgoud and Cayrol developed a framework for argumentation that used logical arguments of the form $\langle \Phi, \alpha \rangle$ where Φ is a minimal consistent subset of the knowledgebase that entails α, with the warrant criteria based on acceptability. Their framework supports any choice of underlying logic for generating arguments, but it is illuminating to focus on classical logic for the underlying logic.

Definition 8.1.11 Let $\langle \Phi, \alpha \rangle$ and $\langle \Psi, \beta \rangle$ be arguments:

1. $\langle \Phi, \alpha \rangle$ **rebuts** $\langle \Psi, \beta \rangle$ iff $\vdash \alpha \leftrightarrow \neg\beta$
2. $\langle \Phi, \alpha \rangle$ **undercuts** $\langle \Psi, \beta \rangle$ iff there exists $\gamma \in \Psi$ such that $\vdash \alpha \leftrightarrow \neg\gamma$

Example 8.1.8 Let $\Delta = \{\alpha, \neg\beta, \alpha \to \gamma, \gamma \to \beta, \neg\beta \to \delta\}$. For the following arguments, we get A_2 undercuts A_1 and A_3 rebuts A_2:

$A_1 = \langle \{\neg\beta, \neg\beta \to \delta\}, \delta \rangle$

$A_2 = \langle \{\alpha, \alpha \to \gamma, \gamma \to \beta\}, \beta \rangle$

$A_3 = \langle \{\neg\beta\}, \neg\beta \rangle$

A key feature of this framework is the use of explicit priorities over the knowledge. This allows arguments of different "force" or "strength" to be captured.

Definition 8.1.12 Let Δ be a set of classical formulae, and let $\geq_{pref}$ be a (partial or total) preordering on subsets of Δ. For two arguments $\langle \Phi_1, \alpha_1 \rangle$ and $\langle \Phi_2, \alpha_2 \rangle$ from Δ, the ordering $\geq_{pref}$ over arguments is defined as follows:

$$\langle \Phi_1, \alpha \rangle \geq_{pref} \langle \Phi_2, \alpha \rangle \quad \text{iff } \Phi_1 \geq_{pref} \Phi_2$$

Let $\langle \Phi_1, \alpha_1 \rangle >_{pref} \langle \Phi_2, \alpha_2 \rangle$ denote $\langle \Phi_1, \alpha_1 \rangle \geq_{pref} \langle \Phi_2, \alpha_2 \rangle$ and $\langle \Phi_2, \alpha_2 \rangle \not\geq_{pref} \langle \Phi_1, \alpha_1 \rangle$.

The ordering relations reviewed in section 8.1.3, $>_{elite}$ and $>_{demo}$, can be used as instances of $>_{pref}$.

The definition of argumentation frameworks is extended with preferences in the following definition. In this brief review, we will assume the set of arguments $\mathscr{A}$ is the set of arguments $\langle \Phi, \alpha \rangle$ where Φ is a minimal subset of the knowledgebase such that $\Phi \vdash \alpha$, and the attacks relation is defined in terms of the undercuts relation and the $>_{elite}$ given above.

Definition 8.1.13 A **preference-based argumentation framework** (PAF) is a tuple of the form $\langle \mathscr{A}, \mathscr{R}, pref \rangle$ where $\mathscr{A}$ is a set of arguments, $\mathscr{R}$ is a binary relation relationship between arguments, $\mathscr{R} \subseteq \mathscr{A} \times \mathscr{A}$, and *pref* is a partial or total preordering on $\mathscr{A} \times \mathscr{A}$. Let A, B be two arguments in $\mathscr{A}$. B attacks A iff $B\mathscr{R}A$ and $A \not>_{pref} B$.

Example 8.1.9 Consider the following arguments where $A_2 >_{elite} A_1$ and $A_3 >_{elite} A_2$:

$$A_1 = \langle \{\neg\beta, \neg\beta \rightarrow \delta\}, \delta \rangle$$

$$A_2 = \langle \{\alpha, \alpha \rightarrow \gamma, \gamma \rightarrow \beta\}, \beta \rangle$$

$$A_3 = \langle \{\neg\beta\}, \neg\beta \rangle$$

Since A_2 undercuts A_1 and A_3 rebuts A_2, we get A_2 attacks A_1 and A_3 attacks A_2.

Definition 8.1.14 Let $\langle \mathscr{A}, \mathscr{R}, pref \rangle$ be a PAF. Let A, B be two arguments of $\mathscr{A}$ such that $B\mathscr{R}A$. A defends itself against B (w.r.t. *pref*) iff $A >_{pref} B$. An argument A defends itself (w.r.t. *pref*) iff for all B, if $B\mathscr{R}A$, then $A >_{pref} B$. Let $\mathscr{C}_{\mathscr{R},pref}$ denote the set of arguments such that $A \in \mathscr{C}_{\mathscr{R},pref}$ iff A defends itself (w.r.t *pref*).

Example 8.1.10 Let $\langle \mathscr{A}, \mathscr{R}, pref \rangle$ be a PAF such that $\mathscr{A} = \{A, B, C, D, E\}$, $\mathscr{R} = \{(C, D), (D, C), (A, E)\}$, and $C >_{pref} D$; then $\mathscr{C}_{\mathscr{R},pref} = \{A, B, C\}$.

Definition 8.1.15 An argument A is defended by $S \subseteq \mathscr{A}$ (w.r.t. *pref*) iff for all $B \in \mathscr{A}$, if $B\mathscr{R}A$ and $A \not>_{pref} B$, then there exists $C \in S$ such that $C\mathscr{R}B$ and $B \not>_{pref} C$.

The acceptable arguments are the ones that defend themselves against their defeaters ($\mathscr{C}_{\mathscr{R},pref}$) and also the arguments that are defended (directly or indirectly) by the arguments of $\mathscr{C}_{\mathscr{R},pref}$. We can formalize this by drawing on the definition of acceptable arguments by Dung using the characteristic function $\mathscr{F}$, where for a set of arguments $S \subseteq \mathscr{A}$,

$$\mathscr{F}(S) = \{A \in \mathscr{A} \mid A \text{ is defended by } S\}$$

Thus, we have

$$\mathscr{F}(\emptyset) = \mathscr{C}_{\mathscr{R},pref}$$

Using this characteristic function, the following definition provides the semantics of a general PAF.

Definition 8.1.16 Let $\langle \mathscr{A}, \mathscr{R}, pref \rangle$ be a PAF such that each argument is defeated by finitely many arguments. The set of acceptable arguments of the PAF is as follows:

$$Acc_{\mathscr{R},pref} = \bigcup \mathscr{F}^{\,i>0}(\emptyset) = \mathscr{C}_{\mathscr{R},pref} \cup \left[\bigcup \mathscr{F}^{\,i\geq 1}(\mathscr{C}_{\mathscr{R},pref})\right]$$

We now focus on the PAFs $\langle \mathscr{A}(\Delta), rebut, pref \rangle$ and $\langle \mathscr{A}(\Delta), undercut, pref \rangle$, where $\mathscr{A}(\Delta)$ is the set of arguments that can be formed from Δ.

Example 8.1.11 Continuing example 8.1.8, consider the following arguments:

$$A_1 = \langle \{\neg\beta, \neg\beta \to \delta\}, \delta \rangle$$

$$A_2 = \langle \{\alpha, \alpha \to \gamma, \gamma \to \beta\}, \beta \rangle$$

$$A_3 = \langle \{\alpha, \alpha \to \gamma, \neg\beta\}, \neg(\gamma \to \beta) \rangle$$

A_2 undercuts A_1 and $A_2 >_{elite} A_1$, so A_1 is not elitistically preferred to A_2 and A_1 does not defend itself against A_2. Hence $A_1 \notin \mathscr{C}_{undercut,elite}$. A_3 undercuts A_2 and $A_3 >_{elite} A_2$, so A_3 defends A_1 against A_2. Moreover, $A_3 \in \mathscr{C}_{\mathscr{R},elite}$. Thus, $A_1 \in \mathscr{F}(\mathscr{C}_{undercut,elite})$. Hence, for the system $\langle \mathscr{A}(\Delta), undercut, elite \rangle$, A_1 is an acceptable argument, and so $A_1 \in Acc_{undercut,elite}$.

There is a proof theory counterpart to this semantics. This means that it is not necessary to calculate the whole of $Acc_{\mathscr{R},pref}$ to find acceptable arguments. This proof theory is based on the notion of a dialogue tree defined as follows. For this, we require the Depth function: For a node n, $\mathsf{Depth}(n)$ is the number of arcs on the branch to the root. See Appendix B for more details.

Definition 8.1.17 A **dialogue tree** for an argument A is a tree where each node is annotated with an argument such that (1) the root is annotated with A; (2) for each node n annotated with argument A_i, and for each argument A_j, if $\mathsf{Depth}(n)$ is odd, and A_j attacks A_i, and A_i does not attack A_j, and there is no ancestor node n'' such that n'' is annotated with A_j, then there is a child n' of n and n' is annotated with A_j; and (3) for each node n annotated with argument A_i, and for each argument A_j, if $\mathsf{Depth}(n)$ is even, and A_j attacks A_i, then there is a child n' of n and n' is annotated with A_j.

Each node in a dialogue tree is either marked D for defeated or U for undefeated. This is determined by the following definition.

Definition 8.1.18 (Marked dialogue tree) Let T be a dialogue tree. Each leaf is U. For each nonleaf node n, if there is a child of n that is marked U, then n is marked D; otherwise n is marked U.

Given a PAF, an argument A is an acceptable argument iff the dialogue tree with the root node annotated with A is undefeated (i.e., the root node is marked U).

Example 8.1.12 Let $\langle \mathscr{A}, \mathscr{R}, \mathit{pref} \rangle$ be a PAF such that

$$\mathscr{A} = \{A_0, A_{01}, A_{02}, A_{03}, A_{10}, A_{11}, A_{12}, A_{13}, A_{14}\}$$

$$\mathscr{R} = \{(A_{10}, A_0), (A_{01}, A_{10}), (A_{12}, A_{02}), (A_{02}, A_{10}), (A_{03}, A_{11}), (A_{11}, A_0), (A_{13}, A_{14}), (A_{14}, A_{13})\}$$

$$A_{03} >_{pref} A_{11}, \quad A_{11} >_{pref} A_0, \quad A_{01} >_{pref} A_{10}, \quad A_{10} >_{pref} A_0, \quad A_{12} >_{pref} A_{02}, \quad A_{02} >_{pref} A_{10}, \quad A_{13} >_{pref} A_{14}$$

The following is a dialogue tree for A_0. Since A_{01} and A_{03} belong to $\mathscr{C}_{\mathscr{R},pref}$, A_0 is acceptable:

$$\begin{tikzcd}
 & & A_0 & \\
 & A_{10} \arrow[ur] & & A_{11} \arrow[ul] \\
A_{01} \arrow[ur] & & A_{02} \arrow[ul] & A_{03} \arrow[u] \\
 & & A_{02} \arrow[u] &
\end{tikzcd}$$

While in this section we have considered classical logic as the underlying logic, we can consider other logics. Interesting possibilities include a defeasible logic or a paraconsistent logic. Later, we will consider another approach that also instantiates Dung's notion of abstract argumentation. It is by Prakken and Sartor (see section 8.2.3), and it is similar in some key respects to PAFs presented here (the formalization of warrant is also based on Dung's notion of acceptability, though the underlying logic is restricted to a particular defeasible logic).

8.1.5 Discussion of Argumentation Based on Classical Logic

The approaches just presented (i.e., section 8.1) are perhaps the easiest to compare with our proposal. Using classical logic as the underlying logic for constructing arguments simplifies the presentation of the notion of an argument and of a counterargument. In these proposals, an argument is a pair $\langle \Phi, \alpha \rangle$ where Φ is a set of formulae that classically entails α, which is a formula. Most proposals also assume that Φ is consistent and there is no subset of Φ that also entails α.

Our framework has obviously been influenced by some of the key developments discussed in this section. Like the vast majority of proposals for logic-based argumentation, our framework is based on some of the key formal concepts, such as defeater, rebutter, and undercut, developed by Pollock—though our framework for logical argumentation goes beyond existing proposals for argumentation based on classical logic by focusing on canonical undercuts.

However, our framework does not include consideration of priorities over formulae. Priorities can be a useful and intuitive way of capturing some natural examples of argumentation. It should be straightforward to extend our proposal with priority information. A good starting point for doing this would be to adapt the setup by Amgoud and Cayrol that we reviewed in section 8.1.4.

8.2 Argumentation Based on Defeasible Logic

There are a number of proposals for defeasible logics. The common feature for these logics is the incorporation of a defeasible implication into the language. A good starting point for considering defeasible logic is logic for defeasible reasoning (LDR), which we consider next.

8.2.1 Logic for Defeasible Reasoning

Defeasible logics such as LDR can also be used to formalize aspects of argumentation [Nut88]. The motivation for LDR was to capture defeasible reasoning with information that is normally correct but can sometimes be incorrect, and hence the impetus for its development is similar to that for the argument-based defeasible reasoning reviewed in section 8.1.1. While LDR was not proposed for explicitly capturing argumentation, there are ideas introduced in LDR that are precursors to important ideas introduced in argumentation formalisms. In particular, the ideas of absolute rules, defeasible rules, and defeater rules, and a notion of specificity of rules, have influenced developments in a number of argumentation formalisms.

Definition 8.2.1 The language of LDR is composed of a set of atoms and the connectives $\{\neg, \Rightarrow, \rightarrow, \rightarrow\!\!\!\!\times, \wedge\}$. Clauses are formed as follows, where β is a conjunct of literals, and α is a literal: (1) $\beta \rightarrow \alpha$ is an absolute rule, (2) $\beta \Rightarrow \alpha$ is a defeasible rule, and (3) $\beta \rightarrow\!\!\!\!\times \alpha$ is a defeater rule. A knowledgebase is a set of absolute rules, defeasible rules, and defeater rules. For an atom α, the complement of α is $\neg\alpha$, and the complement of $\neg\alpha$ is α.

The proof theory is a form of modus ponens that allows inference of the consequents of rules subject to the following criteria:

- An absolute rule cannot be defeated, and the consequent can be inferred when the antecedent is inferred.
- For a defeasible rule, the consequent can be inferred when the antecedent can be inferred, and the rule is not defeated. It is defeated when (1) the complement of the consequent is inferred from an absolute rule, (2) the complement of the consequent is inferred from a more specific defeasible rule, or (3) there is a more specific defeater rule that has its antecedent satisfied and has a consequent that is the complement of the defeasible rule.
- The defeater rules do not give inferences. They can only defeat defeasible inferences.

The specificity ordering over defeater rules is determined from the antecedents of the defeasible and defeater rules. One rule is more specific than another according to the following criterion: If the antecedent to the first rule can be used in conjunction with the remainder of the knowledgebase (i.e., the knowledgebase minus the two competing rules) to prove the antecedent of the other rule, but not vice versa, then the first rule is more specific.

Example 8.2.1 Assume a knowledgebase containing just $\beta \Rightarrow \alpha$ and $\beta \wedge \gamma \rightarrow\!\!\!\!\times \neg\alpha$. If we add the literal β, then we infer α. However, if we add the literals β and γ, then we do not infer α, nor do we infer $\neg\alpha$.

The intuition for specificity is that a rule with a more specific antecedent is more appropriate for a specialized situation than a more general rule. This is analogous to the use of the principle of specificity in using conditional probability statements [CH93].

Developments of LDR include ordered logic [VL90] and prioritized logic [CHS93], which generalized the notion of specificity to include explicit ordering or priority information over the defeasible rules. Using such explicit information means the language can be simplified to just defeasible rules. A more general framework, for a class of defeasible logics, together with algorithms, has been developed by Antoniou et al. [ABGM00].

While LDR does not develop an explicit notion of argument or counterargument, it is a valuable example of a logic with defeasible implication. Following the proposal of LDR, a number of proposals for argumentation systems have used a logic with a defeasible implication, that is, a defeasible logic, as the underlying logic. We consider three key examples of such argumentation systems in the rest of this section.

8.2.2 Defeasible Argumentation with Specificity-Based Preferences

The defeasible argument system of [SL92] extends classical formulae with defeasible (metalevel) rules of the form $\alpha \succ\!\!- \beta$, where α and β are propositional formulae. The consequence relation $\mid\!\sim$ is the classical consequence relation extended with modus ponens for the defeasible rules. Thus, if $\alpha \succ\!\!- \beta \in \Delta$, and $\Delta \mid\!\sim \alpha$ holds, then $\Delta \mid\!\sim \beta$ holds. This defeasible argumentation can be seen as a refinement of the formalization for argumentation we reviewed in section 8.1.1.

Definition 8.2.2 A knowledgebase is a pair (Δ, Λ) where Δ is a set of classical formulae that is consistent and Λ is a set of defeasible rules.

Definition 8.2.3 For a knowledgebase (Δ, Λ), an argument for α is a pair $\langle \Phi, \alpha \rangle$ where (1) $\Phi \subseteq \Lambda$, (2) $\Delta \cup \Phi \mid\!\sim \alpha$, (3) $\Delta \cup \Phi \not\mid\!\sim \perp$, and (4) there is no $\Phi' \subset \Phi$ such that $\Delta \cup \Phi' \mid\!\sim \alpha$.

The definition of an argument is a refinement of the Pollock definition, since it requires consistency and minimality, and it extends the Pollock definition with the extended language.

From a knowledgebase of classical (nondefeasible) formulae, plus defeasible (meta-level) rules, arguments "for" and "against" any particular inference can be constructed. Each argument is a proof tree, where each step in the proof tree is either an application of modus ponens or an application of universal instantiation. In case of conflict, "more informed" arguments are preferred, as illustrated by the following examples. For this, one argument is "more or equally informative" than another if the nondefeasible formulae used with the first argument, together with the defeasible formulae of the second argument, imply the consequent of the second argument.

Example 8.2.2 Consider $\langle\{\alpha \wedge \beta \succ\!\!- \gamma\}, \gamma\rangle$ as an argument for γ using the nondefeasible set $\{\alpha, \beta\}$ and $\langle\{\alpha \succ\!\!- \neg\gamma\}, \neg\gamma\rangle$ as an argument for $\neg\gamma$ using the nondefeasible set $\{\alpha\}$. Here the first nondefeasible set $\{\alpha, \beta\}$ can also be used to derive $\neg\gamma$ according to the second argument $\langle\{\alpha \succ\!\!- \neg\gamma\}, \neg\gamma\rangle$, but we cannot use the second nondefeasible set $\{\alpha\}$ to derive γ according to the first argument. In other words, $\{\alpha \wedge \beta \succ\!\!- \gamma\}$ with $\{\alpha\}$ does not give an argument for γ. Thus, the first argument is more informed than the second argument.

Example 8.2.3 Consider $\langle\{\alpha \succ\!\!- \neg\gamma\}, \neg\gamma\rangle$ as an argument for $\neg\gamma$ using the nondefeasible set $\{\alpha\}$ and $\langle\{\alpha \succ\!\!- \beta, \beta \succ\!\!- \gamma\}, \gamma\rangle$ as an argument for γ using either of the nondefeasible sets $\{\alpha\}$ or $\{\beta\}$. Although the nondefeasible set $\{\alpha\}$ can also be used to derive $\neg\gamma$ from the rule $\{\alpha \succ\!\!- \neg\gamma\}$ in the first argument, we cannot use the nondefeasible set $\{\beta\}$ from the second argument to derive $\neg\gamma$ using the rule in the first argument. In other words, $\{\alpha \succ\!\!- \neg\gamma\}$ with $\{\beta\}$ does not give an argument for $\neg\gamma$. Thus, the first argument is more informed than the second argument, because $\{\alpha\}$ can also be used to derive γ by the rules in the second argument.

We will not provide further details on this approach, since this framework for argument-based defeasible reasoning has been further refined in the DeLP framework, which we describe in detail in section 8.2.4.

8.2.3 Argument-Based Extended Logic Programming

The argument-based extended logic programming with defeasible priorities by Prakken and Sartor is based on a defeasible logic as the underlying logic [PS97]. A knowledgebase in this framework also incorporates a preference relation over formulae. The semantics for reasoning with sets of arguments in this framework is based on the argumentation semantics by Dung that we reviewed in chapter 2.

Definition 8.2.4 For any atom of classical logic α, then α and $\neg\alpha$ are **strong literals**. In the metalanguage, if γ is a strong literal, then $\bar{\gamma}$ is $\neg\gamma$, and $\overline{\neg\gamma}$ is γ. A **weak literal** is a formula of the following form $\sim\alpha$ where α is a strong literal. A **defeasible rule** is a formula where r is a name for the rule, $\alpha_0, \ldots, \alpha_m$ are strong literals, $\beta_0, \ldots, \beta_n$ are weak literals, and δ is a strong literal:

$$r : \alpha_0 \wedge \cdots \wedge \alpha_m \wedge \beta_0 \wedge \cdots \wedge \beta_n \Rightarrow \delta$$

A **strict rule** is a formula where r is a name for the rule, $\alpha_0, \ldots, \alpha_m$ are strong literals, and δ is a strong literal:

$$r : \alpha_0 \wedge \cdots \wedge \alpha_m \rightarrow \delta$$

The $\sim$ negation is a default negation in the sense that $\sim\alpha$ holds when there is a failure to proof α.

Definition 8.2.5 An **ordered theory** is tuple $(S, D, <)$ where S is a set of strict rules, D is a set of defeasible rules, and $<$ is a strict partial ordering over D.

Definition 8.2.6 An **argument** is a finite sequence $A = [r_0, \ldots, r_n]$ of rules such that (1) for every i $(0 \leq i \leq n)$, for every strong literal α_j in the antecedent of r_i, there is a $k < i$ such that α_j is the consequent of r_k, and (2) no two distinct rules in the sequence have the same consequent. For a sequence of rules T and an argument A, $A + T$ is the concatenation of A and T.

Condition 1 above assumes that there are three proof rules for chaining reasoning with sequences of rules: The first is a conjunction rule for conjoining strong literals, the second is modus ponens for the strict rules, and the third is the following form of defeasible modus ponens:

$$\frac{\alpha_0 \wedge \cdots \wedge \alpha_m \qquad r : \alpha_0 \wedge \cdots \wedge \alpha_m \wedge \beta_0 \wedge \cdots \wedge \beta_n \Rightarrow \delta}{\delta}$$

Thus, given a defeasible rule $r : \alpha_0 \wedge \cdots \wedge \alpha_m \wedge \beta_0 \wedge \cdots \wedge \beta_n \Rightarrow \delta$, if we can obtain each of $\alpha_0, \ldots, \alpha_m$ using the strict rules or from using defeasible modus ponens, then we obtain δ as an inference, albeit defeasible.

For any ordered theory Γ, $Arg(\Gamma)$ denotes the set of all arguments that can be formed from Γ.

Definition 8.2.7 For any argument A,

1. A is **strict** iff it does not contain any defeasible rule; it is defeasible otherwise.
2. An argument A' is a **(proper) subargument** of A iff A' is a (proper) subsequence of A.
3. A literal L is a **conclusion** of A iff L is the consequent of some rule in A.
4. A literal L is an **assumption** of A iff $\sim\overline{L}$ occurs in some rule in A.

Example 8.2.4 Consider the following argument.

$A = [r_1 :\rightarrow \alpha, r_2 : \alpha \wedge {\sim}\neg\beta \Rightarrow \gamma, r_3 : \gamma \wedge {\sim}\delta \Rightarrow \varepsilon]$

The conclusions from A are $\{\alpha, \gamma, \varepsilon\}$, the subarguments are the following, of which only the first two (i.e., [ ], and $[r_1]$) are strict:

$\{[\ ], [r_1], [r_1, r_2], [r_1, r_2, r_3]\}$

The assumptions for A are $\{\beta, \neg\delta\}$.

Definition 8.2.8 Let A_1 and A_2 be two arguments. A_1 **attacks** A_2 iff there are sequences S_1 and S_2 of strict rules such that $A_1 + S_1$ is an argument with conclusion L and $A_2 + S_2$ is an argument with a conclusion $\bar{L}$ or A_2 is an argument with an assumption $\bar{L}$.

Example 8.2.5 Given the following rules, the argument $[r_3]$ attacks the argument $[r_1]$ and $[r_3]$ attacks the argument $[r_1, r_2]$:

$r_1 :\Rightarrow \alpha \qquad r_2 : \alpha \Rightarrow \neg\beta \qquad r_3 :\Rightarrow \neg\alpha$

Example 8.2.6 Consider the following defeasible rules, and the strict rule $s_1 :\rightarrow \gamma$:

$r_1 :\Rightarrow \alpha \qquad r_2 : \alpha \Rightarrow \neg\beta \qquad r_3 : \gamma \Rightarrow \beta \qquad r_4 : \beta \Rightarrow \neg\alpha$

Some arguments, and their interrelationships, include the following:

$[r_1, r_2]$ attacks $[s_1, r_3]$

$[r_1, r_2]$ attacks $[s_1, r_3, r_4]$

$[s_1, r_3, r_4]$ attacks $[r_1]$

$[s_1, r_3, r_4]$ attacks $[r_1, r_2]$

Example 8.2.7 Given the following rules, we get $[s_1, r_1]$ attacks $[s_2, r_2]$.

$r_1 : \alpha \Rightarrow \beta \qquad r_2 : \gamma \Rightarrow \neg\beta \qquad s_1 :\rightarrow \alpha \qquad s_2 :\rightarrow \gamma$

Definition 8.2.9 An argument is **coherent** if it does not attack itself; otherwise it is **incoherent**.

Example 8.2.8 Each of the following arguments are incoherent:

$[r_1 :{\sim}\alpha \Rightarrow \beta, r_2 : \beta \Rightarrow \alpha]$

$[r_1 :\Rightarrow \alpha, r_2 : \alpha \Rightarrow \neg\alpha]$

Definition 8.2.10 For any argument A and a set of strict rules S such that $A + S$ is an argument with conclusion L, the set $R_L(A + S)$ of the defeasible rules relevant to L in the argument $A + S$ is as follows:

1. $R_L(A + S) = \{r_d\}$ iff A includes a defeasible rule r_d with consequent L
2. $R_{L_1}(A + S) \cup \cdots \cup R_{L_n}(A + S)$ iff A is defeasible and S includes a strict rule $r_s : L_1 \wedge \cdots \wedge L_n \rightarrow L$

Definition 8.2.11 For any two sets R and R' of defeasible rules, $R < R'$ iff for some $r \in R$ and all $r' \in R'$ it holds that $r < r'$.

Definition 8.2.12 Let A_1 and A_2 be two arguments. A_1 **rebuts** A_2 iff there are sequences S_1 and S_2 of strict rules such that $A_1 + S_1$ is an argument with conclusion L and $A_2 + S_2$ is an argument with conclusion $\overline{L}$ and $R_L(A_1 + S_1) \not< R_{\overline{L}}(A_2 + S_2)$.

Definition 8.2.13 Let A_1 and A_2 be two arguments. A_1 **undercuts** A_2 iff there is a sequence S_1 of strict rules such that $A_1 + S_1$ is an argument with conclusion L and A_2 is an argument with assumption $\overline{L}$.

Definition 8.2.14 Let A_1 and A_2 be two arguments:

A_1 **defeats** A_2 iff (1) A_1 is empty and A_2 is incoherent
or (2) A_1 undercuts A_2
or (3) A_1 rebuts A_2 and A_2 does not undercut A_1

Also, A_1 strictly defeats A_2 iff A_1 defeats A_2 and A_2 does not defeat A_1.

Example 8.2.9 Consider the following rules and assume that $<$ is $\emptyset$:

$r_1 : \sim\neg\alpha \Rightarrow \alpha \qquad r_2 : \Rightarrow \neg\alpha$

Thus, $[r_1]$ rebuts $[r_2]$, and $[r_2]$ undercuts $[r_1]$, and hence $[r_1]$ does not defeat $[r_2]$. However, $[r_2]$ strictly defeats $[r_1]$.

We now turn to the semantics for reasoning with this framework.

Definition 8.2.15 An argument A is **acceptable** with respect to a set of arguments X iff each argument defeating A is strictly defeated by an argument in X.

Definition 8.2.16 Let Γ be an ordered theory. The characteristic function of Γ is $F_\Gamma : \wp(Args(\Gamma)) \mapsto \wp(Args(\Gamma))$ such that $S \subseteq Args(\Gamma)$ and

$$F_\Gamma(S) = \{A \in Args(\Gamma) \mid A \text{ is acceptable with respect to } S\}$$

Definition 8.2.17 For an ordered theory Γ and an argument A,

A is **justified** w.r.t. Γ iff A is in the least fixpoint of F_Γ

A is **overruled** w.r.t. Γ iff A is attacked by an argument B
and B is a justified argument w.r.t Γ

A is **defensible** w.r.t. Γ iff A is neither justified nor overruled w.r.t. Γ

Example 8.2.10 Consider the following rules where $r_0 < r_3$:

$r_0 :\Rightarrow \alpha \qquad r_1 : \alpha \Rightarrow \beta \qquad r_2 :\sim\beta \Rightarrow \gamma \qquad r_3 :\Rightarrow \neg\alpha$

Some arguments from this knowledgebase include the following:

$A_0 = [r_0] \qquad A_1 = [r_0, r_1] \qquad A_2 = [r_2] \qquad A_3 = [r_3]$

From this, we obtain that A_0 and A_1 are overruled and A_2 and A_3 are justified.

We now consider the notion of a dialectical proof theory. The idea is that there are two players, P for proponent and O for opponent, and they take turns in a dialogue. Each turn involves putting forward an argument. This process results in the construction of a proof tree.

Definition 8.2.18 A **move** is a pair of the form (P_i, A_i) where $i > 0$, P_i is a player (one of O and P), and A_i is an argument. A dialogue is a finite nonempty sequence of moves such that the following conditions hold:

1. if i is odd, then P_i is P; otherwise P_i is O
2. if $P_i = P_j = P$ and $i \neq j$, then $A_i \neq A_j$
3. if $P_i = P$ and $i > 1$, then A_i is a minimal (w.r.t. set inclusion) argument strictly defeating A_{i-1}
4. if $P_i = O$, then A_i defeats A_{i-1}

Definition 8.2.19 A **dialogue tree** is a finite tree of moves such that each branch is a dialogue. Let T be a dialogue tree for which no further moves can be added (i.e., for each leaf there are no further moves that can be added). A player wins a branch of T (i.e., a dialogue) if that player made the last move (i.e., the move at the leaf). A player wins a dialogue tree iff it wins all branches of T.

Definition 8.2.20 An argument A is a **provably justified argument** iff there is a dialogue tree with A as its root and won by the proponent. A strong literal L is a **provably justified conclusion** iff it is a conclusion of a provably justified argument.

Example 8.2.11 Consider the following rules where $r_4 < r_1$ and $r_3 < r_6$:

$$r_1 :\Rightarrow \alpha \qquad r_2 : \alpha \Rightarrow \beta \qquad r_3 : \beta \Rightarrow \gamma$$
$$r_4 :\Rightarrow \neg\alpha \qquad r_5 : \neg\alpha \Rightarrow \delta \qquad r_6 : \delta \Rightarrow \neg\gamma$$

Now consider the following dialogue tree for which neither player can make further moves:

$$\begin{array}{c} (P_1, [r_1 :\Rightarrow \alpha, r_2 : \alpha \Rightarrow \beta, r_3 : \beta \Rightarrow \gamma]) \\ \uparrow \\ (O_1, [r_4 :\Rightarrow \neg\alpha, r_5 : \neg\alpha \Rightarrow \delta, r_6 : \delta \Rightarrow \neg\gamma]) \\ \uparrow \\ (P_2, [r_1 :\Rightarrow \alpha]) \end{array}$$

It is not possible for player O to add the argument $[r_4 :\Rightarrow \neg\alpha]$ because that argument does not defeat P_2, since $r_4 < r_1$. Thus, the argument at the root of this tree is provably justified, and hence it is an acceptable argument.

This semantics and proof theory coincide. Also it can be shown that all provably justified arguments are justified.

There is an extension of argument-based extended logic programming that supports reasoning about priorities. This means preference relations can be context dependent, and can even be the subject of argumentation. This is potentially important in modeling intelligent reasoning in general, and is particularly important in applications such as legal reasoning.

8.2.4 Defeasible Logic Programming

DeLP is a form of logic programming based on defeasible rules [GS04]. The language and proof theory incorporates a number of ideas from other proposals for defeasible reasoning (including LDR, defeasible argumentation with specificity-based priorities and argument-based extended logic programming), together with a number of design choices for extending these ideas, that lead to a formalism for viable implementations. Furthermore, software implementations of the theorem proving technology are available.

Definition 8.2.21 The **language** for DeLP is composed of the following types of formulae:

1. Fact: A ground literal.
2. Strict rule: A rule of the form $\alpha \leftarrow \beta_1, \ldots, \beta_n$ where α and $\beta_1, \ldots, \beta_n$ are ground literals.

3. Defeasible rule: A rule of the form $\alpha \prec \beta_1, \ldots, \beta_n$ where α and $\beta_1, \ldots, \beta_n$ are ground literals.

A **program** for DeLP is a tuple (Γ, Π, Δ) where Γ is a set of facts, Π is a set of strict rules, and Δ is a set of defeasible rules.

As a notational convenience in the examples, some rules will be represented in schematic form using variables. This is shorthand for the corresponding ground rules that can be obtained by grounding the variables using the constant symbols in the language.

Example 8.2.12 The following are facts:

$c(t) \qquad p(t) \qquad s(t)$

The following are strict rules:

$b(x) \leftarrow c(x) \qquad b(x) \leftarrow p(x) \qquad {\sim}f(x) \leftarrow p(x)$

The following are defeasible rules:

$f(x) \prec b(x) \qquad {\sim}f(x) \prec c(x) \qquad f(x) \prec c(x), s(x) \qquad n(x) \prec f(x)$

A ground literal is obtained, defeasibly, from a program Π by applying a form of modus ponens as captured in the following definition for defeasible derivation.

Definition 8.2.22 Let (Γ, Π, Δ) be a program. A **defeasible derivation** of a ground literal α from $\Gamma \cup \Pi \cup \Delta$, denoted $\Gamma \cup \Pi \cup \Delta \mid\!\sim \alpha$, consists of a finite sequence of ground literals $\gamma_1, \ldots, \gamma_n$, where γ_n is α, and for each $k \in \{1, \ldots, n\}$, the literal γ_k is either a fact in Γ or there is a rule (strict or defeasible) in $\Pi \cup \Delta$ with consequent γ_k, and the literals in the antecedent $\delta_1, \ldots, \delta_i$ are in the sequence $\gamma_1, \ldots, \gamma_{k-1}$.

Example 8.2.13 Continuing example 8.2.12, the sequence $c(t)$, $b(t)$, $f(t)$ is a defeasible derivation for the literal $f(t)$ using the rules $\{b(t) \leftarrow c(t), f(t) \prec b(t)\}$.

Definition 8.2.23 A set of facts and rules Σ is **contradictory** iff for some α, $\Sigma \mid\!\sim \alpha$ and $\Sigma \mid\!\sim {\sim}\alpha$.

The following definition of an argument is analogous to the definition we introduced in chapter 3 (i.e., definition 3.2.1), except the language, the consequence relation, and the notion of consistency have been substituted with those for DeLP. Otherwise, the definitions both use a minimal consistent set of formulae that prove the claim of the argument.

Definition 8.2.24 An **argument** in DeLP is a pair $\langle \Phi, \alpha \rangle$ such that

1. $\Phi \subseteq \Delta$
2. $\Gamma \cup \Pi \cup \Phi \mid\!\sim \alpha$
3. $\Gamma \cup \Pi \cup \Phi$ is not contradictory
4. there is no $\Phi' \subset \Phi$ such that $\Gamma \cup \Pi \cup \Phi' \mid\!\sim \alpha$

Example 8.2.14 Continuing example 8.2.12, arguments we can obtain include the following:

$\langle \{{\sim}f(t) \prec c(t)\}, {\sim}f(t) \rangle$
$\langle \{f(t) \prec b(t)\}, f(t) \rangle$
$\langle \{f(t) \prec c(t), s(t)\}, f(t) \rangle$

Definition 8.2.25 An argument $\langle \Phi, \alpha \rangle$ is a **subargument** of an argument $\langle \Phi', \alpha' \rangle$ if $\Phi \subseteq \Phi'$.

Definition 8.2.26 Let (Γ, Π, Δ) be a program. Two ground literals α and β **disagree** iff the set $\Gamma \cup \Pi \cup \{\alpha, \beta\}$ is contradictory.

Definition 8.2.27 An argument $\langle \Psi, \beta \rangle$ **attacks** an argument $\langle \Phi, \alpha \rangle$ iff there exists a subargument $\langle \Phi', \alpha' \rangle$ of $\langle \Phi, \alpha \rangle$ such that α' and β disagree.

Example 8.2.15 Continuing example 8.2.12, each of the following arguments attacks the other argument. This example of attacks can be regarded as a form of rebutting:

$\langle \{{\sim}f(t) \prec c(t)\}, {\sim}f(t) \rangle$
$\langle \{f(t) \prec b(t)\}, f(t) \rangle$

Example 8.2.16 Continuing example 8.2.12, the first of the following arguments attacks the second argument. This example of attacks can be regarded as a form of undercutting:

$\langle \{{\sim}f(t) \prec b(t)\}, {\sim}f(t) \rangle$
$\langle \{f(t) \prec b(t), n(t) \prec f(t)\}, n(t) \rangle$

In order to judge arguments, two further criteria are taken into account. The first is a form of specificity criterion, and the second takes into account explicit ranking information over rules in the program.

Definition 8.2.28 Let (Γ, Π, Δ) be a program. Let Λ be the set of all literals that have a defeasible derivation from the program. An argument

$\langle \Phi_1, \alpha_1 \rangle$ is **strictly more specific** than an argument $\langle \Phi_2, \alpha_2 \rangle$ if the following conditions hold:

1. For all $\Lambda' \subseteq \Lambda$, if $\Pi \cup \Lambda' \cup \Phi_1 \mathrel{|\!\sim} \alpha_1$ and $\Pi \cup \Lambda' \mathrel{|\!\not\sim} \alpha_1$, then $\Pi \cup \Lambda' \cup \Phi_2 \mathrel{|\!\sim} \alpha_2$
2. There exists $\Lambda' \subseteq \Lambda$, such that $\Pi \cup \Lambda' \cup \Phi_2 \mathrel{|\!\sim} \alpha_2$ and $\Pi \cup \Lambda' \mathrel{|\!\not\sim} \alpha_2$; then $\Pi \cup \Lambda' \cup \Phi_1 \mathrel{|\!\not\sim} \alpha_1$

Example 8.2.17 Continuing example 8.2.12, the first argument below is strictly more specific than the second, and the second is strictly more specific than the third:

$\langle \{f(t) \prec c(t), s(t)\}, f(t) \rangle$
$\langle \{{\sim}f(t) \prec c(t)\}, {\sim}f(t) \rangle$
$\langle \{f(t) \prec b(t)\}, f(t) \rangle$

We now augment the program with an ordering relation over defeasible rules. For a program $(\Gamma, \Pi, \Delta, >)$, and $\rho_1, \rho_2 \in \Delta$, $\rho_1 > \rho_2$ means that ρ_1 is preferred over ρ_2.

Definition 8.2.29 For a program $(\Gamma, \Pi, \Delta, >)$, an argument $\langle \Phi_1, \alpha_1 \rangle$ is **preferred** over $\langle \Phi_2, \alpha_2 \rangle$ if

1. there exists at least one rule $\rho_1 \in \Phi_1$, and one rule $\rho_2 \in \Phi_2$, such that $\rho_1 > \rho_2$,
2. and there is no rule $\rho_1' \in \Phi_1$, and no rule $\rho_2' \in \Phi_2$, such that $\rho_2' > \rho_1'$.

Example 8.2.18 Consider the following program

$b(x) \prec g(x) \qquad {\sim}b(x) \prec r(x) \qquad r(x) \prec i(x, y) \qquad g(a) \qquad i(a, s)$

with the following ranking:

$b(x) \prec g(x) < {\sim}b(x) \prec r(x)$

The first argument below is preferred over the second:

$\langle \{{\sim}b(a) \prec r(a), r(a) \prec i(a, s)\}, {\sim}b(a) \rangle$
$\langle \{b(a) \prec g(a)\}, b(a) \rangle$

We now consider defeaters and argument lines.

Definition 8.2.30 An argument $\langle \Phi_1, \alpha_1 \rangle$ is a **proper defeater** for $\langle \Phi_2, \alpha_2 \rangle$ at a literal α iff there exists a subargument $\langle \Phi, \alpha \rangle$ of $\langle \Phi_2, \alpha_2 \rangle$ such that $\langle \Phi_1, \alpha_1 \rangle$ attacks $\langle \Phi_2, \alpha_2 \rangle$ at α and $\langle \Phi_1, \alpha_1 \rangle > \langle \Phi, \alpha \rangle$.

Example 8.2.19 Continuing example 8.2.17, the first argument is a proper defeater for the second argument, and the second argument is a proper defeater for the third argument.

Definition 8.2.31 An argument $\langle \Phi_1, \alpha_1 \rangle$ is a **blocking defeater** for $\langle \Phi_2, \alpha_2 \rangle$ at a literal β iff there exists a subargument $\langle \Phi, \alpha \rangle$ of $\langle \Phi_2, \alpha_2 \rangle$ such that $\langle \Phi_1, \alpha_1 \rangle$ attacks $\langle \Phi_2, \alpha_2 \rangle$ at β and $\langle \Phi_1, \alpha_1 \rangle$ is unrelated by the preference order to $\langle \Phi, \alpha \rangle$, i.e., $\langle \Phi_1, \alpha_1 \rangle \not\succ \langle \Phi, \alpha \rangle$ and $\langle \Phi, \alpha \rangle \not\succ \langle \Phi, \alpha_1 \rangle$.

Example 8.2.20 Consider the program composed from the following set of formulae: $\{p(x) \prec q(x), {\sim}p(x) \prec r(x), p(n), r(n)\}$. For the arguments below, each is a blocking defeater of the other:

$\langle \{p(n) \prec q(n)\}, p(n) \rangle$

$\langle \{{\sim}p(n) \prec r(n)\}, r(n) \rangle$

Definition 8.2.32 An argument $\langle \Phi_1, \alpha_1 \rangle$ is a **defeater** for $\langle \Phi_2, \alpha_2 \rangle$ iff either

1. $\langle \Phi_1, \alpha_1 \rangle$ is a proper defeater for $\langle \Phi_2, \alpha_2 \rangle$
2. $\langle \Phi_1, \alpha_1 \rangle$ is a blocking defeater for $\langle \Phi_2, \alpha_2 \rangle$

Definition 8.2.23 Let $\langle \Phi_0, \alpha_0 \rangle$ be an argument obtained from a program. An **argument line** for $\langle \Phi_0, \alpha_0 \rangle$ is a sequence of arguments from the program

$\langle \Phi_0, \alpha_0 \rangle, \langle \Phi_1, \alpha_1 \rangle, \langle \Phi_2, \alpha_2 \rangle, \langle \Phi_3, \alpha_3 \rangle, \ldots$

where each argument $\langle \Phi_i, \alpha_i \rangle$ in the sequence $(i > 0)$ is a defeater of its predecessor $\langle \Phi_{i-1}, \alpha_{i-1} \rangle$.

Definition 8.2.34 For an argument line $\langle \Phi_0, \alpha_0 \rangle, \langle \Phi_1, \alpha_1 \rangle, \langle \Phi_2, \alpha_2 \rangle, \langle \Phi_3, \alpha_3 \rangle, \ldots$, the set of **supporting arguments** is $\{\langle \Phi_0, \alpha_0 \rangle, \langle \Phi_2, \alpha_2 \rangle, \ldots .\}$ and the set of **interfering arguments** is $\{\langle \Phi_1, \alpha_1 \rangle, \langle \Phi_3, \alpha_3 \rangle, \ldots\}$.

Definition 8.2.35 Let (Γ, Π, Δ) be a program. Two arguments $\langle \Phi_1, \alpha_1 \rangle$ and $\langle \Phi_2, \alpha_2 \rangle$ are **concordant** iff the set $\Gamma \cup \Pi \cup \Phi_1 \cup \Phi_2$ is noncontradictory. A set of arguments $\{\langle \Phi_1, \alpha_1 \rangle, \ldots, \langle \Phi_n, \alpha_n \rangle\}$ is concordant iff $\Gamma \cup \Pi \cup \Phi_1 \cup \cdots \cup \Phi_n$ is noncontradictory.

Example 8.2.21 Consider a program where $\langle \Phi_1, \alpha_1 \rangle$ defeats $\langle \Phi_0, \alpha_0 \rangle$, $\langle \Phi_2, \alpha_2 \rangle$ defeats $\langle \Phi_0, \alpha_0 \rangle$, $\langle \Phi_3, \alpha_3 \rangle$ defeats $\langle \Phi_1, \alpha_1 \rangle$, $\langle \Phi_4, \alpha_4 \rangle$ defeats $\langle \Phi_2, \alpha_2 \rangle$, and $\langle \Phi_5, \alpha_5 \rangle$ defeats $\langle \Phi_2, \alpha_2 \rangle$. Thus, there are several argument lines starting with $\langle \Phi_0, \alpha_0 \rangle$, of which three are listed below.

$\langle \Phi_0, \alpha_0 \rangle, \langle \Phi_1, \alpha_1 \rangle, \langle \Phi_3, \alpha_3 \rangle$

$\langle \Phi_0, \alpha_0 \rangle, \langle \Phi_2, \alpha_2 \rangle, \langle \Phi_4, \alpha_4 \rangle$

$\langle \Phi_0, \alpha_0 \rangle, \langle \Phi_2, \alpha_2 \rangle, \langle \Phi_5, \alpha_5 \rangle$

A potentially problematical situation is reciprocal defeaters, which occurs when a pair of arguments defeat each other. This is illustrated in the following example.

Example 8.2.22 Consider the program composed from the following set of formulae:

$\{(\delta \prec {\sim}\beta \wedge \gamma), (\beta \prec {\sim}\delta \wedge \alpha), ({\sim}\beta \prec \alpha), ({\sim}\delta \prec \alpha), \alpha, \gamma\}$

Now consider the following arguments:

$A_1 = \langle \{(\delta \prec {\sim}\beta \wedge \gamma), ({\sim}\beta \prec \alpha)\}, \delta \rangle$

$A_2 = \langle \{(\beta \prec {\sim}\delta \wedge \alpha), ({\sim}\delta \prec \alpha)\}, \beta \rangle$

Thus, A_1 is a proper defeater for A_2 and A_2 is a proper defeater for A_1. Note also that A_2 is strictly more specific than the subargument $\langle \{({\sim}\delta \prec \alpha)\}, {\sim}\delta \rangle$ and A_1 is strictly more specific than the subargument $\langle \{({\sim}\beta \prec \alpha)\}, {\sim}\beta \rangle$.

The existence of reciprocal defeaters leads to the potential for infinite chains of reasoning. This is something that is detected and avoided in DeLP via the following definition of acceptable argument line. Another potential problem is circular argumentation, which occurs when an argument structure is introduced again in an argument line to defend itself. Again, this is something that is detected and avoided in DeLP as part of the definition of acceptable argument line.

Definition 8.2.36 The argument line $\langle \Phi_0, \alpha_0 \rangle, \langle \Phi_1, \alpha_1 \rangle, \langle \Phi_2, \alpha_2 \rangle, \langle \Phi_3, \alpha_3 \rangle, \ldots$ is an **acceptable argument line** iff

1. The argument line is a finite sequence.
2. The set of supporting arguments is concordant.
3. The set of interfering arguments is concordant.
4. No argument $\langle \Phi_k, \alpha_k \rangle$ is a subargument of an argument $\langle \Phi_i, \alpha_i \rangle$ appearing earlier in the sequence (i.e., $i < k$).
5. For all i, such that the argument $\langle \Phi_i, \alpha_i \rangle$ is a blocking defeater for $\langle \Phi_{i-1}, \alpha_{i-1} \rangle$, if $\langle \Phi_{i+1}, \alpha_{i+1} \rangle$ exists, then $\langle \Phi_{i+1}, \alpha_{i+1} \rangle$ is a proper defeater for $\langle \Phi_i, \alpha_i \rangle$.

A literal α is warranted if there exists an undefeated argument $\langle \Phi, \alpha \rangle$. In order to determine whether $\langle \Phi, \alpha \rangle$ is undefeated, the defeaters for $\langle \Phi, \alpha \rangle$ are identified and then, by recursion, the defeaters to defeaters are identified. This leads to a tree structure, defined next.

Definition 8.2.37 A **dialectical tree** for an argument $\langle \Phi_0, \alpha_0 \rangle$ is a labeled tree as follows:

1. The root of the tree is labeled with $\langle \Phi_0, \alpha_0 \rangle$

2. Let n be a nonroot node of the tree labeled $\langle \Phi_n, \alpha_n \rangle$, and let the following be the sequence of labels on the path from the root to n:

$\langle \Phi_0, \alpha_0 \rangle, \ldots, \langle \Phi_n, \alpha_n \rangle$

Let $\langle \Psi_0, \beta_0 \rangle, \ldots, \langle \Psi_k, \beta_k \rangle$ be all the defeaters for $\langle \Phi_n, \alpha_n \rangle$. For each defeater, $\langle \Psi_i, \beta_i \rangle$, $(1 \leq i \leq k)$, such that the following argument line B is acceptable; then the node n has a child n_i labeled with $\langle \Psi_i, \beta_i \rangle$:

$[\langle \Phi_0, \alpha_0 \rangle, \ldots, \langle \Phi_n, \alpha_n \rangle, \langle \Psi_i, \beta_i \rangle]$

If there is no defeater for $\langle \Phi_n, \alpha_n \rangle$, or if there is no $\langle \Psi_i, \beta_i \rangle$ such that B is acceptable, then n is a leaf.

Thus, for each node labeled with an argument $\langle \Phi_k, \alpha_k \rangle$, if the defeaters of this argument are $\langle \Phi_{k_1}, \alpha_{k_1} \rangle, \ldots, \langle \Phi_{k_i}, \alpha_{k_i} \rangle$, then each child of this node is labeled with one of these defeaters such that all these defeaters are used as a label, and no defeater is used as a label for more than one child, and no other labels are used.

Example 8.2.23 Consider the program where $\Gamma = \{\varepsilon, \lambda, \gamma, \kappa, \delta, \pi, \psi\}$ and Δ is the following set of defeasible rules:

$$
\begin{array}{ccccc}
\alpha \mathrel{-\!\!<} \beta & {\sim}\beta \mathrel{-\!\!<} \varepsilon & {\sim}\beta \mathrel{-\!\!<} \gamma \wedge \phi & {\sim}\phi \mathrel{-\!\!<} \lambda & {\sim}\phi \mathrel{-\!\!<} \psi \wedge \mu \\
\beta \mathrel{-\!\!<} \gamma & {\sim}\beta \mathrel{-\!\!<} \gamma \wedge \delta & \phi \mathrel{-\!\!<} \psi & {\sim}\mu \mathrel{-\!\!<} \kappa & \mu \mathrel{-\!\!<} \pi
\end{array}
$$

From this program, we can obtain the following dialectical tree:

$$
\begin{tikzcd}
& \langle \{(\alpha \mathrel{-\!\!<} \beta), (\beta \mathrel{-\!\!<} \gamma)\}, \alpha \rangle & \\
\langle \{({\sim}\beta \mathrel{-\!\!<} \gamma \wedge \delta)\}, {\sim}\beta \rangle \arrow[ur] & \langle \{({\sim}\beta \mathrel{-\!\!<} \gamma \wedge \phi), (\phi \mathrel{-\!\!<} \gamma)\}, {\sim}\beta \rangle \arrow[u] & \langle \{({\sim}\beta \mathrel{-\!\!<} \varepsilon)\}, {\sim}\beta \rangle \arrow[ul] \\
\langle \{({\sim}\phi \mathrel{-\!\!<} \gamma \wedge \mu), (\mu \mathrel{-\!\!<} \pi)\}, {\sim}\phi \rangle \arrow[ur] & & \langle \{({\sim}\phi \mathrel{-\!\!<} \lambda)\}, {\sim}\phi \rangle \arrow[ul] \\
\langle \{({\sim}\mu \mathrel{-\!\!<} \kappa)\}, {\sim}\mu \rangle \arrow[u] & &
\end{tikzcd}
$$

The following definition for determining whether a literal is warranted, given a dialectical tree, has been developed from that by Pollock, which we reviewed in section 8.1.1. We have taken this definition and used it as the basis for our definition of the judge function (in chapter 5).

Definition 8.2.38 Let T be a dialectical tree for $\langle \Phi, \alpha \rangle$. The **marked dialectical tree** for T is obtained as follows: (1) If N is a leaf node, then it is marked U; (2) if N is a nonleaf node, and at least one of its children is marked U, then N is marked D; and (3) if N is a nonleaf node, and all of its children are marked D, then N is marked U. For an argument $\langle \Phi, \alpha \rangle$ and a marked dialectical tree for this argument, the literal α is **warranted** iff the roof is marked as U.

Example 8.2.24 The following is a marked dialectical tree that we obtained from the dialectical tree in example 8.2.23:

$$
\begin{tikzcd}[column sep=tiny]
 & \langle \{(\alpha -\!\!< \beta), (\beta -\!\!< \gamma)\}, \alpha \rangle^{D} & \\
\langle \{(\sim\!\beta -\!\!< \gamma \wedge \delta)\}, \sim\!\beta \rangle^{U} \arrow[ur] & \langle \{(\sim\!\beta -\!\!< \gamma \wedge \phi), (\phi -\!\!< \gamma)\}, \sim\!\beta \rangle^{D} \arrow[u] & \langle \{(\sim\!\beta -\!\!< \varepsilon)\}, \sim\!\beta \rangle^{U} \arrow[ul] \\
\langle \{(\sim\!\phi -\!\!< \gamma \wedge \mu), (\mu -\!\!< \pi)\}, \sim\!\phi \rangle^{D} \arrow[ur] & & \langle \{(\sim\!\phi -\!\!< \lambda)\}, \sim\!\phi \rangle^{U} \arrow[ul] \\
\langle \{(\sim\!\mu -\!\!< \kappa)\}, \sim\!\mu \rangle^{U} \arrow[u] & &
\end{tikzcd}
$$

The development of DeLP has involved refining a number of key ideas in using defeasible logic as the underlying logic for argumentation. The net result is an interesting theoretical approach that has also been used in application studies. There are implementations, both compilers and interpreters, of DeLP [GS04]). Furthermore, prototype applications have been built, including a recommender system [CM05] and a tool for analyzing natural language usage based on a Web corpus [CM04b].

8.2.5 Discussion of Argumentation Based on Defeasible Logics

Defeasible logics capture defeasible rules in the object language that can be put together using the proof theory to construct arguments for and against particular conclusions. Furthermore, these logics offer a range of strategies for resolving conflicts that arise between the competing arguments. This range of strategies offers routes for users to select strategies according to the characteristics of the problem domain. The proposals do vary in notational complexity. Some assume just implication connectives for defeasible rules, and some assume the classical connectives in addition.

When we compare this range of possibilities with our framework, the first key difference is that our framework is based on classical logic, whereas these formalisms are based on defeasible logic. The advantage of classical logic is that it has a well-understood language with a clear semantics. When confronted with a classical formula or set of classical formulae, the user has a good understanding of what they mean. This may not be so with defeasible logic. In particular, the semantic conceptualization of the logic is less clear when compared with classical logic. Part of this weakness surrounds the meaning of defeasible implication, and part surrounds the meaning of the negation operators. Even for logic programming, a simpler system than the argumentation systems proposed here, there are numerous proposals for accounting for the semantics of negation.

A key advantage of using defeasible logic as the underlying logic for argumentation, when compared with the classical logic approach, is that if a knowledgebase is to be constructed to provide particular arguments in particular situations, then it seems to be easier to program, that is, edit a knowledgebase, to give those arguments. For example, in the case of a decision support application that is to provide arguments for and against particular treatment plans for particular types of patients, it seems to be easier to do this in a defeasible logic than in classical logic. In defeasible logic, the language and types of inference are more restricted than in classical logic, and so it is easier to predict lines of reasoning that will ensue from a given knowledgebase.

Another advantage of defeasible logics is that machine learning and statistical techniques can be used to learn defeasible rules with priorities [CHS93].

8.3 Argumentation Based on Inference Rules

An alternative to using a defeasible implication in the object language is to extend the inference rules of the underlying logic with domain-specific inference rules. As with defeasible rules, domain-specific rules have a consequent that is inferred if the condition can be inferred. For example, with the domain-specific inference rule below, if α, β, and γ can be inferred from the knowledgebase using the inference rules of the logic together with the domain-specific rules, then δ can be inferred:

$$\frac{\alpha, \beta, \gamma}{\delta}$$

In this way, the underlying logic is composed of a set of inference rules for manipulating the logical connectives, which may include conjunction introduction, implication elimination, and so forth, plus some domain-specific rules.

8.3.1 Abstract Argumentation Systems

An abstract argumentation system is a collection of "defeasible proofs," called arguments, together with an ordering relation that represents the relative difference in force of the arguments [Vre97]. The system does not assume a particular underlying language, and so no consideration is given to connectives including negation, though the language does include $\perp$ as a symbol. An advantage of this setup is that different underlying logics can be used with domain-specific inference rules. Hence, the proposal is an abstract approach. A drawback, though, of this is that if negation is not available systematically, then all contradictions must be defined explicitly beforehand, which may be a huge task.

Assuming $\phi_1, \ldots, \phi_{n-1}, \phi_n$ are object-level formulae, then strict rules of inference are of the form $\phi_1, \ldots, \phi_{n-1} \rightarrow \phi_n$ and defeasible rules of inference are of the form $\phi_1, \ldots, \phi_{n-1} \Rightarrow \phi_n$. Thus, if there are arguments for the premises of a rule holding, then there is an argument for the consequent holding. In this way, rules can be composed into trees to give more complex trees.

The framework focuses on incompatibility of sets of arguments, as opposed to inconsistency of pairs of arguments, and defeat is defined in terms of incompatibility and the ordering relation over arguments, so that in case of incompatibility, less preferred arguments are defeated. This is used via a notion of inductive warrant. We start with a fixed base set of arguments that is a finite compatible subset of the language and then generate all arguments from this. This gives the level 1 arguments. At level 2, some arguments are defeated by arguments not defeated by arguments at level 1. At level 3, some arguments of level 2 that were defeaters at level 2 are defeated by arguments at level 2, and so the arguments they defeated are reinstated. This process continues at level 4,5,... until a fixpoint is reached. This approach is therefore adapted from Pollock's notion of warrant presented in section 8.1.1.

In practice, and for comparison with defeasible logics, we can treat the metalevel symbols $\rightarrow$ and $\Rightarrow$ as object-level symbols. The practical advantage of this is that we then have a relatively simple language together with an intuitive strategy for deciding which arguments should ultimately hold. Some users may appreciate a language that does not incorporate the usual connectives and does not force incompatibility to equate with α and $\neg\alpha$ holding for some formula α.

8.3.2 Assumption-Based Argumentation Frameworks

Assumption-based argumentation is another key approach to developing logical argumentation conforming to the notion of admissible arguments [BDKT97, DKT06]. Furthermore, assumption-based argumentation can be based on first-order classical logic.

Definition 8.3.1 A **deductive system** is a pair $(\mathscr{L}, \mathscr{I})$ where $\mathscr{L}$ is a formal language consisting of countably many formulae and $\mathscr{I}$ is a countable set of inferences rules of the following form:

$$\frac{\alpha_1, \ldots, \alpha_n}{\beta}$$

Here, $\beta \in \mathscr{L}$ is called the conclusion of the inference rule, $\alpha_1, \ldots, \alpha_n \in \mathscr{L}$ are called the premises of the inference rule, and $n \geq 0$.

In these frameworks, there is no distinction between domain-independent axioms, which belong to the specification of the logic, and domain-dependent axioms, which represent a background theory.

Definition 8.3.2 A **deduction** of a conclusion α based on a set of premises P is a sequence $\beta_1, \ldots, \beta_m$ of formulae in $\mathscr{L}$, where $m > 0$ and $\alpha = \beta_m$ such that for all $i = 1, \ldots, m$, either of the following options hold:

1. $\beta_i \in P$
2. there exists $\alpha_1, \ldots, \alpha_n / \beta_i \in \mathscr{I}$ such that $\alpha_1, \ldots, \alpha_n \in \{\beta_1, \ldots, \beta_{i-1}\}$

If there is a deduction of a conclusion α based on a set of premises P, then this is denoted by $P \vdash' \alpha$, and the deduction is said to be supported by P.

Deductions are the basis for the construction of arguments, but to obtain an argument from a deduction, the premises of the deduction are restricted to ones that are acceptable as assumptions.

Definition 8.3.3 An **assumption-based framework** is a tuple $\langle \mathscr{L}, \mathscr{I}, \mathscr{C}, f \rangle$ where

1. $(\mathscr{L}, \mathscr{I})$ is a deductive system.
2. $\mathscr{C}$ is the set of candidate assumptions where $\emptyset \neq \mathscr{C} \subseteq \mathscr{L}$
3. If $\alpha \in \mathscr{C}$, then there is no inference rule of the form $\alpha_1, \ldots, \alpha_n / \alpha \in \mathscr{I}$
4. f is a total mapping from $\mathscr{C}$ into $\mathscr{L}$

In the above definition, f is defined so that $f(\alpha)$ is the contrary of α.

A deduction may fail to be an argument because some of its premises may be conclusions of inference rules or because some of its premises may not be acceptable as assumptions (they do not belong to $\mathscr{C}$).

Definition 8.3.4 An **argument** is a deduction whose premises are all assumptions.

In this approach, the only way to attack an argument is to attack one of its assumptions.

Definition 8.3.5 An argument $\beta_1, \ldots, \beta_m$ attacks an argument $\alpha_1, \ldots, \alpha_n$ iff $\beta_1, \ldots, \beta_m$ attacks an assumption α_i in $\alpha_1, \ldots, \alpha_n$. An argument $\beta_1, \ldots, \beta_m$ attacks an assumption α_i iff $\beta_m = f(\alpha_i)$. A set of assumptions A attacks a set of assumptions B iff there exists an argument $\beta_1, \ldots, \beta_m$ such that $\{\beta_1, \ldots, \beta_m\} \subseteq A$ and $\beta_1, \ldots, \beta_m$ attacks an assumption in B.

Definition 8.3.6 A set of assumptions A is **admissible** iff the following two conditions hold:

1. A attacks every set of assumptions that attacks A
2. A does not attack itself

A belief α_n is admissible iff there exists an argument $\alpha_1, \ldots, \alpha_n$ and an admissible set A such that every assumption in $\alpha_1, \ldots, \alpha_n$ is in A.

The general definition of assumption-based frameworks offers a number of interesting possibilities. Here we consider a class of assumption-based frameworks delineated as follows:

- All formulae in $\mathscr{L}$ are literals (i.e., atoms $p, q, \ldots$ or negations of atoms $\neg p, \neg q, \ldots$).
- The set of assumptions $\mathscr{C}$ is a subset of the set of all literals that do not occur as the conclusions of any inference rule in $\mathscr{I}$.
- The contrary of any assumption p is $\neg p$, and the contrary of any assumption $\neg p$ is p.

For notational convenience, inference rules will be in the format $\alpha \leftarrow \alpha_1 \wedge \cdots \wedge \alpha_m$ where α is the conclusion of the inference rule and $\alpha_1, \ldots, \alpha_m$ are the premises.

Example 8.3.1 Let $\langle \mathscr{L}, \mathscr{I}, \mathscr{C}, f \rangle$ be the following assumption-based framework:

$\mathcal{L} = \{\alpha, \beta, \gamma, \delta, \varepsilon, \neg\alpha, \neg\beta, \neg\gamma\neg\delta, \neg\gamma, \varepsilon\}$
$\mathcal{I} = \{\alpha \leftarrow \beta \wedge \gamma, \gamma \leftarrow \delta, \neg\beta \leftarrow \varepsilon\}$
$\mathcal{C} = \{\beta, \delta, \varepsilon\}$

From this, two arguments for α are given below:

$\delta, \gamma, \beta, \alpha$
$\varepsilon, \neg\beta, \delta, \gamma, \beta, \alpha$

A **backward argument** is an argument such that the argument can be constructed by backward chaining reasoning. This means that the argument is constructed starting with the goal, and then subgoals required to prove the goal, and then by recursion sub-subgoals required to prove each subgoal, and so on by recursion. In the above example, only the first of the two arguments is a backward argument.

An admissible set of assumptions can be viewed using the following notion of an abstract argument tree.

Definition 8.3.7 An **abstract dispute tree** for an initial argument X is a (possibly infinite) tree T such that

1. Every node of T is labeled by an argument and is assigned the status of **proponent** or **opponent** but not both.

2. The root is a proponent node labeled by X.

3. For every proponent node N, labeled by an argument Y, and for every argument Z that attacks Y, there exists a child of N, which is an opponent node labeled by Z.

4. For every opponent node N, labeled by an argument Y, there exists exactly one child of N, which is a proponent node labeled by an argument that attacks some assumption α in the set supporting Y. α is said to be the **culprit** in Y.

5. There are no other nodes in T except those given by 1 to 4 above.

The set of all assumptions belonging to the proponent nodes in T is called the **defense set** of T.

Example 8.3.2 Consider the assumption-based framework $\langle \mathcal{L}, \mathcal{I}, \mathcal{C}, f \rangle$ with $\mathcal{I}$ consisting of the following rules

$\neg\delta \leftarrow \beta, \neg\beta \leftarrow \gamma \wedge \delta, \neg\beta \leftarrow \phi \wedge \psi, \neg\gamma, \neg\psi$

and $\mathcal{C} = \{\beta, \gamma, \delta, \phi\}$, $\mathcal{L} = \mathcal{C} \cup \{\neg\delta, \neg\beta, \neg\gamma, \neg\phi, \psi, \neg\psi\}$, and $f(\lambda) = \neg\lambda$ for all $\lambda \in \mathcal{C}$.

From this, an abstract dispute tree for $\{\beta\} \vdash \neg\delta$ is as follows:

$$\begin{array}{l} \textit{proponent} : \{\beta\} \vdash' \neg\delta \\ \qquad\qquad \uparrow \\ \textit{opponent} : \{\gamma, \delta\} \vdash' \neg\beta \\ \qquad\qquad \uparrow \\ \textit{proponent} : \{\ \} \vdash' \neg\gamma \end{array}$$

Definition 8.3.8 An abstract dispute tree T is admissible iff no culprit in the argument of an opponent node belongs to the defense set of T.

Admissible abstract dispute trees can be infinite in breadth and infinite in depth, though a dispute tree that has no infinitely long branches is an admissible dispute tree. As a development of dispute trees, there is an algorithmic proof procedure for assumption-based argumentation frameworks that finds the admissible sets of assumptions.

In chapter 3, we discussed how our framework compared with Dung's approach. From that discussion, it was clear that to avoid the problems of each logical inconsistency between supports for arguments (according to classical logic) resulting in an attack relationship holding, the attack relationship needed to be restricted somehow. In PAFs (reviewed in section 8.1.4), priorities are used to avoid the symmetry arising in the attack relationship, and in argument-based extended logic programming (reviewed in section 8.2.3), both priorities and an underlying defeasible logic, which is weaker than classical logic (in the sense that it has fewer proof rules), are used. Assumption-based argumentation frameworks can, in a sense, be viewed as a generalization of the argument-based extended logic programming approach, by allowing the user to define which proof theory is used, and normally this would be weaker than classical. However, it is possible to use first-order logic, including classical first-order logic.

8.3.3 Discussion of Argumentation Based on Inference Rules

An alternative to using a defeasible implication in the object language is to extend the inference rules of the underlying logic with domain-specific inference rules. In many respects, the net effect is the same when using it for the underlying logic for an argumentation system. Indeed, it could be argued that the ontological status of defeasible rules and domain-specific rules coincide. Perhaps, more importantly, it seems whatever can be done with one can be done with the others. For example, it seems that an argumentation system based on inference rules could be presented equivalently as an argumentation system based on defeasible rules. These two systems would be isomorphic in the sense that there would be bijection between the formulae (of the object and metalanguage) of the two presentations (i.e., the one based on defeasible rules and the one based on inference rules). Thus, in order to compare our framework with those based on inference rules, the same comments presented in section 8.2.5 are applicable.

8.4 Argumentation Based on Default Logic

Default logic was originally proposed by Reiter [Rei80] as a basis for representing and reasoning with information that is normally correct, but for which exceptions exist. It is one of the best known and most widely studied formalizations for default knowledge [Bes89, BDK97, Ant97, Sch98]. Furthermore, it offers a very expressive and lucid language. In default logic, information is represented as a *default theory*, which consists of a set of first-order formulae and a set of *default rules* for representing default information. Default rules are of the following form, where α, β, and γ are first-order (classical) formulae:

$$\frac{\alpha : \beta}{\gamma}$$

The inference rules are those of classical logic plus a special mechanism to deal with default rules: Basically, if α is inferred, and $\neg\beta$ cannot be inferred, then infer γ. For this, α is called the precondition, β is called the justification, and γ is called the consequent.

The set of formulae that are derivable from a default theory is called an extension. Each extension is a set of classical formulae. There may be more than one extension per default theory. Default logic extends classical logic. Hence, all classical inferences from the classical information in a default theory are derivable (if there is an extension). The default theory then augments these classical inferences by default inferences derivable using the default rules.

The generating defaults (the defaults used) for each extension are the basis of the arguments for and against any possible conclusion. In this way, default logic offers an interesting foundation for argumentation. There are also a number of developments that are potentially useful for argumentation, including prioritized default logic, and a form of defeasible reasoning on normal default logic.

In prioritized default logic [Bre91], a partial order on the defaults is used. To use this, the notion of a default extension is redefined as follows:

If $D_1, \ldots, D_n$ is a collection of sets of normal defaults, where there is a total ordering over these sets, such that D_1 is the most preferred, and W is a set of nondefeasible data, then E is a prioritized default extension of $(D_1, \ldots, D_n, W)$ iff there exist sets of formulae $E_1, \ldots, E_n$ such that

E_1 is an extension of (D_1, W)

E_2 is an extension of (D_2, E_1)

$\vdots$

$E = E_n$ is an extension of (D_n, E_{n-1})

In this way the defaults higher in the ordering have preference over the defaults lower in the ordering when generating extensions. This has been further developed to reason about priorities in the object language [Bre94]. As a result, it is possible to change the priorities according to the context. This is a very useful feature for certain domains, including for legal reasoning.

There have been a number of other proposals to develop default logic by extending the syntax of default rules. For argumentation, an important example, by Prakken and Sartor [PS97], takes the argument-based extended logic programming approach, which we reviewed in section 8.2.3, and bases defeasible reasoning on default rules with an associated label attached. This label is the name for the rule, and it is also a term in the classical language. This means that rules can be referred to in the formulae and hence in the default rules. Arguments are then constructed by composing sequences of default rules and first-order formulae using first-order reasoning. This is building on the notion of default rules as a form of natural deduction rule. There is a notion of defeat that takes into account the relative priorities of the defaults. The priorities are represented by an ordering over the labels for the defaults. Furthermore, since the labels are terms in the language, the priorities can be reasoned about in the language.

There are a number of other approaches based on default logic that could be described as offering argumentative reasoning (i.e., reasoning for constructing individual arguments). The family of default logics is one of the most well-explored formalisms for default knowledge. There are a range of useful variants, and inferencing technology is being developed (for a review, see [Bes89, Bre91, Ant97, Sch98, BDK97]). It is also possible to learn default rules from data [DN99].

8.5 Argumentative Attitudes

Much of the research in formalizing argumentation has, in the monological case, focused on agents who are assumed to be honest and transparent, and in the dialectical case, it has focused on agents who are assumed to be honest and cooperative or who are assumed to be honest and competitive. However, there is need to develop proposals for formalizing other kinds of attitude in argumentation, such as agents who are manipulative or agents who have an agenda.

We start by consider a proposal for formal handling of threats and rewards in argumentation [AP05a, AP05b]. For threats and rewards, we assume there is an agent who is dealing with another agent and that the agent has three sets of propositional formulae for constructing arguments: A set of its own goals, G; A knowledgebase, K; and A set of goals that it thinks are those of the other agent, H.

A **threat argument** is a triple $\langle \Gamma, \alpha, \beta \rangle$ where $\Gamma \subset K$ is the support of the threat argument, $\alpha \in G$ is the aim of the threat, $\beta \in H$ is the penalty of the threat, and the following three conditions hold:

1. $\Gamma \cup \{\neg\alpha\} \vdash \neg\beta$
2. $\Gamma \cup \{\neg\alpha\} \nvdash \perp$
3. Γ is minimal for set inclusion for these conditions

Consider the following illustration taken from [AP05a]. For an agent who is a mother of a schoolboy, the agent may have the following sets of formulae:

$K = \{\neg\texttt{SchoolWorkComplete} \rightarrow \neg\texttt{GoToParty}\}$

$G = \{\texttt{SchoolWorkComplete}\}$

$H = \{\texttt{GoToParty}\}$

Thus, to try to make the schoolboy do his schoolwork, the mother may try the following threat that says the schoolboy will not go to the party if the schoolwork is not completed:

$$\langle \{\neg\texttt{SchoolWorkComplete} \rightarrow \neg\texttt{GoToParty}\}, \texttt{SchoolWorkComplete}, \texttt{GoToParty} \rangle$$

A **reward argument** is a triple $\langle \Gamma, \alpha, \beta \rangle$ where $\Gamma \subset K$ is the support of the reward argument, $\alpha \in G$ is the aim of the reward, $\beta \in H$ is the benefit of the reward, and the following three conditions hold:

1. $\Gamma \cup \{\alpha\} \vdash \beta$
2. $\Gamma \cup \{\alpha\} \nvdash \perp$
3. Γ is minimal for set inclusion for these conditions

Consider the following illustration, taken from [AP05a]. For an agent that is an employer, the agent may have the following sets of formulae:

$K = \{\texttt{CompleteExtraTask} \to \texttt{GetExtraPay}\}$

$G = \{\texttt{CompleteExtraTask}\}$

$H = \{\texttt{GetExtraPay}\}$

Thus, to try to make an employee do an extra task, the employer may try the following reward argument that says the employee will get extra pay if the extra task is completed:

$$\langle\{\texttt{CompleteExtraTask} \to \texttt{GetExtraPay}\}, \texttt{CompleteExtraTask}, \texttt{GetExtraPay}\rangle$$

This framework for reward and threat arguments has been incorporated into a dialogue system supporting negotiation [AP05a, AP05b].

The second proposal we consider here drops the very common assumption that agents are honest [AR06]. In this proposal, agents can lie. This is based on a supraclassical system called X-logic, which allows for the construction of arguments as defined in chapter 3 but allows for an agent to partition his or her knowledge into truthful knowledge (i.e., knowledge that the agent believes to be true) and fallacious knowledge (i.e., knowledge that the agent believes to be false). Arguments that use some fallacious knowledge can then be described as incorporating a lie. This then offers way of providing a richer model of the proponent and offers the possibility for formalizing persuasional and provocational argumentation as delineated in chapter 1.

8.6 Argumentation Technology

Tools for supporting monological argumentation include the following classes: **editing tools** (tools for a user to construct and edit constellations of arguments and counterarguments), **checking tools** (automated reasoning tools for checking that a user's constellation of arguments and counterarguments is correct), and **generating tools** (automated reasoning tools for automatically constructing constellations of argumentation and counterarguments from a knowledgebase).

In this first class, there are numerous implemented software systems that offer the ability for a user to construct arguments and counterarguments using a graphical interface (see, e.g., [RR04, KBC03]). Many of these are based on informal notations for representing arguments that incorporate the facility to represent and edit pros and cons for arguments and various annotations such as representing assumptions, causal explanations, predictions, and questions.

As an interesting application area of the informal argumentation approach, editing tools have also been developed to support scientists in analyzing free text arguments obtained from a collection of papers, allowing the scientist to flag relationships between evidence from different papers such as "supports," "contradicts," and so forth using a graphical notation (see, e.g., ClaimMaker [Buc07]).

In the second class, there are relatively few implemented systems. Perhaps the best known is the ArguMed System, developed as an automated argumentation assistance system for lawyers [Ver99]. The system assists users in making statements, abducing reasons, inferring conclusions, and providing exceptions. There is a graphical representation of attacks of arguments for a propositional logic with syntactic sugar (e.g., see figure 8.2). The underlying formalism is a form of defeasible argumentation that allows for arguments and undercutters to arguments to be represented in

```
Peter has committed a tort.
  ↑
  |  THAT There is a ground of justification for Peter's act,
  |  IS AN EXCEPTION TO THE RULE THAT,
  |  IF Peter has violated a property right,
  |  THEN Peter has committed a tort.
  |    ↑
  |    |  According to art.6:162.2 of the Dutch civil code,
  |    |  violations of property rights are torts,
  |    └─unless there is a ground justification.
  |
  ×─There is a ground of justification for Peter's act.
  |
  └─Peter has violated a property right.
```

Figure 8.2
An argument structure represented in the ArguMed System [Ver99]. The conclusion `Peter has committed a tort` that is derived from the fact `Peter has violated a property right` has the warrant `IF Peter has violated a property right, THEN Peter has committed a tort` with backing `According to art.6:162.2 of the Dutch civil code, violations of property rights are torts, unless there is a ground justification`. In addition, there is the undercutter `There is a good of justification for Peter's act` obtained by the exception to the warrant.

a form analogous to that originally proposed by Toulmin [Tou58]. The system supports the process of argument construction by allowing for annotation of various kinds of incompleteness including an exception that is not justified and an undercutter that is not justified. The ArguMed software incorporates algorithms for checking that argument structures are complete. It also incorporates various editing functions.

Another example of a system that offers both editing and checking functionality is the Zeno Argumentation Framework [GK97]. This system is designed to help users analyze questions, so called "issues," by identifying choices and constraints for solving each issue and by identifying pros and cons for justifying the choices and constraints considered. The result of using the system is that the user has constructed a tree with the main issue at the root, and then each subsequent node is a choice, a constraint, a subsidiary issue, a pro, or a con. The leaves are assumed to be undisputed facts. Each such fact is a pro, a con, or a constraint. An example of a graph produced in this way is given in figure 8.3. Different formalized criteria can be used for drawing a conclusion from a Zeno argumentation graph including beyond reasonable doubt, best choice, no better alternative, preponderance of evidence, and scintilla of evidence.

A further example of a system with editing and checking functionality is the Room 5 system [LNA+97]. This was implemented as a system for argumentation with U.S. Supreme Court precedents. It is partly based on formal argument games and defeasible logics.

In the third class, there are a few examples of implementations of automated reasoning systems for argumentation. These include implementations of interpreters for DeLP [GS04] (discussed in section 8.2.4), for argumentation as a form of defeasible reasoning (discussed in section 8.1.1) called OSCAR [Pol95], for probabilistic argumentation (related to existential arguments discussed in section 8.1.2 with associated probabilities) called ABEL [Hae98, HKL00], for assumption-based argumentation (related to the proposal discussed in section 8.3.2) developed in logic programming [KT99], and a query answering system for computing minimal lines of defense for an argument to be in a stable or admissible extension according to the definitions given in chapter 2 [Vre06].

Given the progress in theories for argumentation, and in the three classes of tools discussed above, there is increasing interest in refining formal theories of argumentation and developing supporting tools in order to deploy the conjectured benefits of argumentation systems in applications. Some key application areas for which argumentation systems are being developed include legal reasoning [Pra97, Ver99, Ver05], medical

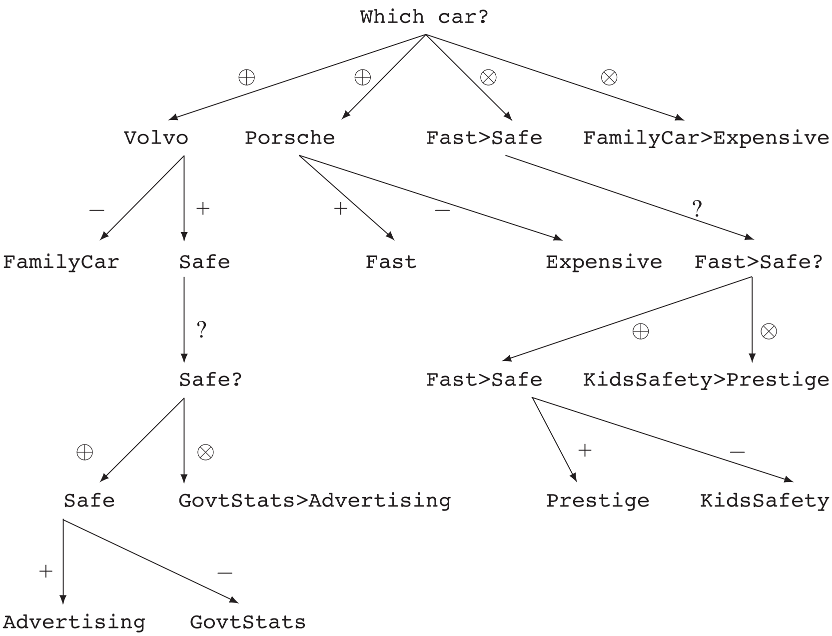


Figure 8.3
An example of a Zeno argumentation graph where ? denotes an issue, $+$ denotes a pro, $-$ denotes a con, $\oplus$ denotes a choice, and $\otimes$ denotes a constraint. In this example, the main issue is `which car?` There are two choices, `Volvo` and `Porsche`, and two constraints: `Fast` is preferred to `Safe`, and `FamilyCar` is preferred to `Expensive`. There are pros and cons for the choices. Of these pros and cons, the only issue is `Safe?` The others are accepted as undisputed facts. The only choice for the issue `Safe?` is `Safe`, and there is a pro and a con, both of which are accepted as undisputed facts. However, there is also a contraint that `GovtStats` (Government Statistics) are preferred to `Advertising`. In an analogous way, the constraint `Fast` is preferred to `Safe` is also an issue. This constraint can be rejected because the con `KidsSafety` is preferred to the pro `Prestige`. This example is adapted from [GK97].

reasoning [FJR98, FD00], scientific reasoning [Hun05], risk assessment [KAEF95, KFJP98], decision making [Amg05, ABP05], access control management [BE04], software engineering [HN98, HMLN05], agent systems [PSJ98, RRJ^{+}03, PWA03, AP05a, AP05b], Web search technology [CM05], and recommender systems [CM04a, CMS06]. For a review of argumentation technologies and applications, see [CRL00, ASP04].

8.7 Discussion

We started this chapter by saying that we would consider a variety of proposals for logic-based argumentation in terms of the underlying logic and the notion of warrant.

By considering the underlying logic, we see that there are a number of proposals like ours that use classical logic as the underlying logic. However, there are also a number of approaches that consider either a defeasible implication or inference rules for capturing defeasible/default rules. These defeasible/default rules have been extensively studied in the literature on nonmonontonic reasoning as well as in argumentation. Primary drivers for using such rules instead of using classical implication are (1) the requirement for obviating contrapositive reasoning with the defeasible/default rule and (2) the emphasis on implementation with a formalism that can be developed as an adaptation of the logic programming implication. In some proposals, the defeasible/default rule is part of a language that extends classical logic, whereas in others, the language consists just of rules and literals (atoms and negated atoms) and so can be viewed as logic substantially weaker than classical logic.

As we have discussed in chapter 3, our framework goes beyond existing proposals for argumentation based on classical logic by focusing on canonical undercuts. With regard to introducing connectives for defeasible rules or introducing domain-specific inference rules, these can provide alternative logical characteristics to our framework. If we view these different proposals for logical argumentation as conceptual tools for modeling aspects of cognition, or for engineering intelligent systems, then it seems that the choice of whether to use classical logic as the underlying logic or to use a defeasible/default formalism as the underlying logic depends on the requirements of the task to which the conceptual tool is being applied. There does not appear to be any indication yet of a convergence on a single underlying logic for logical argumentation. Rather there are pros and cons with each of the candidates considered in the literature.

Whatever underlying logic is used, all the logic-based argumentation systems use a similar notion of argument. Numerous approaches use a definition that is effectively the same as ours. In other words, numerous approaches assume an argument is a pair where the first item is a minimal consistent subset of the knowledgebase that entails the second item, which is a formula. With this conceptualization, the underlying logic provides the required definitions for consistency and entailment. Some approaches put more information into the representation of an argument. In particular, some require both the assumptions and some representation of the proof.

By considering the notion of warrant, we see that Dung's notion of acceptable argument has been particularly influential in providing a semantics for finding arguments that are warranted. Various proof theoretic counterparts to this semantics have been proposed by Prakken and Sartor, as discussed in section 8.2.3; Dung et al., as discussed in section

8.3.2; and Amgoud and Cayrol, as discussed in section 8.1.4. We compared our approach to argumentation with that of Dung in section 3.9.

A number of approaches incorporate metalevel information in the knowledgebase that provides priority information over the formulae in the knowledgebase. This is a useful feature for capturing numerous intuitive examples. It is also something that, as we discussed in section 3.9, is important for getting around the problem of interpreting logical inconsistency as an attack relation (i.e., if we say that argument A attacks argument B when the support of A negates the support of B, then we have symmetry because A attacks B implies B attacks A). In the presentation of our framework, we have not introduced priority information over formulae. However, it can be seen from this chapter that it should be straightforward to extend our proposal with priority information. A good starting point for doing this would be to adapt some of the proposals by Cayrol et al. that we discussed in sections 8.1.3 and 8.1.4. Another good starting point would be to harness reasoning with defeasible priorities as considered by Prakken and Sartor [PS97] and reasoning with implicit priorities based on specificity of information as developed for argumentation by Simari and Loui [SL92].

There are some substantial features of our proposal for logical argumentation that are unique to our proposal. In this book, we have been advocating the development of logic-based techniques for practical argumentation. This is based around the idea that argumentation has to be selective and reformative and that it should be based on analysis of both intrinsic and extrinsic aspects of argumentation. Not least, this should include taking the audience into account. Prior to our proposal, there has been relatively little proposed for analyzing arguments by taking the intended audience into account. An exception that we discussed in chapter 2 is by Bench-Capon [Ben03], and there are some developments in conceptualizing argumentative attitudes, discussed in section 8.5, by Amgoud and Prade [AP05a, AP05b] and by Aubry and Risch [AR06] that, in a sense, take aspects of the audience into account. These developments in formalizing attitudes offer further ways that we could extend our framework in order to have a richer model of the audience and to better tailor arguments according to need.

Finally, there are also techniques for taking a set of argumentation frameworks and merging them to produce a single argumentation framework [CDK+05]. These techniques treat each argumentation framework as an "agent" and then use voting theory to select the argumentation framework that is output. These techniques raise some interesting possibilities for logic-based argumentation.

9 Future Directions

This book has set out to show that key elements of deductive argumentation can be formalized in logic. In this chapter, we assess how well this goal has been achieved and consider some of what is still missing in the endeavor to understand the principles underlying argumentation as a cognitive process and in order to make argumentation a useful part of artificial intelligence technology.

9.1 Assessment of Our Approach

We listed some of the key elements of argumentation that we wanted to formalize in our logic-based framework in chapter 1, and we developed these elements further in chapter 4 when we considered some aspects of monological argumentation arising in practical argumentation situations. To address these requirements, we have presented a framework for propositional and first-order argumentation based on classical logic that has the following features: (1) definitions for argument trees composed from maximally conservative arguments; (2) analytic techniques for intrinsic factors including degree of undercut; (3) analytic techniques for extrinsic factors including degree of resonance, degree of empathy, and degree of antipathy; (4) rationalization techniques for argument trees including pruning, contracting, compressing, and shifting; (5) judgment criteria for constellations of arguments for determining the warranted claims, using structural features of the constellation, and other analyses of intrinsic and extrinsic factors; and (6) algorithms for constructing arguments and argument trees including compilation, contouring, and approximation techniques.

Overall, we have drawn out a number of key elements of argumentation using this logic-based framework. The basic elements of argumentation that have been considered in the book include coverage of the concepts of argument, support, attack, constellations of arguments and counterarguments, and judgment of arguments. Furthermore, we have argued that the basic elements arise in many of the other logic-based proposals for argumentation and that the more advanced elements (such as analysis of intrinsic and extrinsic factors, and reformation of constellations of arguments) can be adapted and incorporated into the other logic-based proposals for argumentation.

9.2 Key Issues for Future Development

While we believe that our logic-based framework goes some way towards addressing the needs for intelligent practical argumentation, as sought in chapter 4, there remain numerous issues that need to be addressed in future research for logic-based argumentation to better capture practical argumentation and to undertake monological argumentation automatically. We discuss some of these issues here, some of them being short-term research issues and others being open-ended research issues.

9.2.1 Harmony in Argumentation

Clearly, inconsistency is a core issue in argumentation. If there are no inconsistencies in a knowledgebase, then no counterargument can be found with that knowledgebase. In chapter 5, we saw how the degree of undercut can be viewed as a measure of the degree to which an undercut negates, or conflicts with, its parent. Degree of undercut is based on a pairwise measure of inconsistency between an argument and one of its undercuts. This means that degree of undercut does not consider sets of arguments (e.g., a set of siblings, a set of defending arguments, or a set of attacking arguments). In other words, it is not possible to use this measure to determine, say, the "degree of consistency" of the union of the supports for a set of defending arguments.

Yet characterizing the conflicts arising in a set of arguments may be useful in making an overall assessment of the quality of a constellation of arguments and counterarguments. To illustrate this, consider the following three informal arguments:

(a_1) The foreign takeover of a major oil company is ok because a takeover is a form of inward investment in a national economy and oil companies are not a special case in a national economy.

(a_2) Oil companies are a special case in a national economy because the government needs to ensure the rate of production of oil is increased, and this is because the national rate of consumption of oil should increase, which in turn is because a national economy needs to become more technologically advanced to address the problems arising from global warming.

(a_3) Oil companies are a special case in a national economy because the government needs to closely manage the oil consumption in the country, and this is because the national rate of consumption of oil should not increase, which in turn is because each nation needs to address the problems arising from global warming.

We can see informally that a_1 is an argument with the claim that *the foreign takeover of a major oil company is ok*, and a_2 and a_3 are undercuts to a_1. Furthermore, we can see that a_2 and a_3 are mutually contradictory, since a_2 includes the premise that *the national rate of consumption of oil should increase* and a_3 includes the premise that *the national rate of consumption of oil should not increase*. Thus, while each of a_2 and a_3 might be regarded as good arguments, they are conflicting. It is straightforward to see this contradiction in a logical formalization.

Example 9.2.1 Consider the following propositions:

t Foreign takeover of a major oil company is ok.

i Each takeover is a form of investment in a national economy.

s Oil companies are a special case in a national economy.

p Government needs to ensure that the rate of production of oil is increased.

c National rate of consumption of oil should increase.

a National economy needs to become more technologically advanced.

g There are problems arising from global warming.

m Government needs to closely manage the oil consumption in the country.

Using these propositions, we can form the following knowledgebase Δ:

$$\{\neg s, \neg s \wedge i \rightarrow t, g, g \rightarrow a, a \rightarrow c, c \rightarrow p, p \rightarrow s, g \rightarrow \neg c, \neg c \rightarrow m, m \rightarrow s\}$$

Hence, we can form the following argument tree that reflects the information in the informal arguments A_1, A_2, and A_3 as follows:

$$\begin{array}{cc}
\multicolumn{2}{c}{\langle\{\neg s, \neg s \wedge i \rightarrow t\}, t\rangle} \\
\nearrow & \nwarrow \\
\langle\{g, g \rightarrow a, a \rightarrow c, c \rightarrow p, p \rightarrow s\}, \diamond\rangle & \langle\{g, g \rightarrow \neg c, \neg c \rightarrow m, m \rightarrow s\}, \diamond\rangle \\
\uparrow & \uparrow \\
\langle\{g, g \rightarrow \neg c\}, \diamond\rangle & \langle\{g, g \rightarrow a, a \rightarrow c\}, \diamond\rangle
\end{array}$$

As another example, consider a court case where a witness is testifying about an alibi for the defendant. Suppose we represent the prosecution main case by the argument $\langle\Phi, \alpha\rangle$ at the root of the argument tree. Thus, in effect, Φ is the "proof" that the defendant is guilty of the crime, and α is a statement saying the defendant is guilty. Now the witness provides information for an argument $\langle\Psi, \diamond\rangle$ that undercuts $\langle\Phi, \alpha\rangle$. Perhaps the main case says that the accused was in the office when the victim was murdered, and the undercuts say that the accused was not in the office because he was away from the office that week taking "sick leave" while skiving off on holiday abroad. Later, under cross-examination, the same witness may provide information for another undercut to the root. Perhaps the main case says that the victim was the boss of the accused and the accused was always in trouble with the boss about taking sick leave under false pretences, and that at the time of the murder, the boss was sacking the accused. The undercut may contain information to the effect that the defendant was not being fired by the boss because the defendant never took sick leave. Thus, although the witness has now provided information for two undercuts to the main case, these undercuts have supports that are inconsistent. This inconsistency may be enough to render these undercuts useless in the overall assessment of the validity of the main case. Moreover, the jury may regard the witness as unreliable.

We saw further examples of this kind in section 4.5 where we considered harmony in coalitions of arguments. The common feature that we see in all these examples is that not all conflicts between an argument and a counterargument are equal: There are differences in character, there are differences in significance, and there seem to be conditional dependencies that would allow one to ignore some conflicts in the presence of other conflicts.

To address these kinds of situation, we need to evaluate the nature of conflicts arising between the supports of siblings. We also need to evaluate the nature of inconsistency between the supports of defending arguments and between the supports of attacking arguments. For this, there are a range of options that can be developed by drawing on developments in logic-based techniques for analyzing inconsistent information. A number of proposals have been made for measuring the degree of information in the presence of inconsistency [Loz94, WB01], for measuring the degree of inconsistency in information [Gra78, Kni01, Hun02, Kni03, KLM03, GH06, Hun06c, HK06], and for measuring the significance of inconsistency [Hun03]. For a review see [HK05].

We suggest that we can characterize harmony in a set of arguments in terms of the measures of the degree and the significance of the inconsistency of the union of the supports of those arguments. Measuring the degree and the significance of conflicts arising in argumentation is potentially valuable in resolving the examples we discussed above and in section 4.5. This may allow us to consider the degree to which sets of arguments in an argument tree work together or work against each other. If we also take the significance of conflicts into account, we may capture when it is appropriate to ignore some counterattacks to an argument because the conflicts are not sufficiently significant.

9.2.2 Argumentation with Temporal Knowledge

The ability to reason about time is an important aspect of human intelligence. Much important data and knowledge is temporal in nature, such as *always submit your tax form before the end of the next tax year or you will be fined*, or *Bill Clinton was U.S. President before George W. Bush.* Temporal logics have been developed for diverse applications in natural language understanding, planning and scheduling systems, diagnostic systems and real-time, and distributed systems (for a review of the fundamentals of temporal logics, see [GHR94]).

From the successful applications of temporal logics, it seems that there is much potential in using a temporal language as the object language for argumentation. Furthermore, there appear to be some interesting challenges for developing logic-based argumentation that can handle temporal logic. Therefore, if we are considering the pros and cons for some formula α, then this formula may be a temporal logic formula, and the assumptions used may also be temporal formulae. Hence, consideration needs to be given to arguments over time, since, for example, an undercut to an argument may only be an undercut some of the time: If, in tense logic, a formula α is true at all time points in the future, and a formula $\neg\alpha$ is true at some time point in the future, can the first be undercut by the second? Thus, bringing richer languages, such as modal languages, together with richer underlying semantics, raises questions about what the basic definitions for argumentation should be.

There are a few proposals for argumentation with temporal knowledge, including: a proposal to replace classical logic in coherence-based reasoning (as presented in section 8.1.2) with a temporal logic with modal operators for "yesterday," "tomorrow," "last week," "next week," and so forth [Hun01]; a proposal for extending a form of argumentation based on defeasible logic with temporal knowledge about intervals [AS01]; and a proposal for extending the logic-based argumentation presented in chapter 3 with temporal knowledge about intervals [Man07]. Notwithstanding these proposals, there are many further questions and issues that need to be considered. We discuss a few of these in the remainder of this section.

Some arguments are only "good" arguments periodically. In other words, the impact and/or believability of an argument can depend on the time of day or the time of the year. As an illustration, *I haven't done my Christmas shopping yet, and so I must do it next week* is an argument that could be considered at any time of the year, but obviously it has higher impact in mid-December than in mid-July.

Similarly, the impact and/or believability in the support of some arguments increases or decreases over time. For example, suppose it is a summer morning, in London (where the weather is quite changeable), and it is sunny and warm outside. Thus, we could construct an argument that says *because it is so nice now, let's organize a picnic in the park*. This would be a highly convincing argument for the next few hours, and perhaps even for the rest of the day. However, as we look into the future, the argument becomes less convincing. Given that the location is London, having a sunny day does not provide a good indication of what the weather will be like a week or two into the future.

Temporal reasoning is also closely related with some kinds of commonsense reasoning. For example, some arguments that are "good" for some intervals are good for subintervals. As an illustration, we may have an argument that *the robot traveled less than 2 km in interval i*, and by commonsense reasoning, we know that for every subinterval j of i, we have that *the robot traveled less than 2 km in interval j*. And as an illustration of the contrary, we may have an argument that *the robot traveled 2 km in interval i*, and by commonsense reasoning, we cannot get for a proper subinterval j of i that *the robot traveled 2 km in interval j*. To address these issues, there are a number of proposals for characterizing temporal aspects of commonsense reasoning that could be harnessed with an argumentation formalization (see, e.g., [Sho88]).

9.2.3 Argumentation with Uncertain Knowledge

Uncertainty is a complex and multidimensional aspect of information. Most information about the real world has some uncertainty associated with it. Numerous theoretical frameworks have been developed to formalize aspects of it, including probability theory, possibility theory, and Dempster–Shafer theory.

Clearly, much information used for argumentation has some uncertainty associated with it. A number of proposals for argumentation use possibility theory for capturing preferences over formulae (e.g.); in the ABEL framework reasoning with arguments is augmented with probabilistic information [Hae98, HKL00], and some form of argumentation with qualitative probabilities has been proposed in the qualitative probabilistic reasoner (QPR) framework [Par96, Par98, Par01]. Yet, the incorporation of uncertainty formalisms into argumentation systems is still relatively underdeveloped. To illustrate, we give a couple of examples of situations that argumentation systems are currently unable to handle adequately.

Our first example is of integrating reasoning about probabilities with reasoning about risk. Consider an airliner. It probably will not crash, and it is even less likely to crash over the sea. However, there is a good argument for equipping the aircraft with a life jacket for every passenger. So we could have the following abstract arguments and construct an abstract argumentation framework with a_1 being attacked by a_2 and a_2 being attacked by a_3:

(a_1) An Airbus A340 is safe to fly long haul routes.

(a_2) There is a risk that an airliner can crash in the sea, and passengers may drown as a result.

(a_3) The risk of drowning can be ameliorated by fitting the airliner with life jackets.

While we could write some simple propositional formulae to capture this information directly and represent the arguments in the form of an argument tree, it would be desirable to investigate the relationship between the low probability of crashing and the decrease in the risk of mortality gained by carrying the life jackets. Moreover, this relationship should be established in a structured way, so that, for example, an argument based on probability should be identifiable as such in a constellation of arguments and the (qualitative) probability associated with it should be represented separately from the proposition, and similarly, an argument based on decrease in risk should be identifiable as such in a constellation of arguments and the (qualitative) risk associated with it should be represented separately from the proposition. This calls for drawing on existing formalisms, perhaps including utility theory, and bringing them into an argumentation formalism in a way that maintains their heterogeneity.

For our second example, consider a debate concerning the pros and cons of a new cancer drug that is used in patients who have been treated for cancer to prevent recurrence of cancer. Doctors normally use absolute risk to evaluate such a treatment. Thus, for example, for a particular class of patients, the chance of recurrence may be 2% if given no preventative treatment, whereas the chance of recurrence may be 1% if treated with the drug. For the general public, halving this chance of recurrence is naturally regarded as the basis of a very important argument for taking the drug. However, it can be misleading given that the absolute risk of recurrence is low, and the risk of dying from a recurrence may be significantly lower than that. This situation may be further complicated when considering potentially life-threatening side effects of the drug, and perhaps the high cost of the drug. Thus, in this problem we have probabilities of recurrence, recurrence periods, cost per life saved, cost per patient, side effects (minor and major), as well as quality of life factors, to consider in argumentation.

While we can already capture this example, to some extent, in our logic-based framework, we are not drawing out the general features of the situation. There are many examples of drugs that can be considered in terms of probabilities of recurrence, recurrence periods, cost per life saved, cost per patient, side effects, and quality of life. Therefore, what we require is a more generic model with these terms as "parameters," so that we can see the effect of changing the value of some parameters on the overall constellation of arguments and counterarguments, and furthermore, we can see these in a principled way that conforms to the use of probability theory.

9.2.4 Argumentation with Metalevel Knowledge

It is well established that metalevel information is important in argumentation. Priorities over formulae are the most studied form of metalevel information in argumentation. Many approaches rely on priorities when explicitly using logical formulae as support (e.g., PAFs [AC02] and DeLP [GS04]). Furthermore, Prakken has undertaken an extensive study of how priorities can be reasoned with within logic [Pra97]. However, there is a much wider and richer range of metalevel knowledge that occurs in real-world argumentation. Here we briefly consider how this kind of metalevel

(Legality warrant)	`X should not do Y` `because Y is illegal`

(Reciprocity warrant)	`X should do Y`
	`because Z asked X to do Y`
	`and X owes Z a favor`

(Beneficiary warrant)	`X should do Y` `because X will then get benefit Z`

Figure 9.1
Formats for some types of warrant [Sil96, Sil97].

representation and reasoning could be put into a more general context, drawing on more general research in metalevel knowledge representation and reasoning, and we consider some examples of metalevel argumentation including *argumentum ad verecundiam* (appeals to authority) and *argumentum ad hominem* (attacking the person) that could be put into a common formal setting.

We start by considering Toulmin's framework (which we introduced in chapter 1) and how this has been extended to better capture aspects of rhetorical argumentation by incorporating a range of types of warrant relationship [Sil96, Sil97]. We present some of these types of warrant in figure 9.1. The problem with this approach is that there is a lack of a formal definition for reasoning with such formats. Representing these different types of warrant can be useful, but we also need to undertake inferencing. This criticism includes the criticism we raised of Toulmin's approach. However, here with different types of warrant, it is unclear which should take precedence or what other information we are allowed to assume to resolve the conflict.

Formalizing the different types of warrant in a logic-based framework with priorities is an option (see, e.g., the framework of Prakken and Sartor [PS97, Pra97], which was discussed in section 8.2.3). However, we do need more. We should seek a rich axiomatization for defining the relationships that can hold between the constructs proposed in rhetorical argumentation. To understand this point, we should turn to the informal logics and argumentation community, where there is a substantial literature on the nature of argumentation that looks at questions like "what is a strong argument," and "what is a valid argument." This literature offers numerous directions relevant to developing richer frameworks for metalevel argumentation. For example, consider "appeals to authority" (*ad verecundiam*) here. In this, we equate "authority" with "expert authority" rather than "administrative authority." The former is authority based on knowledge, whereas the later is authority based on power.

Appealing to authority can be justifiable. For example, having a medical operation done is normally justifiable when recommended by a clinician, but it is not normally justifiable when recommended by, say, an architect. There is an approach to reasoning with inconsistent information, based on modal logics, that incorporates a notion of expertise, and in case of conflicts, prefers information from a source that is more expert [Cho95]. However, a difficulty with appealing to authority is determining when it is justifiable.

In [Wal89], various suggestions are made for when appealing to authority is appropriate, which we attempt to synthesize into the following criteria: (1) The expert is suitably qualified in the field, (2) ignoring the expert's advice may not be feasible or wise, (3) the expert offers sufficient confidence in the advice, (4) none of the expert's caveats have been overlooked, and (5) the expert's advice seems the best out of any competing sets of expert's advice.

These criteria suggest that a metalevel formalization of reasoning about experts and their advice would be desirable and that inferences from such a formalization could affect the object-level reasoning with arguments. This should not just be reducing reasoning about expertise to a preference over sources of information but should support reasoning with these criteria explicitly. In other words, these criteria should be formulae in the knowledgebase, perhaps as metalevel formulae, and used in the premises in arguments and, thereby, be subject to undercutting where appropriate.

In a metalevel language, terms for metalevel predicates include object-level formulae. Therefore, if we use classical logic for both the object language and the metalanguage, we can use the metalanguage to represent knowledge about the object level. We can reason about the object language in the metalanguage. This means we reason about arguments and the source of arguments in the metalanguage. Thus, for example, we may represent the formats in figure 9.1, and the criteria above for when to appeal to authority, as metalevel formulae. For more information on metalevel formulae, see [BBF$^+$96].

Another example of metalevel reasoning arising in argumentation is *argumentum ad hominem*. Normally, logicians do not regard such arguments as acceptable. Yet constantly they occur in the real world. Political debates and courtroom trials are situations that routinely incorporate such arguments. Since they occur in the real world, we need to deal with them in artificial intelligence formalisms.

There are a wide variety of further types of argument that have been described as problematical, unacceptable, or invalid, such as red herrings, straw men, and sacred cows (for a review, see [Gul06]). Again, formal accounts of argumentation should be able to discriminate and account for these diverse forms of argumentation. For example, we may wish to construct an intelligent agent that has, among its various cognitive facilities, the ability to identify an argument given to it by another agent as being a red herring or a straw man if that is indeed the case.

9.2.5 Persuasiveness of Argumentation

Persuasional argumentation is a very important and common kind of argumentation. Consider, for example, how a prosecution lawyer tries to persuade a jury to condemn the accused, how a salesperson tries to persuade a potential customer to buy the product, how a child tries to persuade his or her parents to give him or her more pocket money, or how a politician tries to persuade the electorate to vote for him or her.

While in this book we have considered some key elements of persuasional argumentation, not least consideration of the audience in terms of empathy/antipathy, and resonance of arguments, we still require a more comprehensive framework for characterizing persuasional argumentation in terms of outcomes in the audience. In other words, for an argument to be persuasive, the recipient has to be persuaded to change his or her commitments. A change in commitments may be the addition of a commitment (e.g., *to buy something* or *to vote for somebody* or *to do something*), or it may be the deletion of a commitment or the reprioritization of commitments. Thus, in order to evaluate persuasive arguments, we need a systematic way of determing how an argument can change beliefs and commitments in an audience.

A richer model of the recipient seems necessary, where the recipient is an intelligent agent with, perhaps, beliefs, desires, and intentions, so that commitments are defined in terms of desires and intentions. The conceptualizations of argumentative attitudes, discussed in section 8.5, particularly the framework for rewards and threats by Amgoud and Prade [AP05a, AP05b], offer some interesting starting points. However, there is still a gap in determining what would be a persuasive argument. An account of persuasiveness should also consider actual outcomes, not just what the proponent thinks the recipient would regard as persuasive.

Further complications include how the arguments are presented and by whom. A reason can be inappropriate for an audience (e.g., *there is no point in telling a young child to take care with a vase because it is* *expensive*), and a proponent can be inappropriate for an audience (e.g., *not sounding like a rebel, a government minister is the wrong person to tell teenagers to use a helmet when riding a mountain bike*). It seems difficult to conceive how these aspects of argumentation could be accommodated with the current state of the art for argumentation systems.

9.3 Toward Dialogical Argumentation

The emphasis of this book has been on formalizing features of monological argumentation as opposed to dialogical argumentation. We have therefore given little consideration in the book to dialogical argumentation as it arises in multi-agent systems. The emphasis of the dialogical view is on the interactions between agents involved in argumentation, and this raises numerous further complications and issues for formalizing argumentation. A set of entities or agents interact to construct arguments for and against a particular claim. If an agent offers an argument, one or more of the other agents may dispute the argument, or they may concede that the argument is a good argument. Agents may use strategies to persuade the other agents to draw some conclusion on the basis of the assembled arguments.

Dialogical argumentation is undertaken via dialogue. The existence of different kinds of dialogue leads to a range of different kinds of argumentation. There is no definitive classification of dialogue types. However, dialogues for negotiation, dialogues for persuasion, and dialogues for seeking information have been particularly important goals for a lot of research into multi-agent argumentation.

For negotiation, there are a number of proposals for formalizations for multi-agent systems (see, e.g., [RRJ$^+$03, PWA03, STT01, AMP00]). To support some kinds of negotiations, formalizations of problem solving and decision making have been proposed [AP05a, AP05b].

Key notions from monological argumentation, such as undercut and warrant, have been adapted for use in formalizations for dialogical argumentation. There are a number of further elements of argumentation presented in this book that could be harnessed for dialogical argumentation, such as analysis of intrinsic and extrinsic factors and, in particular, formalizations of the audience to allow for higher impact and more believable arguments to be used by agents. Nonetheless, there remain many aspects of dialogical argumentation to be conceptualized and formalized.

A Table of Uniform Notation

Cardinality of a set X	$\|X\|$
Powerset of a set X	$\wp(X)$
Atom/Formula	$\alpha, \beta, \gamma, \delta, \alpha_1, \ldots, \beta_1, \ldots, \delta_1, \ldots$
Atom (used in "real-world" examples)	$a, b, c, \ldots\ldots$
Set of all atoms	$\mathcal{AT}$
Set of all formulae	$\mathcal{L}$
Set of atoms	Π
Set of formulae	Δ, Φ, Ψ, Γ
Knowledgebase (a set of formulae)	Δ
Beliefbase (a set of formulae)	Λ
Set of concerns (a set of formulae)	Θ_η
Concernbase	η
Set of atoms in a formula α	$\mathsf{Atoms}(\alpha)$
Truth values	$\mathtt{T}$, $\mathtt{F}$
Interpretation/model	w
Restricted set of models	$M(\Phi, \Pi)$
Classical consequence closure	Cn
Classical consequence relation	$\vdash$
Argument $\langle \Phi, \alpha \rangle$	$A, A', A_1, \ldots$
Set of all arguments $\langle \Phi, \alpha \rangle$	Ω
Tree	$T, T', T_1, \ldots$
Pros in argument structure	$\mathcal{P}$
Cons in argument structure	$\mathcal{C}$
Degree of conflict	C

B Review of Trees

If Σ is a set of symbols, the *set of strings over* Σ, denoted Σ^*, is the free monoid over Σ with operation concatenation, denoted by juxtaposition of symbols. The identity of the monoid is the empty string, written ε. If $s \in \Sigma^*$, the *length* of s, denoted $|s|$, is the number of symbols in s. In particular, $|\varepsilon| = 0$. The set of prefixes of a string $s \in \Sigma^*$ is $\mathit{prefix}(s) = \{u \in \Sigma^* \mid \exists v \in \Sigma^*, s = uv\}$.

Definition B.0.1 A **tree** is a triple (A, Σ, W) where

1. A is any set.
2. Σ is a set of symbols in a one-to-one correspondence with A.
3. W is a nonempty subset of the free monoid over Σ, in symbols $W \subseteq \Sigma^*$, such that

(i) $\bigcup_{w \in W} \mathit{prefix}(w) \subseteq W$.

(ii) $v \in \bigcap_{w \in W} \mathit{prefix}(w)$ for a unique $v \in \Sigma$.

W is the **skeleton** of the tree labeled with A. The tree is **finite** iff W is finite. If $u \in W$ and $u \neq \varepsilon$, then u is a **node** of the tree. If u is a node of the tree and there is no $v \in \Sigma$ such that $uv \in W$, then u is a **leaf** of the tree.

For v in (ii), v is the **root** of the tree. For each node u of the tree and each $v \in \Sigma$, if $uv \in W$, then u is the **parent node** of uv and uv is a **child node** of u. If u and w are distinct nodes of the tree such that $u \in \mathit{prefix}(w)$, then u is an **ancestor node** of w. A **branch** of the tree is a subset W' of W such that $W' = \mathit{prefix}(w)$ for some leaf w of the tree. The **length** of a finite branch is the length of the longest node on the branch. The **depth** of a finite tree is the length of its longest branch. If w is a node of the tree, the **subtree rooted at** w is the tree $(A, \Sigma, W' \cup \{\varepsilon\})$ such that $W' = \{vx \mid uvx \in W\}$ where $w = uv$ for some $v \in \Sigma$.

C Review of Classical Logic

Classical logic provides an ideal approach to knowledge representation and reasoning for many applications in computer science. Though there are some shortcomings, which have motivated a plethora of alternative formalisms, classical logic nonetheless constitutes an excellent starting point for knowledge representation and reasoning—not least because it provides a standard with which to compare alternatives. In this appendix, we review some of the features of classical logic, focusing on the propositional case.

C.1 Language of Classical Propositional Logic

A formal language for classical propositional logic is based on an arbitrary set of propositional symbols, called *atoms*. They can be interpreted by any declarative statement, and they can take either of the two truth values, *false* or *true*. As an illustration, if the propositional symbols include $\alpha, \beta, \gamma, \ldots$ among others, then these can be read as any statement, for example,

α: It rains.

α: Anna is married.

$\vdots$

All that matters is that all occurrences of the same symbol be interpreted by the same statement (e.g., $\alpha \vee \alpha$ cannot be read as "It rains, or it snows"—but $\alpha \vee \beta$ can).

More complex statements are expressed by means of more complex formulae using connectives.

Definition C.1.1 The set of **logical connectives** is conjunction, denoted $\wedge$; disjunction, denoted $\vee$; negation, denoted $\neg$; implication, denoted $\rightarrow$; and equivalence, denoted $\leftrightarrow$.

For example, if α is read as "It rains," then $\neg\alpha$ is read as "It does not rain." Or, if α is read as "It rains" and β is read as "It snows," then $\alpha \wedge \neg\beta$ is read as "It rains, and it does not snow."

As already mentioned, statements can either be *true* or *false*. Whether a statement is true or false usually depends on what the circumstances are, but some statements are always true and some are always false. In order to represent them in a generic way, two special symbols are introduced.

Definition C.1.2 Let $\top$ denote **truth** and $\bot$ denote **falsity**.

Thus, $\top$ is always true, and $\bot$ is always false.

Definition C.1.3 Let $\mathscr{L}$ be the set of **classical propositional formulae** formed from the set of atoms, denoted $\mathscr{AT}$, and the logical connectives: All atoms, as well as $\top$ and $\bot$, are formulae; if α and β are formulae, then $(\alpha \wedge \beta)$ is a formula, $(\alpha \vee \beta)$ is a formula, $(\alpha \to \beta)$ is a formula, $(\alpha \leftrightarrow \beta)$ is a formula, and $(\neg\alpha)$ is a formula. As no formula exists that would not be obtained in this way, $\mathscr{L}$ is the **language of classical propositional logic**. If no ambiguity arises, parentheses can be omitted.

Here are some examples of propositional formulae:

$\alpha \leftrightarrow \beta$

$\neg(\alpha \to \neg\beta)$

$\alpha \to (\beta \wedge \gamma \wedge \delta)$

For readability reasons, it often happens that atoms are taken to be of a mnemonic form, for example,

`BoeingMakesPlanes`

`VenezuelaExportsOil`

The fact that these refer to propositions that are notoriously true does *not* mean that these formulae are true. Similarly, for instance, the following two atoms need not be false:

`BrusselsIsCapitalOfUSA`

`MadridIsCapitalOfAustralia`

Indeed, ascribing `MadridIsCapitalOfAustralia`, for example, the (truth) value true would be a mischievous but lawful move according to classical logic.

Definition C.1.4 For each formula $\alpha \in \mathscr{L}$, the set of all atoms that occur in α is denoted $\mathrm{Atoms}(\alpha)$.

Definition C.1.5 For each atom $\alpha \in \mathcal{AT}$, either of α and $\neg\alpha$ is a **literal**. For $\alpha_1, \ldots, \alpha_n \in \mathcal{L}$, $\alpha_1 \vee \cdots \vee \alpha_n$ is a **clause** iff each of $\alpha_1, \ldots, \alpha_n$ is a literal.

Definition C.1.6 For $\alpha_1, \ldots, \alpha_n \in \mathcal{L}$, $\alpha_1 \vee \cdots \vee \alpha_n$ is a **disjunctive normal form** (DNF) iff each of $\alpha_1, \ldots, \alpha_n$ is a conjunction of literals.

Definition C.1.7 For $\alpha_1, \ldots, \alpha_n \in \mathcal{L}$, $\alpha_1 \wedge \cdots \wedge \alpha_n$ is a **conjunctive normal form** (CNF) iff each of $\alpha_1, \ldots, \alpha_n$ is a clause.

For any $\alpha \in \mathcal{L}$, a CNF, and similarly DNF, of α can be obtained by repeated applications of distributivity, double negation elimination, and De Morgan's laws (these properties are given below).

C.2 Semantics for Classical Propositional Logic

The semantics for classical logic is based on the truth values "true" and "false," denoted `T` and `F`, respectively. For example, assuming that the atoms `BoeingMakesPlanes` and `VenezuelaExportsOil` are both true (i.e., they have been given the truth value `T`), then the following formulae, for example, are true:

`BoeingMakesPlanes` $\wedge$ `VenezuelaExportsOil`

`BoeingMakesPlanes` $\vee$ `VenezuelaExportsOil`

However, the following formula is false:

$\neg$`BoeingMakesPlanes` $\vee$ $\neg$`VenezuelaExportsOil`

More generally, the evaluation of truth and falsity is extended to complex formulae by means of truth tables. For the truth table for negation (table C.1), if we know the truth value for α (the left column), then we can calculate the truth value for $\neg\alpha$ (the right column), and vice versa. For the truth table for conjunction (table C.2), if we know the value of α and β, then we can calculate the value for their conjunction. The truth table for the other connectives (tables C.3–C.5) are used similarly.

Table C.1
Truth table for negation

α	$\neg\alpha$
T	F
F	T

Table C.2
Truth table for conjunction

α	β	$\alpha \wedge \beta$
T	T	T
T	F	F
F	T	F
F	F	F

Table C.3
Truth table for disjunction

α	β	$\alpha \vee \beta$
T	T	T
T	F	T
F	T	T
F	F	F

Table C.4
Truth table for implication

α	β	$\alpha \rightarrow \beta$
T	T	T
T	F	F
F	T	T
F	F	T

Table C.5
Truth table for equivalence

α	β	$\alpha \leftrightarrow \beta$
T	T	T
T	F	F
F	T	F
F	F	T

Actually, if we know the truth value of all atoms in a formula, then we know the truth value of the formula.

Definition C.2.1 An **interpretation** for classical propositional logic is a function from $\mathscr{AT}$ into $\{\mathtt{T}, \mathtt{F}\}$. An interpretation w can be extended to any formula in $\mathscr{L}$ using the following equivalences given by the truth tables:

$$
\begin{array}{lll}
w(\alpha \wedge \beta) = \mathtt{T} & \text{iff} & w(\alpha) = w(\beta) = \mathtt{T} \\
w(\alpha \wedge \beta) = \mathtt{F} & \text{iff} & w(\alpha) = \mathtt{F} \text{ or } w(\beta) = \mathtt{F} \\
w(\alpha \vee \beta) = \mathtt{T} & \text{iff} & w(\alpha) = \mathtt{T} \text{ or } w(\beta) = \mathtt{T} \\
w(\alpha \vee \beta) = \mathtt{F} & \text{iff} & w(\alpha) = w(\beta) = \mathtt{F} \\
w(\neg\alpha) = \mathtt{T} & \text{iff} & w(\alpha) = \mathtt{F} \\
w(\neg\alpha) = \mathtt{F} & \text{iff} & w(\alpha) = \mathtt{T} \\
w(\alpha \rightarrow \beta) = \mathtt{T} & \text{iff} & w(\alpha) = \mathtt{F} \text{ or } w(\beta) = \mathtt{T} \\
w(\alpha \rightarrow \beta) = \mathtt{F} & \text{iff} & w(\alpha) = \mathtt{T} \text{ and } w(\beta) = \mathtt{F} \\
w(\alpha \leftrightarrow \beta) = \mathtt{T} & \text{iff} & w(\alpha) = w(\beta) \\
w(\alpha \leftrightarrow \beta) = \mathtt{F} & \text{iff} & w(\alpha) \neq w(\beta)
\end{array}
$$

It is often convenient to identify an interpretation with its characteristic set, consisting of the atoms that are assigned true: $w = \{\alpha \in \mathscr{AT} \mid w(\alpha) = \mathtt{T}\}$.

Definition C.2.2 An interpretation is a **model** for a formula α iff $w(\alpha) = \mathtt{T}$. If an interpretation w is a model for a formula α, then w **satisfies** α, denoted $w \models \alpha$. An interpretation is a model for a set of formulae $\Delta \in \wp(\mathscr{L})$ iff for each formula $\alpha \in \Delta$, $w(\alpha) = \mathtt{T}$. A set of formulae Δ is **consistent** iff Δ has a model.

For small examples, we can easily represent all the interpretations by a truth table. Each row in the truth table is an interpretation, and we can easily check whether an interpretation is a model. For a set of formulae Δ, where $|\mathtt{Atoms}(\Delta)| = n$, the truth table for Δ will have 2^n rows.

Example C.2.1 Consider $\Delta = \{\neg\alpha, \alpha \vee \beta\}$. There are four interpretation for Δ as represented in table C.6. The interpretation in the third row is a model for Δ, since both $\neg\alpha$ and $\alpha \vee \beta$ are both true in it.

Definition C.2.3 A **tautology**, or synonymously a **valid formula**, is a formula that every interpretation satisfies. A tautology can be represented by the abbreviation $\top$.

Table C.6
Truth table for example C.2.1

α	β	$\neg\alpha$	$\alpha \vee \beta$
T	T	F	T
T	F	F	T
F	T	T	T
F	F	T	F

Thus, a tautology is a formula whose every interpretation is a model. Equivalently, a tautology is a formula α that is always true. In symbols, $\top \models \alpha$, which can also be written $\models \alpha$.

Example C.2.2 Tautologies include $\alpha \vee \neg\alpha$, $\alpha \rightarrow \alpha$, and $\beta \vee \alpha \vee \neg\alpha \vee \gamma$.

Definition C.2.4 A **contradiction**, or synonymously an **inconsistent formula**, is a formula that no interpretation satisfies. A contradiction can be represented by the abbreviation $\perp$.

Thus, there is no model for a contradiction α (and so no model for a set of formulae that contains an inconsistent formula). In symbols, $\alpha \models \perp$.

Example C.2.3 Contradictions include $\alpha \wedge \neg\alpha$, $\alpha \leftrightarrow \neg\alpha$, $(\alpha \wedge \neg\alpha) \wedge (\alpha \rightarrow \beta)$, and $\neg\alpha \wedge (\alpha \vee \beta) \wedge \neg\beta$.

Of course, a consistent formula is a formula α that is not a contradiction: $\alpha \not\models \perp$.

Definition C.2.5 The **entailment relation** $\models \; \subseteq \wp(\mathscr{L}) \times \mathscr{L}$ is defined as follows for $\Delta \subseteq \mathscr{L}$ and for $\alpha \in \mathscr{L}$:

$\Delta \models \alpha$ iff for all models w of Δ, $w \models \alpha$

Proposition C.2.1 For $\Delta \subseteq \mathscr{L}$, and $\alpha, \beta, \gamma \in \mathscr{L}$, the following properties hold:

$\Delta \models \alpha \vee \beta$ iff $\Delta \models \beta \vee \alpha$ [Commutativity]
$\Delta \models \alpha \wedge \beta$ iff $\Delta \models \beta \wedge \alpha$

$\Delta \models (\alpha \wedge \beta) \wedge \gamma$ iff $\Delta \models \alpha \wedge (\beta \wedge \gamma)$ [Associativity]
$\Delta \models (\alpha \vee \beta) \vee \gamma$ iff $\Delta \models \alpha \vee (\beta \vee \gamma)$

$\Delta \models \alpha \vee (\beta \wedge \gamma)$ iff $\Delta \models (\alpha \vee \beta) \wedge (\alpha \vee \gamma)$ [Distributivity]
$\Delta \models \alpha \wedge (\beta \vee \gamma)$ iff $\Delta \models (\alpha \wedge \beta) \vee (\alpha \wedge \gamma)$

$\Delta \models \neg(\alpha \vee \beta)$ iff $\Delta \models \neg\alpha \wedge \neg\beta$ [De Morgan's laws]
$\Delta \models \neg(\alpha \wedge \beta)$ iff $\Delta \models \neg\alpha \vee \neg\beta$

$\Delta \models \alpha$ iff $\Delta \models \neg\neg\alpha$ [Double negation elimination]

$\Delta \models \alpha \wedge \alpha$ iff $\Delta \models \alpha$ [Idempotency]

$\Delta \models \alpha \vee \alpha$ iff $\Delta \models \alpha$

Example C.2.4 Let $\Delta = \{\alpha \vee \beta, \neg\alpha\}$, where $\alpha \vee \beta, \neg\alpha \in \mathscr{L}$. There are four interpretations as represented in table C.6. Only the interpretation in the third row satisfies both formulae in Δ. Furthermore, β is true in this interpretation, and so $\Delta \models \beta$.

An inference can be described as valid if there is no possible situation in which its premises are all true and its conclusion is not true. We can formalize this as follows.

Definition C.2.6 For $\Delta \subseteq \mathscr{L}$ and $\alpha \in \mathscr{L}$, α is a **valid inference** from Δ iff $\Delta \models \alpha$.

It sometimes is useful to restrict models to a given subset of all atoms.

Definition C.2.7 For $\Pi \subseteq \mathscr{AT}$ and $\Phi \subseteq \mathscr{L}$, define

$$\mathsf{Models}(\Phi, \Pi) = \{w \in \wp(\Pi) \mid \forall\varphi \in \Phi, w \models \varphi\}.$$

While a given formula can be arbitrarily complicated, we have a formal semantics that provides a "meaning" of the formula in terms of the meaning of the components of the formula. However, formal semantics do not provide meaning in themselves. There is still a call to the user's intuition to understand the meaning of the formal semantics. In other words, formal semantics are only a bridge between a possibly complicated formal language and the concepts in the real world with which they may be associated. Of course this is subjective. Some people might find a particular semantics transparent, where others might not.

C.3 Consequence Relation

To help us develop our formalisms for knowledge representation and reasoning, we need some alternative ways to consider inference. A proof theory is a system for manipulating formulae that defines a relation $\vdash \subseteq \wp(\mathscr{L}) \times \mathscr{L}$.

By the **classical consequence relation** $\vdash$, we mean any proof theory such that for all $\Delta \subseteq \mathscr{L}$ and $\alpha \in \mathscr{L}$, $\Delta \models \alpha$ iff $\Delta \vdash \alpha$.

Proposition C.3.1 Properties of the consequence relation include the following:

$\Delta \cup \{\alpha\} \vdash \alpha$ [Reflexivity]

$\Delta \vdash \alpha$ implies $\Delta \cup \{\beta\} \vdash \alpha$ [Unit monotonicity]

$\Delta \cup \{\alpha\} \vdash \beta$ and $\Gamma \vdash \alpha$ implies $\Delta \cup \Gamma \vdash \beta$ [Unit cut]

$\Delta \cup \{\alpha\} \vdash \beta$ implies $\Delta \vdash \alpha \rightarrow \beta$ [Conditionalization]

$\Delta \vdash \alpha \rightarrow \beta$ implies $\Delta \cup \{\alpha\} \vdash \beta$ [Deduction]

$\vdash \alpha \rightarrow \beta$ and $\Delta \vdash \alpha$ implies $\Delta \vdash \beta$ [Right weakening]

$\Delta \vdash \alpha$ and $\Delta \vdash \beta$ implies $\Delta \cup \{\alpha\} \vdash \beta$ [Unit cautious monotonicity]

$\Delta \cup \{\alpha\} \vdash \gamma$ and $\vdash \alpha \leftrightarrow \beta$ implies $\Delta \cup \{\beta\} \vdash \gamma$ [Left logical equivalence]

$\Delta \cup \{\alpha\} \vdash \gamma$ and $\Delta \cup \{\beta\} \vdash \gamma$ implies $\Delta \cup \{\alpha \vee \beta\} \vdash \gamma$ [Or]

$\Delta \vdash \alpha \wedge \beta$ iff $\Delta \vdash \alpha$ and $\Delta \vdash \beta$ [And]

Conditionalization and deduction are often referred to as a single united principle known as the *deduction theorem*.

Definition C.3.1 The **classical consequence closure** for a set of formulae Δ is denoted $\mathsf{Cn}(\Delta)$ and is defined as follows: $\mathsf{Cn}(\Delta) = \{\alpha \mid \Delta \vdash \alpha\}$.

Proposition C.3.2 Properties of classical consequence closure include the following:

$\Delta \subseteq Cn(\Delta)$ [Inclusion]

$Cn(Cn(\Delta)) = Cn(\Delta)$ [Idempotence]

$Cn(\Delta) = \mathscr{L}$ iff Δ is inconsistent [Trivialization]

$\Gamma \subseteq \Delta \subseteq Cn(\Gamma)$ implies $Cn(\Delta) \subseteq Cn(\Gamma)$ [Restricted cut]

$\Gamma \subseteq \Delta \subseteq Cn(\Gamma)$ implies $Cn(\Gamma) \subseteq Cn(\Delta)$ [Cautious monotonicity]

$\Gamma \subseteq \Delta \subseteq Cn(\Gamma)$ implies $Cn(\Gamma) = Cn(\Delta)$ [Cumulativity]

$\Gamma \subseteq \Delta$ implies $Cn(\Gamma) \subseteq Cn(\Delta)$ [Monotonicity]

C.4 Classical Predicate Logic

In this book we work mainly with classical propositional logic. However, as demonstrated in chapter 3, key parts of the logic-based framework for argumentation can be generalized for use with classical predicate logic (equivalently called classical first-order logic). Here we just briefly remind the reader of the basic notation of classical predicate logic. The development of classical predicate logic comes from the need for a more general language that captures relations, variables, functions, and quantifiers.

Using predicate logic, we present statements using the logical connectives introduced for classical propositional logic, together with predicate symbols (relation symbols), and the symbols $\forall$, denoting *For all*, and $\exists$, denoting *There exists*. The $\forall$ symbol is called the **universal quantifier**, and the $\exists$ symbol is called the **existential quantifier**. We give a few intuitive illustrations below:

$\forall \texttt{X}\ \texttt{human}(\texttt{X}) \rightarrow \texttt{male}(\texttt{X}) \vee \texttt{female}(\texttt{X})$

$\exists \texttt{X}\ \texttt{President-of-the-USA}(\texttt{X})$

$\forall \texttt{X}\ \texttt{human}(\texttt{X}) \rightarrow \exists \texttt{Y}\ \texttt{mother}(\texttt{Y}, \texttt{X})$

$\forall \texttt{X}\ \texttt{even}(\texttt{X}) \rightarrow \texttt{odd}(\texttt{successor}(\texttt{X}))$

Thus, by using the notion of relations in classical logic, we are generalizing on the notion of a proposition. A relation is a way of adding structure to the knowledge representation. This generalization can be further extended by the use of variables and quantifiers.

Given that most of this book can be understood without a detailed knowledge of predicate logic, we will not review predicate logic further in this appendix and rather refer the interested reader to one of the many introductory textbooks on predicate logic.

D Proofs for Results in Book

Proposition 3.2.1 $\langle\Phi, \alpha\rangle$ is an argument iff $\Phi \cup \{\neg\alpha\}$ is a minimal inconsistent set.

Proof ($\Rightarrow$) Assume that $\langle\Phi, \alpha\rangle$ is an argument. Thus, Φ is a minimal subset of Δ such that $\Phi \vdash \alpha$. That is, $\Phi \vdash \alpha$ and $\Phi' \nvdash \alpha$ for every proper subset Φ' of Φ. By classical logic, $\Phi \cup \{\neg\alpha\}$ is an inconsistent set and every proper subset Φ' of Φ is such that $\Phi' \cup \{\neg\alpha\}$ is a consistent set. Now, $\Phi \nvdash \bot$ in the definition of an argument means that every (proper or improper) subset of Φ is a consistent set. Thus, every proper subset of $\Phi \cup \{\neg\alpha\}$ is a consistent set. ($\Leftarrow$) Assume that $\Phi \cup \{\neg\alpha\}$ is a minimal inconsistent set. Then, Φ is a consistent set (in symbols, $\Phi \nvdash \bot$). Also, every proper subset Φ' of Φ is such that $\Phi' \cup \{\neg\alpha\}$ is a consistent set. By classical logic, $\Phi' \nvdash \alpha$ ensues for every proper subset Φ' of Φ. Lastly, that $\Phi \cup \{\neg\alpha\}$ is an inconsistent set yields $\Phi \vdash \alpha$, according to classical logic. ■

Proposition 3.2.2 Let $\langle\Phi, \alpha\rangle$ be an argument. If $\Phi \vdash \alpha \rightarrow \beta$ and $\beta \rightarrow \alpha$ is a tautology, then $\langle\Phi, \beta\rangle$ is also an argument.

Proof By definition of an argument, $\Phi \nvdash \bot$ (thus taking care of the first requirement for $\langle\Phi, \beta\rangle$ to be an argument) and $\Phi \vdash \alpha$. Due to the assumption, $\Phi \vdash \alpha \rightarrow \beta$. By classical logic, $\Phi \vdash \beta$. Thus, the second requirement is met as well. There remains to show that each proper subset Ψ of Φ is such that $\Psi \nvdash \beta$. Assume the contrary: In symbols, $\Psi \vdash \beta$ for some $\Psi \subset \Phi$. This yields $\Psi \vdash \alpha$ using classical logic in view of the assumption that $\beta \rightarrow \alpha$ is a tautology. This contradicts the fact that Φ is a minimal subset of Δ entailing α (cf. the definition of $\langle\Phi, \alpha\rangle$ being an argument). ■

Proposition 3.2.3 Let Φ and Ψ be such that there exists a bijection f from Ψ to some partition $\{\Phi_1, \ldots, \Phi_n\}$ of Φ where $Cn(\{\psi\}) = Cn(f(\psi))$ for all $\psi \in \Psi$. If $\langle\Phi, \alpha\rangle$ is an argument then $\langle\Psi, \alpha\rangle$ is also an argument.

Proof In view of f being bijective, $\Psi = \{\psi_1, \dots, \psi_n\}$ with $f(\psi_i) = \Phi_i$ for $i = 1 \dots n$. Since Cn is a closure operation, $Cn(\{\psi_1\} \cup \dots \cup \{\psi_n\}) = Cn(f(\psi_1) \cup \dots \cup f(\psi_n))$ as $Cn(\{\psi_i\}) = Cn(f(\psi_i))$ for $i = 1 \dots n$. Thus, $Cn(\Psi) = Cn(\Phi)$ (remember, $\{\Phi_1, \dots, \Phi_n\}$ is a partition of Φ). Then, $\Psi \nvdash \bot$ and $\Psi \vdash \alpha$. Similarly, $Cn(\{\psi_1, \dots, \psi_{i-1}, \psi_{i+1}, \dots, \psi_n\}) = Cn(\Phi_1 \cup \dots \cup \Phi_{i-1} \cup \Phi_{i+1} \cup \dots \cup \Phi_n)$. However, $\alpha \notin Cn(\Phi_1 \cup \dots \cup \Phi_{i-1} \cup \Phi_{i+1} \cup \dots \cup \Phi_n)$ because $\Phi \neq \Phi_1 \cup \dots \cup \Phi_{i-1} \cup \Phi_{i+1} \cup \dots \cup \Phi_n$ (as $\{\Phi_1, \dots, \Phi_n\}$ is a partition of Φ) and because Φ is a minimal subset of Δ such that $\Phi \vdash \alpha$. Hence, $\alpha \notin Cn(\{\psi_1, \dots, \psi_{i-1}, \psi_{i+1}, \dots, \psi_n\})$. That is, every proper subset of Ψ fails to entail α. Accordingly, Ψ is a minimal subset of Δ such that $\Psi \vdash \alpha$. ∎

Corollary 3.2.1 Let $\Phi = \{\phi_1, \dots, \phi_n\} \subseteq \Delta$ and $\Psi = \{\psi_1, \dots, \psi_n\} \subseteq \Delta$ such that $\phi_i \leftrightarrow \psi_i$ is a tautology for $i = 1 \dots n$. Let α and β be such that $\alpha \leftrightarrow \beta$ is a tautology. Then, $\langle \Phi, \alpha \rangle$ is an argument iff $\langle \Psi, \beta \rangle$ is an argument.

Proof Combine proposition 3.2.2 and proposition 3.2.3. ∎

Proposition 3.2.4 If $\langle \Phi, \alpha \rangle$ is more conservative than $\langle \Psi, \beta \rangle$, then $\Psi \backslash \Phi \vdash \phi \rightarrow (\alpha \rightarrow \beta)$ for some formula ϕ such that $\Phi \vdash \phi$ and $\phi \nvdash \alpha$ unless α is a tautology.

Proof Since $\langle \Phi, \alpha \rangle$ is an argument, Φ is finite and $\Phi \vdash \alpha$ so that Φ is logically equivalent to $\alpha \wedge (\alpha \rightarrow \phi')$ for some formula ϕ' (i.e., Φ is logically equivalent to $\alpha \wedge \phi$ where ϕ is $\alpha \rightarrow \phi'$). Since $\langle \Phi, \alpha \rangle$ is more conservative than $\langle \Psi, \beta \rangle$, $\Psi = \Phi \cup \Psi \backslash \Phi$. Since $\langle \Psi, \beta \rangle$ is an argument, $\Psi \vdash \beta$. Hence, $\Phi \cup \Psi \backslash \Phi \vdash \beta$. Then, $\{\alpha \wedge \phi\} \cup \Psi \backslash \Phi \vdash \beta$. It follows that $\Psi \backslash \Phi \vdash \phi \rightarrow (\alpha \rightarrow \beta)$. There only remains to show that $\Phi \vdash \phi$ (which is trivial) and that $\phi \nvdash \alpha$ unless α is a tautology. Assuming $\phi \vdash \alpha$ gives $\alpha \rightarrow \phi' \vdash \alpha$, but the latter means that α is a tautology. ∎

Proposition 3.2.5 Being more conservative defines a pre-ordering over arguments. Minimal arguments always exist, unless all formulae in Δ are inconsistent. Maximal arguments always exist: They are $\langle \emptyset, \top \rangle$ where $\top$ is any tautology.

Proof Reflexivity and transitivity result from the fact that these two properties are satisfied by set inclusion and logical consequence.

Easy as well is the case of $\langle \emptyset, \top \rangle$: An argument is maximal iff it is of this form because $\emptyset$ and $\top$ are extremal with respect to set inclusion and logical consequence, respectively. Assuming that some formula in Δ is consistent, Δ has at least one maximal consistent subset Θ. Since Δ is finite, so is Θ and there exists a formula α that Θ is logically equivalent with. Also, there is some minimal $\Phi \subseteq \Theta \subseteq \Delta$ such that Φ and Θ are logically equivalent. Clearly, Φ is consistent and Φ is a minimal subset of Δ such that $\Phi \vdash \alpha$. In other words, $\langle \Phi, \alpha \rangle$ is an argument. There only remains to show that it is minimal. Consider an argument $\langle \Psi, \beta \rangle$ such that $\langle \Phi, \alpha \rangle$ is more conservative than $\langle \Psi, \beta \rangle$. According to definition 3.2.2, $\Phi \subseteq \Psi$ and $\beta \vdash \alpha$. Since Φ is logically equivalent with a maximal consistent subset of Δ, it follows that Ψ is logically equivalent with Φ (because Ψ is a consistent subset of Δ by definition of an argument). Thus, α is logically equivalent with each of Φ and Ψ. As a consequence, $\Psi \vdash \beta$ and $\beta \vdash \alpha$ yield that α is logically equivalent with β, too. Since Φ is a minimal subset of Δ such that $\Phi \vdash \alpha$ (cf. above), it follows that Φ is a minimal subset of Δ such that $\Phi \vdash \beta$. However, Ψ is also a minimal subset of Δ such that $\Psi \vdash \beta$ (by definition of an argument). Hence, $\Phi = \Psi$ (due to $\Phi \subseteq \Psi$). In all, $\langle \Psi, \beta \rangle$ is more conservative than $\langle \Phi, \alpha \rangle$. Stated otherwise, we have just shown that if $\langle \Phi, \alpha \rangle$ is more conservative than $\langle \Psi, \beta \rangle$, then the converse is true as well. That is, $\langle \Phi, \alpha \rangle$ is minimal with respect to being more conservative (as applied to arguments). ■

Proposition 3.2.6 Given a normal form, being more conservative defines an ordering provided that only arguments that have a consequent in normal form are considered. The ordered set of all such arguments is an upper semilattice (when restricted to the language of Δ). The greatest argument always exists; it is $\langle \emptyset, \top \rangle$.

Proof Let $\langle \Phi, \alpha \rangle$ and $\langle \Psi, \beta \rangle$ be more conservative than each other. Clearly, $\Phi = \Psi$. Also, $\alpha \vdash \beta$ and $\beta \vdash \alpha$. That is, α and β are logically equivalent. Since α and β are in normal form, $\alpha = \beta$. Thus, antisymmetry holds, while reflexivity and transitivity follow from proposition 3.2.5.

Since Δ is finite, there are only a finite number of arguments $\langle \Omega, \gamma \rangle$ for each γ. Also, the language of Δ contains only a finite number of atomic formulae, and it allows for only a finite number of formulae that are not logically equivalent. That is, there are only a finite number of arguments if they are restricted to the language of Δ. Thus, all the upper bounds of two given arguments $\langle \Phi, \alpha \rangle$ and $\langle \Psi, \beta \rangle$ form a finite set $\{\langle \Omega_1, \gamma_1 \rangle, \ldots, \langle \Omega_n, \gamma_n \rangle\}$. For $i = 1 \ldots n$, $\Omega_i \subseteq \Phi$ and $\Omega_i \subseteq \Psi$. Hence, $\Omega_i \subseteq \Phi \cap \Psi$. Then, $\Phi \cap \Psi \vdash \gamma_i$ because $\langle \Omega_i, \gamma_i \rangle$ is an argument. It follows that $\Phi \cap \Psi \vdash \gamma_1 \wedge \cdots \wedge \gamma_n$ and an argument $\langle \Theta, \gamma_1 \wedge \cdots \wedge \gamma_n \rangle$ can be constructed where Θ is a minimal subset of $\Phi \cap \Psi$ such that $\Theta \vdash \gamma_1 \wedge \cdots \wedge \gamma_n$

(since Φ and Ψ are consistent by definition of an argument, Θ is consistent). For matter of convenience, $\gamma_1 \wedge \cdots \wedge \gamma_n$ is assumed to be in normal form, as doing so obviously causes no harm here. Clearly, every $\langle \Omega_i, \gamma_i \rangle$ is more conservative than $\langle \Theta, \gamma_1 \wedge \cdots \wedge \gamma_n \rangle$. This means that $\langle \Theta, \gamma_1 \wedge \cdots \wedge \gamma_n \rangle$ is the l.u.b. of $\langle \Phi, \alpha \rangle$ and $\langle \Psi, \beta \rangle$ (uniqueness of Θ is imposed by the usual property of orderings of allowing for only one l.u.b. of any subset).

As for the greatest argument, it is trivial that $\emptyset \subseteq \Phi$ and $\alpha \vdash \top$ for all Φ and α. ∎

Proposition 3.2.7 Two arguments are equivalent whenever each is more conservative than the other. In partial converse, if two arguments are equivalent, then either each is more conservative than the other or neither is.

Proof We only prove the second part. Consider two equivalent arguments $\langle \Phi, \alpha \rangle$ and $\langle \Psi, \beta \rangle$ such that $\langle \Phi, \alpha \rangle$ is more conservative than $\langle \Psi, \beta \rangle$. Of course, $\Phi \vdash \alpha$. According to the definition of equivalent arguments, α is logically equivalent with β. Thus, $\Phi \vdash \beta$. By definition of an argument, Ψ is a minimal subset of Δ satisfying $\Psi \vdash \beta$. Hence, $\Psi = \Phi$ because $\Phi \subseteq \Psi$ follows from the fact that $\langle \Phi, \alpha \rangle$ is more conservative than $\langle \Psi, \beta \rangle$. Finally, $\Psi = \Phi$ and α being logically equivalent with β make each of $\langle \Psi, \beta \rangle$ and $\langle \Phi, \alpha \rangle$ to be more conservative than the other. ∎

Proposition 3.3.1 If $\langle \Psi, \beta \rangle$ is a rebuttal for an argument $\langle \Phi, \alpha \rangle$, then $\langle \Psi, \beta \rangle$ is a defeater for $\langle \Phi, \alpha \rangle$.

Proof By definition of an argument, $\Phi \vdash \alpha$. By classical logic, $\neg\alpha \vdash \neg(\phi_1 \wedge \cdots \wedge \phi_n)$ where $\Phi = \{\phi_1, \ldots, \phi_n\}$. As $\beta \leftrightarrow \neg\alpha$ is a tautology, $\beta \vdash \neg(\phi_1 \wedge \cdots \wedge \phi_n)$ follows. ∎

Proposition 3.3.2 If $\langle \Psi, \beta \rangle$ is a defeater for $\langle \Phi, \alpha \rangle$, then there exists an undercut for $\langle \Phi, \alpha \rangle$ that is more conservative than $\langle \Psi, \beta \rangle$.

Proof By definition of a defeater, $\Psi \nvdash \bot$ and $\Psi \vdash \beta$ while $\beta \vdash \neg(\phi_1 \wedge \cdots \wedge \phi_n)$ for some $\{\phi_1, \ldots, \phi_n\} \subseteq \Phi$. There then exists a minimal subset $\Psi' \subseteq \Psi \subseteq \Delta$ such that $\Psi' \vdash \neg(\phi_1 \wedge \cdots \wedge \phi_n)$. Of course, $\Psi' \nvdash \bot$. Therefore, $\langle \Psi', \neg(\phi_1 \wedge \cdots \wedge \phi_n) \rangle$ is an argument, and it clearly is an undercut for $\langle \Phi, \alpha \rangle$. Verification that $\langle \Psi', \neg(\phi_1 \wedge \cdots \wedge \phi_n) \rangle$ is more conservative than $\langle \Psi, \beta \rangle$ is immediate. ∎

Corollary 3.3.1 If $\langle \Psi, \beta \rangle$ is a rebuttal for $\langle \Phi, \alpha \rangle$, then there exists an undercut for $\langle \Phi, \alpha \rangle$ that is more conservative than $\langle \Psi, \beta \rangle$. ∎

Proposition 3.3.3 If an argument has a defeater, then there exists an undercut for its defeater.

Proof Let $\langle \Psi, \beta \rangle$ be a defeater for $\langle \Phi, \alpha \rangle$. That is, $\Psi \vdash \beta$ and $\beta \vdash \neg(\phi_1 \wedge \cdots \wedge \phi_n)$ for some $\{\phi_1, \ldots, \phi_n\} \subseteq \Phi$. Writing Ψ as $\{\psi_1, \ldots, \psi_m\}$, we get $\{\psi_1, \ldots, \psi_m\} \vdash \neg(\phi_1 \wedge \cdots \wedge \phi_n)$. By classical logic, $\{\phi_1, \ldots, \phi_n\} \vdash \neg(\psi_1 \wedge \cdots \wedge \psi_m)$ and $\Phi \vdash \neg(\psi_1 \wedge \cdots \wedge \psi_m)$. Let $\Phi' \subseteq \Phi$ be a minimal subset entailing $\neg(\psi_1 \wedge \cdots \wedge \psi_m)$. Then, $\langle \Phi', \neg(\psi_1 \wedge \cdots \wedge \psi_m) \rangle$ is an argument, and it is an undercut for $\langle \Psi, \beta \rangle$. ∎

Corollary 3.3.2 If an argument A has at least one defeater, then there exists an infinite sequence of arguments $(A_n)_{n \in \omega^*}$ such that A_1 is A and A_{n+1} is an undercut of A_n for every $n \in \omega^*$. ∎

Proposition 3.3.4 Let $\langle \Phi, \alpha \rangle$ be an argument for which $\langle \Psi, \beta \rangle$ is a defeater. Then, $\Psi \not\subseteq \Phi$.

Proof Assume the contrary, $\Psi \subseteq \Phi$. By definition of a defeater, $\Psi \vdash \beta$ and $\beta \vdash \neg(\phi_1 \wedge \cdots \wedge \phi_n)$ for some $\{\phi_1, \ldots, \phi_n\} \subseteq \Phi$. Therefore, $\Psi \vdash \neg(\phi_1 \wedge \cdots \wedge \phi_n)$. Since $\Psi \subseteq \Phi$, monotonicity then yields $\Phi \vdash \neg(\phi_1 \wedge \cdots \wedge \phi_n)$. However, $\Phi \vdash \phi_1 \wedge \cdots \wedge \phi_n$ because $\Phi = \{\phi_1, \ldots, \phi_n\}$. That is, $\Phi \vdash \bot$, and this is contradicts the assumption that $\langle \Phi, \alpha \rangle$ is an argument. ∎

Proposition 3.3.5 If $\langle \Gamma, \neg\psi \rangle$ is an undercut for $\langle \Psi, \neg\phi \rangle$, which is itself an undercut for an argument $\langle \Phi, \alpha \rangle$, then $\langle \Gamma, \neg\psi \rangle$ is not a defeater for $\langle \Phi, \alpha \rangle$.

Proof We prove a little more, considering that $\langle \Psi, \neg\phi \rangle$ is a defeater of $\langle \Phi, \alpha \rangle$, not necessarily an undercut of it. Let $\langle \Phi, \alpha \rangle$. Consider $\langle \Psi, \neg(\varphi_1 \wedge \cdots \wedge \varphi_n) \rangle$ such that $\{\varphi_1, \ldots, \varphi_n\} \subseteq \Phi$ and $\langle \Gamma, \neg(\psi_1 \wedge \cdots \wedge \psi_m) \rangle$ such that $\{\psi_1, \ldots, \psi_m\} \subseteq \Psi$ (by the definition of a formula, $m \geq 1$). Assume that $\langle \Gamma, \neg(\psi_1 \wedge \cdots \wedge \psi_m) \rangle$ is a defeater for $\langle \Phi, \alpha \rangle$. That is, $\neg(\psi_1 \wedge \cdots \wedge \psi_m) \vdash \neg(\varphi_{i_1} \wedge \cdots \wedge \varphi_{i_r})$ for some $\{\varphi_{i_1}, \ldots, \varphi_{i_r}\} \subseteq \Phi$. Therefore, $\neg(\psi_1 \wedge \cdots \wedge \psi_m) \vdash \neg(\varphi_1 \wedge \cdots \wedge \varphi_p)$ where $\Phi = \{\varphi_1, \ldots, \varphi_p\}$. Since $\langle \Psi, \neg\phi \rangle$ is a defeater of $\langle \Phi, \alpha \rangle$, a similar reasoning gives $\Psi \vdash \neg(\varphi_1 \wedge \cdots \wedge \varphi_p)$. As $\Psi = \{\psi_1, \ldots, \psi_m, \psi_{m+1}, \ldots, \psi_q\}$ for some $q \geq m$, an immediate consequence is $\psi_1 \wedge \cdots \wedge \psi_m \wedge \psi_{m+1} \wedge \cdots \wedge \psi_q \vdash \neg(\varphi_1 \wedge \cdots \wedge \varphi_p)$. In view of $\neg(\psi_1 \wedge \cdots \wedge \psi_m) \vdash \neg(\varphi_1 \wedge \cdots \wedge \varphi_p)$, it then follows, by classical logic, that $\psi_{m+1} \wedge \cdots \wedge \psi_q \vdash \neg(\varphi_1 \wedge \cdots \wedge \varphi_p)$. That is, $\{\psi_{m+1}, \ldots, \psi_q\} \vdash \neg(\varphi_1 \wedge \cdots \wedge \varphi_p)$. By the definition of an argument, no proper subset of Ψ entail $\neg(\varphi_1 \wedge \cdots \wedge \varphi_n)$: Due to $m \geq 1$, a contradiction arises. ∎

Proposition 3.3.6 If $\langle \Psi, \beta \rangle$ is a maximally conservative defeater of $\langle \Phi, \alpha \rangle$, then $\langle \Psi, \beta' \rangle$ is an undercut of $\langle \Phi, \alpha \rangle$ for some β' that is logically equivalent with β.

Proof Let $\langle \Psi, \beta \rangle$ be a defeater for $\langle \Phi, \alpha \rangle$ such that for all defeaters $\langle \Psi', \beta' \rangle$ of $\langle \Phi, \alpha \rangle$, if $\Psi' \subseteq \Psi$ and $\beta \vdash \beta'$, then $\Psi \subseteq \Psi'$ and $\beta' \vdash \beta$. By definition of a defeater, $\beta \vdash \neg(\phi_1 \wedge \cdots \wedge \phi_n)$ for some $\{\phi_1, \ldots, \phi_n\} \subseteq \Phi$ while $\Psi \vdash \beta$ and $\Psi \nvdash \bot$. Then, $\langle \Psi', \neg(\phi_1 \wedge \cdots \wedge \phi_n) \rangle$ is clearly an argument whenever Ψ' is taken to denote a minimal subset of Ψ such that $\Psi' \vdash \neg(\phi_1 \wedge \cdots \wedge \phi_n)$. It is an undercut for $\langle \Phi, \alpha \rangle$. Therefore, it is a defeater for $\langle \Phi, \alpha \rangle$. Applying the assumption stated at the start of the proof, $\Psi = \Psi'$ and $\neg(\phi_1 \wedge \cdots \wedge \phi_n) \vdash \beta$. Thus, β is logically equivalent with $\neg(\phi_1 \wedge \cdots \wedge \phi_n)$ while $\langle \Psi, \neg(\phi_1 \wedge \cdots \wedge \phi_n) \rangle$ is an undercut for $\langle \Phi, \alpha \rangle$. ∎

Proposition 3.3.7 Given two arguments $\langle \Phi, \alpha \rangle$ and $\langle \Psi, \beta \rangle$ such that $\{\alpha, \beta\} \vdash \phi$ for each $\phi \in \Phi$, if $\langle \Psi, \beta \rangle$ is a defeater for $\langle \Phi, \alpha \rangle$, then $\langle \Phi, \alpha \rangle$ is a defeater for $\langle \Psi, \beta \rangle$.

Proof By assumption, $\Psi \vdash \beta$ and $\beta \vdash \neg(\phi_1 \wedge \cdots \wedge \phi_n)$ for some $\{\phi_1, \ldots, \phi_n\} \subseteq \Phi$. By classical logic, it follows that $\Psi \cup \Phi \vdash \bot$. Thus, $\Psi \cup \{\alpha\} \vdash \bot$ because $\Psi \vdash \beta$ and $\{\alpha, \beta\}$ entails Φ. Hence, $\alpha \vdash \neg(\psi_1 \wedge \cdots \wedge \psi_m)$ where $\{\psi_1, \ldots, \psi_m\} = \Psi$. ∎

Corollary 3.3.3 Let α be logically equivalent with Φ. If $\langle \Psi, \beta \rangle$ is a defeater for $\langle \Phi, \alpha \rangle$, then $\langle \Phi, \alpha \rangle$ is a defeater for $\langle \Psi, \beta \rangle$.

Proof Since $\langle \Psi, \beta \rangle$ is a defeater for $\langle \Phi, \alpha \rangle$, we have $\beta \vdash (\phi_1, \ldots, \phi_n)$ for some $\{\phi_1, \ldots, \phi_n\} \subseteq \Phi$. By classical logic, $\Phi \cup \{\beta\} \vdash \bot$ and thus $\{\alpha, \beta\} \vdash \bot$ because α is logically equivalent with Φ. By classical logic, $\{\alpha, \beta\} \vdash \phi$ for each $\phi \in \Phi$, and Proposition 3.3.7 applies. ∎

Corollary 3.3.4 If $\langle \Psi, \beta \rangle$ is a rebuttal for $\langle \Phi, \alpha \rangle$, then $\langle \Phi, \alpha \rangle$ is a rebuttal for $\langle \Psi, \beta \rangle$.

Proof As $\langle \Psi, \beta \rangle$ is a rebuttal for $\langle \Phi, \alpha \rangle$, $\alpha \vee \beta$ is a tautology and $\{\alpha, \beta\} \vdash \phi$ for each $\psi \in \Psi$. Apply proposition 3.3.1 to the assumption; then apply proposition 3.3.7; and, in view of what we just proved, apply proposition 3.3.9.* ∎

* It is also possible to prove the result directly, in the obvious way.

Proposition 3.3.8 Given two arguments $\langle \Phi, \alpha \rangle$ and $\langle \Psi, \beta \rangle$ such that $\neg(\alpha \wedge \beta)$ is a tautology, $\langle \Psi, \beta \rangle$ is a defeater for $\langle \Phi, \alpha \rangle$, and $\langle \Phi, \alpha \rangle$ is a defeater for $\langle \Psi, \beta \rangle$.

Proof As $\langle \Psi, \beta \rangle$ is an argument, $\Psi \vdash \beta$. By assumption, $\{\alpha, \beta\} \vdash \bot$. Therefore, $\Psi \cup \{\alpha\} \vdash \bot$. That is, $\alpha \vdash \neg(\psi_1 \wedge \cdots \wedge \psi_m)$ where $\{\psi_1, \ldots, \psi_m\} = \Psi$. The other case is clearly symmetric. ∎

Proposition 3.3.9 Let $\langle \Psi, \beta \rangle$ be a defeater for an argument $\langle \Phi, \alpha \rangle$. If $\alpha \vee \beta$ is a tautology and $\{\alpha, \beta\} \vdash \phi$ for each $\phi \in \Phi$, then $\langle \Psi, \beta \rangle$ is a rebuttal for $\langle \Phi, \alpha \rangle$.

Proof Since $\langle \Psi, \beta \rangle$ is a defeater of $\langle \Phi, \alpha \rangle$, we have $\beta \vdash \neg(\phi_1 \wedge \cdots \wedge \phi_n)$ for some $\{\phi_1, \ldots, \phi_n\} \subseteq \Phi$. By assumption, $\{\alpha, \beta\} \vdash \phi_i$ for $i = 1 \ldots n$. Therefore, $\{\alpha, \beta\} \vdash \bot$. Using classical logic, it follows that $\beta \leftrightarrow \neg\alpha$ is a tautology because $\alpha \vee \beta$ is a tautology (cf. the assumptions). ∎

Proposition 3.3.10 If $\langle \Phi, \alpha \rangle$ is an argument where Φ is logically equivalent with α, then each defeater $\langle \Psi, \beta \rangle$ of $\langle \Phi, \alpha \rangle$ such that $\alpha \vee \beta$ is a tautology is a rebuttal for $\langle \Phi, \alpha \rangle$.

Proof By definition of a defeater, $\beta \vdash \neg(\phi_1 \wedge \cdots \wedge \phi_n)$ for some $\{\phi_1, \ldots, \phi_n\} \subseteq \Phi$. By classical logic, $\Phi \vdash \neg\beta$. Since Φ is logically equivalent with α, it follows that $\alpha \vdash \neg\beta$. That is, $\neg\alpha \vee \neg\beta$ is a tautology. By assumption, $\alpha \vee \beta$ is also a tautology. Therefore, $\beta \leftrightarrow \neg\alpha$ is a tautology. ∎

Proposition 3.3.11 Let $\langle \Phi, \alpha \rangle$ and $\langle \Psi, \beta \rangle$ be two arguments. $\langle \Psi, \beta \rangle$ is both a rebuttal and an undercut for $\langle \Phi, \alpha \rangle$ iff Φ is logically equivalent with α and β is $\neg(\phi_1 \wedge \cdots \wedge \phi_n)$ such that $\Phi = \{\phi_1, \ldots, \phi_n\}$.

Proof ($\Rightarrow$) First, β is logically equivalent with $\neg\alpha$ because $\langle \Psi, \beta \rangle$ is a rebuttal for $\langle \Phi, \alpha \rangle$. Second, β is $\neg(\phi_{i_1} \wedge \cdots \wedge \phi_{i_k})$ for some $\{\phi_{i_1}, \ldots, \phi_{i_k}\} \subseteq \Phi$ because $\langle \Psi, \beta \rangle$ is an undercut for $\langle \Phi, \alpha \rangle$. Then, α is logically equivalent with $\{\phi_{i_1}, \ldots, \phi_{i_k}\}$. Accordingly, $\{\phi_{i_1}, \ldots, \phi_{i_k}\}$ entails α. Since $\langle \Phi, \alpha \rangle$ is an argument, there can be no proper subset of Φ entailing α. Hence, $\{\phi_{i_1}, \ldots, \phi_{i_k}\} = \Phi$. This has two consequences: Φ is logically equivalent with α, and β is $\neg(\phi_1 \wedge \cdots \wedge \phi_n)$.

($\Leftarrow$) Clearly, $\langle \Psi, \beta \rangle$ is an undercut for $\langle \Phi, \alpha \rangle$. Also, it trivially follows from β being $\neg(\phi_1 \wedge \cdots \wedge \phi_n)$ and Φ being logically equivalent with α that $\beta \leftrightarrow \neg\alpha$ is a tautology. That is, $\langle \Psi, \beta \rangle$ is a rebuttal for $\langle \Phi, \alpha \rangle$. ∎

Proposition 3.3.12 Let $\langle \Psi, \beta \rangle$ be a maximally conservative defeater for an argument $\langle \Phi, \alpha \rangle$. Then, $\langle \Psi, \gamma \rangle$ is a maximally conservative defeater for $\langle \Phi, \alpha \rangle$ iff γ is logically equivalent with β.

Proof We prove the nontrivial part. Let $\langle\Psi, \beta\rangle$ and $\langle\Psi, \gamma\rangle$ be maximally conservative defeaters for $\langle\Phi, \alpha\rangle$. Applying classical logic on top of the definition of a defeater, $\beta \vdash \neg(\phi_1 \wedge \cdots \wedge \phi_n)$ and $\gamma \vdash \neg(\phi_1 \wedge \cdots \wedge \phi_n)$ where $\Phi = \{\phi_1, \ldots, \phi_n\}$. Thus, $\beta \vee \gamma \vdash \neg(\phi_1 \wedge \cdots \wedge \phi_n)$. Now, there exists some minimal $\Psi' \subseteq \Psi$ such that $\Psi' \vdash \beta \vee \gamma$ (as $\Psi \vdash \beta$ and $\Psi \vdash \gamma$). Moreover, $\Psi' \nvdash \perp$ because $\Psi \nvdash \perp$ by definition of a defeater (which is required to be an argument). Hence, $\langle\Psi', \beta \vee \gamma\rangle$ is an argument, and it is a defeater for $\langle\Phi, \alpha\rangle$ as we have already proven $\beta \vee \gamma \vdash \neg(\phi_1 \wedge \cdots \wedge \phi_n)$. Due to $\Psi' \subseteq \Psi$, it follows that $\langle\Psi', \beta \vee \gamma\rangle$ is more conservative than $\langle\Psi, \beta\rangle$ and $\langle\Psi, \gamma\rangle$. Since each of these two is a maximally conservative defeater for $\langle\Phi, \alpha\rangle$, we obtain $\beta \vee \gamma \vdash \beta$ and $\beta \vee \gamma \vdash \gamma$. That is, β and γ are logically equivalent. ■

Proposition 3.3.13 Δ is inconsistent if there exists an argument that has at least one defeater. Should there be some inconsistent formula in Δ, the converse is untrue. When no formula in Δ is inconsistent, the converse is true in the following form: If Δ is inconsistent, then there exists an argument that has at least one rebuttal.

Proof Suppose $\langle\Psi, \beta\rangle$ is a defeater for $\langle\Phi, \alpha\rangle$. Hence, there exists $\{\phi_1, \ldots, \phi_n\} \subseteq \Phi$ such that $\Psi \vdash \neg(\phi_1 \wedge \cdots \wedge \phi_n)$. By classical logic, $\Psi \cup \{\phi_1, \ldots, \phi_n\} \vdash \perp$ and $\Psi \cup \Phi \vdash \perp$. Since $\Psi \cup \Phi \subseteq \Delta$, we have $\Delta \vdash \perp$. As for the converse, if each formula in Δ is consistent and Δ is inconsistent, then there exists a minimal inconsistent subset Φ. That is, $\Phi \vdash \perp$. By classical logic, $\Phi \backslash \{\phi\} \vdash \phi \rightarrow \perp$ for any $\phi \in \Phi$. That is, $\Phi \backslash \{\phi\} \vdash \neg\phi$. Clearly, $\{\phi\}$ and $\Phi \backslash \{\phi\}$ are consistent. Also, there exists a minimal subset Ψ of $\Phi \backslash \{\phi\}$ such that $\Psi \vdash \neg\phi$. Thus, $\langle\{\phi\}, \phi\rangle$ and $\langle\Psi, \neg\phi\rangle$ are arguments. Of course, $\langle\Psi, \neg\phi\rangle$ is a rebuttal for $\langle\{\phi\}, \phi\rangle$. ■

Corollary 3.3.5 Δ is inconsistent if there exists an argument that has at least one undercut. The converse is true when each formula in Δ is consistent. ■

Proposition 3.4.1 If $\langle\Psi, \neg(\phi_1 \wedge \cdots \wedge \phi_n)\rangle$ is a maximally conservative undercut for an argument $\langle\Phi, \alpha\rangle$, then $\Phi = \{\phi_1, \ldots, \phi_n\}$.

Proof Of course, $\{\phi_1, \ldots, \phi_n\} \subseteq \Phi$. Assume that there exists ϕ such that $\phi \in \Phi \backslash \{\phi_1, \ldots, \phi_n\}$. Since $\langle\Phi, \alpha\rangle$ is an argument, Φ is a minimal subset of Δ such that $\Phi \vdash \alpha$. Hence, $\{\phi_1, \ldots, \phi_n\} \nvdash \phi$ and $\neg(\phi \wedge \phi_1 \wedge \cdots \wedge \phi_n) \nvdash \neg(\phi_1 \wedge \cdots \wedge \phi_n)$.

Now, $\Psi \vdash \neg(\phi_1 \wedge \cdots \wedge \phi_n)$. Then, $\Psi \vdash \neg(\phi \wedge \phi_1 \wedge \cdots \wedge \phi_n)$. Thus, there exists $\Psi' \subseteq \Psi$ such that $\langle\Psi', \neg(\phi \wedge \phi_1 \wedge \cdots \wedge \phi_n)\rangle$ is an argument. Since $\neg(\phi_1 \wedge \cdots \wedge \phi_n) \vdash \neg(\phi \wedge \phi_1 \wedge \cdots \wedge \phi_n)$, it follows that $\langle \Psi', \neg(\phi \wedge \phi_1 \wedge \cdots \wedge \phi_n)\rangle$ is more conservative than $\langle \Psi, \neg(\phi_1 \wedge \cdots \wedge \phi_n)\rangle$. In fact, $\langle \Psi', \neg(\phi \wedge \phi_1 \wedge \cdots \wedge \phi_n)\rangle$ is strictly more conservative than $\langle \Psi, \neg(\phi_1 \wedge \cdots \wedge \phi_n)\rangle$ because $\neg(\phi \wedge \phi_1 \wedge \cdots \wedge \phi_n) \nvdash \neg(\phi_1 \wedge \cdots \wedge \phi_n)$. Moreover, $\langle \Psi', \neg(\phi \wedge \phi_1 \wedge \cdots \wedge \phi_n)\rangle$ is clearly an undercut for $\langle \Phi, \alpha\rangle$ so that $\langle \Psi, \neg(\phi_1 \wedge \cdots \wedge \phi_n)\rangle$ being a maximally conservative undercut for $\langle \Phi, \alpha\rangle$ is contradicted. ■

Proposition 3.4.2 An argument $\langle \Psi, \neg(\phi_1 \wedge \cdots \wedge \phi_n)\rangle$ is a canonical undercut for $\langle \Phi, \alpha\rangle$ iff it is an undercut for $\langle \Phi, \alpha\rangle$ and $\langle \phi_1, \ldots, \phi_n\rangle$ is the canonical enumeration of Φ.

Proof We prove the nontrivial part. Let $\langle \Psi, \neg(\phi_1 \wedge \cdots \wedge \phi_n)\rangle$ be an undercut for $\langle \Phi, \alpha\rangle$ such that $\langle \phi_1, \ldots, \phi_n\rangle$ is the canonical enumeration of Φ. We only need to show that $\langle \Psi, \neg(\phi_1 \wedge \cdots \wedge \phi_n)\rangle$ is a maximally conservative undercut for $\langle \Phi, \alpha\rangle$. Assume that $\langle \Theta, \neg(\gamma_1 \wedge \cdots \wedge \gamma_m)\rangle$ is an undercut for $\langle \Phi, \alpha\rangle$ that is more conservative than $\langle \Psi, \neg(\phi_1 \wedge \cdots \wedge \phi_n)\rangle$. By definition 3.3.2, $\{\gamma_1, \ldots, \gamma_m\} \subseteq \Phi$. Now, $\Phi = \{\phi_1, \ldots, \phi_n\}$. It follows that $\neg(\gamma_1 \wedge \cdots \wedge \gamma_m) \vdash \neg(\phi_1 \wedge \cdots \wedge \phi_n)$. Hence, $\Theta \vdash \neg(\phi_1 \wedge \cdots \wedge \phi_n)$ because $\langle \Theta, \neg(\gamma_1 \wedge \cdots \wedge \gamma_m)\rangle$ is an argument. However, $\Theta \subseteq \Psi$ due to the assumption that $\langle \Theta, \neg(\gamma_1 \wedge \cdots \wedge \gamma_m)\rangle$ is more conservative than $\langle \Psi, \neg(\phi_1 \wedge \cdots \wedge \phi_n)\rangle$. Should Θ be a proper subset of Ψ, it would then be the case that Ψ is not a minimal subset of Δ entailing $\neg(\phi_1 \wedge \cdots \wedge \phi_n)$ and this would contradict the fact that $\langle \Psi, \neg(\phi_1 \wedge \cdots \wedge \phi_n)\rangle$ is an argument. Thus, $\Theta = \Psi$, and it follows that $\langle \Psi, \neg(\phi_1 \wedge \cdots \wedge \phi_n)\rangle$ is more conservative than $\langle \Theta, \neg(\gamma_1 \wedge \cdots \wedge \gamma_m)\rangle$, as we already proved $\neg(\gamma_1 \wedge \cdots \wedge \gamma_m) \vdash \neg(\phi_1 \wedge \cdots \wedge \phi_n)$. ■

Corollary 3.4.1 A pair $\langle \Psi, \neg(\phi_1 \wedge \cdots \wedge \phi_n)\rangle$ is a canonical undercut for $\langle \Phi, \alpha\rangle$ iff it is an argument and $\langle \phi_1, \ldots, \phi_n\rangle$ is the canonical enumeration of Φ. ■

Proposition 3.4.3 Any two different canonical undercuts for the same argument have the same consequent but distinct supports. ■

Proposition 3.4.4 Given two different canonical undercuts for the same argument, neither is more conservative than the other.

Proof In view of proposition 3.4.3, it is enough to consider the case of $\langle \Psi_1, \neg\phi\rangle$ and $\langle \Psi_2, \neg\phi\rangle$ being two different canonical undercuts for the same argument $\langle \Phi, \alpha\rangle$. Assume that $\langle \Psi_1, \neg\phi\rangle$ is more conservative than $\langle \Psi_2, \neg\phi\rangle$. Then, $\langle \Psi_2, \neg\phi\rangle$ is more conservative than $\langle \Psi_1, \neg\phi\rangle$ because

$\langle \Psi_1, \neg\phi \rangle$ is an undercut for $\langle \Phi, \alpha \rangle$ and $\langle \Psi_2, \neg\phi \rangle$ is a maximally conservative undercut for $\langle \Phi, \alpha \rangle$. Overall, $\Psi_1 = \Psi_2$. Thus, $\langle \Psi_1, \neg\phi \rangle = \langle \Psi_2, \neg\phi \rangle$, and this is a contradiction. ■

Proposition 3.4.5 For each defeater $\langle \Psi, \beta \rangle$ of an argument $\langle \Phi, \alpha \rangle$, there exists a canonical undercut for $\langle \Phi, \alpha \rangle$ that is more conservative than $\langle \Psi, \beta \rangle$.

Proof Consider an argument $\langle \Phi, \alpha \rangle$, and write $\langle \phi_1, \dots, \phi_n \rangle$ for the canonical enumeration of Φ. Applying proposition 3.3.2, it is enough to prove the result for any undercut $\langle \Psi, \neg(\gamma_1 \wedge \dots \wedge \gamma_m) \rangle$ of $\langle \Phi, \alpha \rangle$. By definition 3.3.2, $\{\gamma_1, \dots, \gamma_m\} \subseteq \{\phi_1, \dots, \phi_n\}$. Thus, $\neg(\gamma_1 \wedge \dots \wedge \gamma_m) \vdash \neg(\phi_1 \wedge \dots \wedge \phi_n)$. Then, $\Psi \vdash \neg(\phi_1 \wedge \dots \wedge \phi_n)$ because $\langle \Psi, \neg(\gamma_1 \wedge \dots \wedge \gamma_m) \rangle$ is an undercut, hence an argument. Accordingly, there exists a minimal subset $\Psi' \subseteq \Psi \subseteq \Delta$ such that $\Psi' \vdash \neg(\phi_1 \wedge \dots \wedge \phi_n)$. Moreover, Ψ' is clearly consistent. Therefore, $\langle \Psi', \neg(\phi_1 \wedge \dots \wedge \phi_n) \rangle$ is an argument. Obviously, it is an undercut for $\langle \Phi, \alpha \rangle$. It is a canonical undercut for $\langle \Phi, \alpha \rangle$ in view of proposition 3.4.2. It is more conservative than $\langle \Psi, \neg(\gamma_1 \wedge \dots \wedge \gamma_m) \rangle$ as we have $\Psi' \subseteq \Psi$ and $\neg(\gamma_1 \wedge \dots \wedge \gamma_m) \vdash \neg(\phi_1 \wedge \dots \wedge \phi_n)$. ■

Proposition 3.4.6 If $\langle \Psi, \beta \rangle$ is a maximally conservative undercut for $\langle \Phi, \alpha \rangle$ then $\langle \Psi, \beta \rangle$ also is a maximally conservative defeater for $\langle \Phi, \alpha \rangle$.

Proof Consider $\langle \Psi', \beta' \rangle$ a defeater of $\langle \Phi, \alpha \rangle$ such that $\Psi' \subseteq \Psi$ and $\beta \vdash \beta'$. Hence, $\beta' \vdash \neg(\phi_1 \wedge \dots \wedge \phi_n)$ for some $\{\phi_1, \dots, \phi_n\} \subseteq \Phi$. By classical logic, $\beta' \vdash \neg(\phi_1 \wedge \dots \wedge \phi_m)$ where $\Phi = \{\phi_1, \dots, \phi_m\}$. However, proposition 3.4.1 states that β is logically equivalent with $\neg(\phi_1 \wedge \dots \wedge \phi_n)$. Therefore, $\beta' \vdash \beta$. By proposition 3.2.2, it follows that $\langle \Psi', \beta \rangle$ is an argument. It is an undercut of $\langle \Phi, \alpha \rangle$ because $\Phi = \{\phi_1, \dots, \phi_m\}$. Since $\langle \Psi, \beta \rangle$ is a maximally conservative undercut of $\langle \Phi, \alpha \rangle$, it happens that $\Psi' \subseteq \Psi$ entails $\Psi \subseteq \Psi'$. The proof is over because $\beta' \vdash \beta$ and $\Psi \subseteq \Psi'$ have been shown, establishing that $\langle \Psi, \beta \rangle$ is a maximally conservative defeater of $\langle \Phi, \alpha \rangle$. ■

Proposition 3.4.7 If $\langle \Psi, \neg(\phi_1 \wedge \dots \wedge \phi_n) \rangle$ is an undercut for $\langle \Phi, \alpha \rangle$ such that $\Phi = \{\phi_1, \dots, \phi_n\}$, then it is a maximally conservative undercut for $\langle \Phi, \alpha \rangle$.

Proof Let $\langle \Psi, \neg(\phi_1 \wedge \dots \wedge \phi_n) \rangle$ be an undercut for $\langle \Phi, \alpha \rangle$ such that $\Phi = \{\phi_1, \dots, \phi_n\}$. Assume that $\langle \Theta, \neg(\gamma_1 \wedge \dots \wedge \gamma_m) \rangle$ is an undercut for $\langle \Phi, \alpha \rangle$ that is more conservative than $\langle \Psi, \neg(\phi_1 \wedge \dots \wedge \phi_n) \rangle$. By definition 3.3.2, $\{\gamma_1, \dots, \gamma_m\} \subseteq \Phi$. Now, $\Phi = \{\phi_1, \dots, \phi_n\}$. It follows that $\neg(\gamma_1 \wedge \cdots \wedge \gamma_m) \vdash \neg(\phi_1 \wedge \cdots \wedge \phi_n)$. Hence, $\Theta \vdash \neg(\phi_1 \wedge \cdots \wedge \phi_n)$ because $\langle \Theta, \neg(\gamma_1 \wedge \cdots \wedge \gamma_m) \rangle$ is an argument. However, $\Theta \subseteq \Psi$ due to the assumption that $\langle \Theta, \neg(\gamma_1 \wedge \cdots \wedge \gamma_m) \rangle$ is more conservative than $\langle \Psi, \neg(\phi_1 \wedge \cdots \wedge \phi_n) \rangle$. Should Θ be a proper subset of Ψ, it would then be the case that Ψ is not a minimal subset of Δ entailing $\neg(\phi_1 \wedge \cdots \wedge \phi_n)$ and this would contradict the fact that $\langle \Psi, \neg(\phi_1 \wedge \cdots \wedge \phi_n) \rangle$ is an argument. Thus, $\Theta = \Psi$, and it follows that $\langle \Psi, \neg(\phi_1 \wedge \cdots \wedge \phi_n) \rangle$ is more conservative than $\langle \Theta, \neg(\gamma_1 \wedge \cdots \wedge \gamma_m) \rangle$, as we already proved $\neg(\gamma_1 \wedge \cdots \wedge \gamma_m) \vdash \neg(\phi_1 \wedge \cdots \wedge \phi_n)$. ∎

Corollary 3.4.2 Let $\langle \Psi, \neg(\phi_1 \wedge \cdots \wedge \phi_n) \rangle$ be an undercut for $\langle \Phi, \alpha \rangle$. $\langle \Psi, \neg(\phi_1 \wedge \cdots \wedge \phi_n) \rangle$ is a maximally conservative undercut for $\langle \Phi, \alpha \rangle$ iff $\Phi = \{\phi_1, \dots, \phi_n\}$.

Proof Combine proposition 3.4.1 and proposition 3.4.7. ∎

Proposition 3.4.8 If $\langle \Psi, \beta \rangle$ is both a rebuttal and an undercut for $\langle \Phi, \alpha \rangle$ then $\langle \Psi, \beta \rangle$ is a maximally conservative undercut for $\langle \Phi, \alpha \rangle$.

Proof By proposition 3.3.11, β is $\neg(\phi_1 \wedge \cdots \wedge \phi_n)$ where $\Phi = \{\phi_1, \dots, \phi_n\}$. Applying proposition 3.4.7, $\langle \Psi, \beta \rangle$ is a maximally conservative undercut for $\langle \Phi, \alpha \rangle$. ∎

Proposition 3.4.9 Let $\langle \Phi, \alpha \rangle$ be an argument. Each maximally conservative undercut for $\langle \Phi, \alpha \rangle$ is a rebuttal for $\langle \Phi, \alpha \rangle$ iff Φ is logically equivalent with α.

Proof ($\Rightarrow$) Let $\langle \Psi, \beta \rangle$ be a maximally conservative undercut for $\langle \Phi, \alpha \rangle$. By the assumption, it is also a rebuttal for $\langle \Phi, \alpha \rangle$. Hence, $\beta \leftrightarrow \neg\alpha$ is a tautology. As $\langle \Psi, \beta \rangle$ is a maximally conservative undercut for $\langle \Phi, \alpha \rangle$, proposition 3.4.1 yields that β is $\neg(\phi_1 \wedge \cdots \wedge \phi_n)$ where $\Phi = \{\phi_1, \dots, \phi_n\}$. Thus, $\neg(\phi_1 \wedge \cdots \wedge \phi_n) \leftrightarrow \neg\alpha$ is a tautology. That is, Φ is logically equivalent with α. ($\Leftarrow$) Let $\langle \Psi, \beta \rangle$ be a maximally conservative undercut for $\langle \Phi, \alpha \rangle$. By proposition 3.4.1, β is $\neg(\phi_1 \wedge \cdots \wedge \phi_n)$ where $\Phi = \{\phi_1, \dots, \phi_n\}$. Since Φ is logically equivalent with α, it follows that $\beta \leftrightarrow \neg\alpha$ is a tautology. That is, $\langle \Psi, \beta \rangle$ is a rebuttal for $\langle \Phi, \alpha \rangle$. ∎

Proposition 3.4.10 If $\langle \Psi, \beta \rangle$ is a maximally conservative undercut for $\langle \Phi, \alpha \rangle$, then there exists $\Phi' \subseteq \Phi$ and γ such that $\langle \Phi', \gamma \rangle$ is a maximally conservative undercut for $\langle \Psi, \beta \rangle$.

Proof Since $\langle \Psi, \beta \rangle$ is a maximally conservative undercut for $\langle \Phi, \alpha \rangle$, proposition 3.4.1 implies that $\neg\beta$ is logically equivalent with Φ. Since

$\Psi \vdash \beta$, it follows that $\Phi \vdash \neg(\psi_1 \wedge \cdots \wedge \psi_n)$ where $\Psi = \{\psi_1, \ldots, \psi_n\}$. By classical logic, there exists a minimal subset Φ' of Φ such that $\Phi' \vdash \neg(\psi_1 \wedge \cdots \wedge \psi_n)$. Also, $\Phi' \nvdash \perp$ because $\Phi \nvdash \perp$ (as $\langle \Phi, \alpha \rangle$ is an argument). Hence, $\langle \Phi', \gamma \rangle$ is an undercut of $\langle \Psi, \beta \rangle$, taking γ to be $\neg(\psi_1 \wedge \cdots \wedge \psi_n)$. Lastly, apply proposition 3.4.7. ■

Corollary 3.4.3 If $\langle \Psi, \beta \rangle$ is a canonical undercut for $\langle \Phi, \alpha \rangle$, then there exists $\Phi' \subseteq \Phi$ such that $\langle \Phi, \neg(\psi_1 \wedge \cdots \wedge \Psi_n) \rangle$ is a canonical undercut for $\langle \Psi, \beta \rangle$ (where $\langle \psi_1, \ldots, \psi_n \rangle$ is the canonical enumeration of Ψ). ■

Proposition 3.4.11 If $\langle \Phi, \alpha \rangle$ is an argument such that Φ is logically equivalent with α, then each rebuttal of $\langle \Phi, \alpha \rangle$ is equivalent with a canonical undercut for $\langle \Phi, \alpha \rangle$.

Proof Let $\langle \Psi, \beta \rangle$ be a rebuttal of $\langle \Phi, \alpha \rangle$. Then, $\beta \leftrightarrow \neg\alpha$ is a tautology. Taking $\langle \phi_1, \ldots, \phi_n \rangle$ to be the canonical enumeration of Φ, it follows that $\beta \leftrightarrow \neg(\phi_1 \wedge \cdots \wedge \phi_n)$ is a tautology because Φ is logically equivalent with α. Hence, $\langle \Psi, \beta \rangle$ is equivalent with $\langle \Psi, \neg(\phi_1 \wedge \cdots \wedge \phi_n) \rangle$, which is clearly an undercut for $\langle \Phi, \alpha \rangle$. Due to proposition 3.4.2, $\langle \Psi, \neg(\phi_1 \wedge \cdots \wedge \phi_n) \rangle$ is a canonical undercut for $\langle \Phi, \alpha \rangle$. ■

Proposition 3.5.1 Argument trees are finite.

Proof Since Δ is finite, there are only a finite number of subsets of Δ. By condition 2 of definition 3.5.1, no branch in an argument tree can then be infinite. Also, there is then a finite number of canonical undercuts (definition 3.4.2). By condition 3 of definition 3.5.1, the branching factor in an argument tree is finite. ■

Proposition 3.5.2 If Δ is consistent, then all argument trees have exactly one node. The converse is true when no formula in Δ is inconsistent.

Proof Apply corollary 3.3.5. ■

Proposition 3.5.3 Let T be an argument tree whose root node $\langle \Phi, \alpha \rangle$ is such that no subset of Δ is logically equivalent with α. Then, no node in T is a rebuttal for the root.

Proof Assume that T contains a node $\langle \Psi, \beta \rangle$ that is a rebuttal for the root $\langle \Phi, \alpha \rangle$. Hence, β is logically equivalent with $\neg\alpha$. However, $\langle \Psi, \beta \rangle$ is a canonical undercut of some node $\langle \Theta, \gamma \rangle$ in T. By the definition of a canonical undercut, β is $\neg(\theta_1 \wedge \cdots \wedge \theta_n)$ where $\langle \theta_1, \ldots, \theta_n \rangle$ is the canonical enumeration of Θ. That is, Θ is logically equivalent with $\neg\beta$. Accordingly, Θ is logically equivalent with α and a contradiction arises. ■

Proposition 3.6.1 For every maximally conservative undercut $\langle \Psi, \beta \rangle$ to an argument $\langle \Phi, \alpha \rangle$, there exist at least $2^m - 1$ arguments each of which undercuts the undercut (m is the size of Ψ). Each of these m arguments is a duplicate to the undercut.

Proof Let β be $\neg(\phi_1 \wedge \cdots \wedge \phi_n)$ where $\{\phi_1, \ldots, \phi_n\} \subseteq \Phi \subseteq \Delta$. Let $\Psi = \{\psi_1, \ldots, \psi_m\}$. According to the assumption, $\Psi \nvdash \perp$ and Ψ is a minimal subset of Δ such that

$$\Psi \vdash \neg(\phi_1 \wedge \cdots \wedge \phi_n).$$

We show the result for each nonempty subset of Ψ, which gives us $2^m - 1$ cases, considering only $\{\psi_1, \ldots, \psi_p\} \subseteq \{\psi_1, \ldots, \psi_m\}$ in the proof, as the case is clearly the same for all sets $\{\psi_1, \ldots, \psi_p\}$ that can be selected from all possible permutations of $\langle \psi_1, \ldots, \psi_m \rangle$: We show that $\langle \{\phi_1, \ldots, \phi_n, \psi_{p+1}, \ldots, \psi_m\}, \neg(\psi_1 \wedge \cdots \wedge \psi_p) \rangle$ is an argument.

From the hypothesis $\{\psi_1, \ldots, \psi_m\} \vdash \neg(\phi_1 \wedge \cdots \wedge \phi_n)$, we get $\{\phi_1, \ldots, \phi_n, \psi_{p+1}, \ldots, \psi_m\} \vdash \neg(\psi_1 \wedge \cdots \wedge \psi_p)$ by classical logic. If we assume $\{\phi_1, \ldots, \phi_n, \psi_{p+1}, \ldots, \psi_m\} \vdash \perp$, we would get $\{\psi_{p+1}, \ldots, \psi_m\} \vdash \neg(\phi_1 \wedge \cdots \wedge \phi_n)$. Therefore, $\langle \{\psi_1, \ldots, \psi_m\}, \neg(\phi_1 \wedge \cdots \wedge \phi_n) \rangle$ would not be an argument because $\{\psi_1, \ldots, \psi_m\}$ would not be a minimal subset of Δ entailing $\neg(\phi_1 \wedge \cdots \wedge \phi_n)$. This would contradict the fact that $\langle \Psi, \neg(\phi_1 \wedge \cdots \wedge \phi_n) \rangle$ is an undercut, hence an argument. That is, we have proven that $\{\phi_1, \ldots, \phi_n, \psi_{p+1}, \ldots, \psi_m\} \nvdash \perp$. There only remains to show that $\{\phi_1, \ldots, \phi_n, \psi_{p+1}, \ldots, \psi_m\}$ is a minimal subset of Δ for $\neg(\psi_1 \wedge \cdots \wedge \psi_p)$ to be entailed. We use reductio ad absurdum, assuming that at least one ϕ_i or ψ_j is unnecessary when deducing $\neg(\psi_1 \wedge \cdots \wedge \psi_p)$. In symbols, either $\{\phi_2, \ldots, \phi_n, \psi_{p+1}, \ldots, \psi_m\} \vdash \neg(\psi_1 \wedge \cdots \wedge \psi_p)$ or $\{\phi_1, \ldots, \phi_n, \psi_{p+2}, \ldots, \psi_m\} \vdash \neg(\psi_1 \wedge \cdots \wedge \psi_p)$ (again, all cases are symmetrical, so it is enough to consider only $i = 1$ and $j = p + 1$).

Let us first assume $\{\phi_1, \ldots, \phi_n, \psi_{p+2}, \ldots, \psi_m\} \vdash \neg(\psi_1 \wedge \cdots \wedge \psi_p)$. Thus, $\{\psi_1, \ldots, \psi_p, \psi_{p+2}, \ldots, \psi_m\} \vdash \neg(\phi_1 \wedge \cdots \wedge \phi_n)$. This would contradict $\{\psi_1, \ldots, \psi_m\}$ being a minimal subset of Δ entailing $\neg(\phi_1 \wedge \cdots \wedge \phi_n)$.

Turning to the second case, let us assume that $\{\phi_2, \ldots, \phi_n, \psi_{p+1}, \ldots, \psi_m\} \vdash \neg(\psi_1 \wedge \cdots \wedge \psi_p)$. Thus, $\{\psi_1, \ldots, \psi_m\} \vdash \neg(\phi_2 \wedge \cdots \wedge \phi_n)$. We also know that $\{\psi_1, \ldots, \psi_m\} \nvdash \perp$, so, $\langle \Psi', \neg(\phi_2 \wedge \cdots \wedge \phi_n) \rangle$ would be an argument for some $\Psi' \subseteq \Psi$. However, it then would be an undercut for $\langle \Phi, \alpha \rangle$, and it would be more conservative than $\langle \Psi, \neg(\phi_1 \wedge \cdots \wedge \phi_n) \rangle$. As a consequence, $\langle \Psi, \neg(\phi_1 \wedge \cdots \wedge \phi_n) \rangle$ would not be a maximally conservative undercut for $\langle \Phi, \alpha \rangle$, and this again is a contradiction. Either case yields a contradiction; hence, no proper subset of $\{\phi_1, \ldots, \phi_n,$

$\psi_{p+1}, \ldots, \psi_m\}$ entails $\neg(\psi_1 \wedge \cdots \wedge \psi_p)$. Thus, $\langle\{\phi_1, \ldots, \phi_n, \psi_{p+1}, \ldots, \psi_m\}, \neg(\psi_1 \wedge \cdots \wedge \psi_p)\rangle$ is an argument, and it clearly is an undercut to the undercut. Verification that it is a duplicate of the undercut is routine. ∎

Proposition 3.6.2 No two maximally conservative undercuts of the same argument are duplicates.

Proof Let $\langle\Gamma \cup \Theta, \neg(\alpha_1 \wedge \cdots \wedge \alpha_n)\rangle$ and $\langle\Gamma \cup \Theta', \neg(\alpha'_1 \wedge \cdots \wedge \alpha'_m)\rangle$ be two maximally conservative undercuts for $\langle\Phi, \beta\rangle$ that are duplicates of each other. Then, Θ is logically equivalent with $\alpha'_1 \wedge \cdots \wedge \alpha'_m$. Hence, $\Gamma \cup \Theta \vdash \alpha'_1 \wedge \cdots \wedge \alpha'_m$ while $\Gamma \cup \Theta \vdash \neg(\alpha_1 \wedge \cdots \wedge \alpha_n)$. According to proposition 3.4.1, $\Phi = \{\alpha_1, \ldots, \alpha_n\} = \{\alpha'_1, \ldots, \alpha'_m\}$. That is, $\alpha_1 \wedge \cdots \wedge \alpha_n$ and $\alpha'_1 \wedge \cdots \wedge \alpha'_m$ are logically equivalent. Therefore, $\Gamma \cup \Theta \vdash \perp$. This would contradict the fact that $\Gamma \cup \Theta$ is the support of an argument. ∎

Corollary 3.6.1 No two canonical undercuts of the same argument are duplicates. ∎

Proposition 3.6.3 No branch in an argument tree contains duplicates, except possibly for the child of the root to be a duplicate to the root.

Proof Assume the contrary: There is $\langle\Gamma \cup \Phi, \neg\psi\rangle$, which is an ancestor node of $\langle\Gamma \cup \Psi, \neg\phi\rangle$ where Φ is $\{\phi_1, \ldots, \phi_n\}$, Ψ is $\{\psi_1, \ldots, \psi_m\}$, ϕ is $\phi_1 \wedge \cdots \wedge \phi_n$ and ψ is $\psi_1 \wedge \cdots \wedge \psi_m$. By definition of an argument tree, $\langle\Gamma \cup \Psi, \neg\phi\rangle$ is a canonical undercut for its parent node, which is $\langle\Phi, \alpha\rangle$ for some α and which also has $\langle\Gamma \cup \Phi, \neg\psi\rangle$ as an ancestor node. Unless $\langle\Phi, \alpha\rangle$ is the same node as $\langle\Gamma \cup \Phi, \neg\psi\rangle$, this contradicts condition 2 of definition 3.5.1 because Φ is, of course, a subset of $\Gamma \cup \Phi$. In case $\langle\Phi, \alpha\rangle$ coincides with $\langle\Gamma \cup \Phi, \neg\psi\rangle$, then $\Gamma \subseteq \Phi$. Also, $\langle\Phi, \neg\psi\rangle$ is the parent node for $\langle\Gamma \cup \Psi, \neg\phi\rangle$. Should the former not be the root, it has a parent node of the form $\langle\Psi, \beta\rangle$ for some β. Therefore, $\langle\Psi, \beta\rangle$ is an ancestor node of $\langle\Gamma \cup \Psi, \neg\phi\rangle$. Moreover, $\langle\Phi, \neg\psi\rangle$ has already been proven the parent node for $\langle\Gamma \cup \Psi, \neg\phi\rangle$ where $\Gamma \subseteq \Phi$. Thus, condition 2 of definition 3.5.1 is again contradicted.[†] ∎

Proposition 3.7.1 Let $\langle\mathscr{P}, \mathscr{C}\rangle$ be an argument structure. If there exists an argument tree in $\mathscr{P}$ that has exactly one node, then $\mathscr{C}$ is the empty set. The converse is untrue, even when assuming that $\mathscr{P}$ is nonempty.

[†] Strictly speaking, it is assumed thoughout the proof that rewriting (a canonical enumeration of) a subset of Δ into a conjunctive formula is an injective function. Such a restriction is inessential because it only takes an easy procedure to have the assumption guaranteed to hold.

Proof Assume the contrary: $\mathcal{P}$ contains an argument tree consisting of a single argument while $\mathcal{C}$ is nonempty. Let the single node tree in $\mathcal{P}$ be the argument $\langle \Phi, \alpha \rangle$, and let the root node of the tree in $\mathcal{C}$ be the argument $\langle \Psi, \neg\alpha \rangle$. By definition of an argument, $\Phi \vdash \alpha$ and $\Psi \vdash \neg\alpha$. Therefore, $\Psi \vdash \neg\phi$ where ϕ is $\phi_1 \wedge \cdots \wedge \phi_n$ such that $\langle \phi_1, \ldots, \phi_n \rangle$ is the canonical enumeration of Φ. Accordingly, $\langle \Psi', \neg\phi \rangle$ is an undercut of $\langle \Phi, \alpha \rangle$ for some $\Psi' \subseteq \Psi$. That is, there exists a canonical undercut for $\langle \Phi, \alpha \rangle$ (cf. proposition 3.4.5) and this contradicts the fact that there is an argument tree consisting only of $\langle \Phi, \alpha \rangle$. ∎

Proposition 3.7.2 Let $\langle \mathcal{P}, \mathcal{C} \rangle$ be an argument structure. If Δ is consistent, then each argument tree in $\mathcal{P}$ has exactly one node and $\mathcal{C}$ is the empty set. The converse is untrue, even when assuming that $\mathcal{P}$ is nonempty and that each formula in Δ is consistent.

Proof For each argument that is a root node in an argument tree of $\mathcal{P}$, apply corollary 3.3.5. Thus, we have just proved that each member of $\mathcal{P}$ consists of a single argument. As for $\mathcal{C}$ being the empty set, it is then enough to apply proposition 3.7.1. ∎

Proposition 3.7.3 Let $\langle [X_1, \ldots, X_n], [Y_1, \ldots, Y_m] \rangle$ be an argument structure. For any i and any j, the support of the root node of Y_j (resp. X_i) is a superset of the support of a canonical undercut for the root node of X_i (resp. Y_j).

Proof Let $\langle \Phi, \alpha \rangle$ be the root node of X_i and $\langle \Psi, \neg\alpha \rangle$ be the root node of Y_j. Then, $\Phi \vdash \alpha$ and $\Psi \vdash \neg\alpha$. Thus, $\Phi \cup \Psi \vdash \perp$. Accordingly, $\Psi \vdash \neg\phi$ where ϕ is $\phi_1 \wedge \cdots \wedge \phi_k$ and $\langle \phi_1, \ldots, \phi_k \rangle$ is the canonical enumeration of Φ. Also, Ψ is consistent because $\langle \Psi, \neg\alpha \rangle$ is an argument. Let Ψ' be a minimal subset of Ψ such that $\Psi' \vdash \neg\phi$. Then, $\langle \Psi', \neg\phi \rangle$ is an argument. It clearly is a canonical undercut for $\langle \Phi, \alpha \rangle$. ∎

Proposition 3.7.4 Let $\langle \mathcal{P}, \mathcal{C} \rangle$ be an argument structure. Then, both $\mathcal{P}$ and $\mathcal{C}$ are finite.

Proof We only prove the result for $\mathcal{P}$. Clearly, no two argument trees have the same root node. Therefore, all argument trees of $\mathcal{P}$ have a different support. Thus, there can only be as many argument trees of $\mathcal{P}$ as there are subsets of Δ. These are finitely many because Δ is finite. ∎

Proposition 7.2.1 If $\langle \Psi, \diamond \rangle$ is a canonical undercut, then there is an $X \in \mathsf{MinIncon}(\Delta)$ such that $\Psi \subset X$.

Proof Let $\langle\Phi, \alpha\rangle$ be the argument for which $\langle\Psi, \diamond\rangle$ is a canonical undercut. Therefore, $\Phi \cup \Psi \vdash \perp$. Therefore, there is a subset $\Phi' \subseteq \Phi$ where $\Phi' \cup \Psi \vdash \perp$ and for all $\Phi'' \subset \Phi'$, $\Phi'' \cup \Phi' \nvdash \perp$. Therefore, $\Phi' \cup \Psi \in \mathsf{MinIncon}(\Delta)$ and $\Psi \subset (\Phi' \cup \Psi)$. ∎

Proposition 7.2.2 Let $\langle\Phi, \alpha\rangle$ be an argument. For all $X \in \mathsf{MinIncon}(\Delta)$, if Φ is a nonempty subset of X, then $\langle X \backslash \Phi, \diamond\rangle$ is a canonical undercut for $\langle\Phi, \alpha\rangle$.

Proof Since $X \in \mathsf{MinIncon}(\Delta)$, and $\Phi \cap X \neq \emptyset$, we have $\Phi \cup (X \backslash \Phi) \vdash \perp$. Furthermore, there is no $W \subset (X \backslash \Phi)$ such that $\Phi \cup W \vdash \perp$. Therefore, $\langle X \backslash \Phi, \diamond\rangle$ is a canonical undercut of $\langle\Phi, \alpha\rangle$. ∎

Proposition 7.2.4 If $\langle\Psi, \diamond\rangle$ is a canonical undercut for $\langle\Phi, \alpha\rangle$, then there is an X such that $X \in \mathsf{MinIncon}(\Delta)$ and $\Psi = X \backslash \Phi$.

Proof Assume $\langle\Psi, \diamond\rangle$ is canonical undercut for $\langle\Phi, \alpha\rangle$. Therefore, $\Psi \vdash \neg(\wedge\Phi)$ and there is no $\Psi' \subset \Psi$ such that $\Psi' \vdash \neg(\wedge\Phi)$. Therefore, $\Psi \cup \Phi \vdash \perp$, and there is no $\Psi' \subset \Psi$ such that $\Psi' \cup \Phi \vdash \perp$. Therefore, there is a $\Phi' \subseteq \Phi$ such that $\Psi \cup \Phi' \in \mathsf{MinIncon}(\Delta)$. Let X be $\Psi \cup \Phi'$. Furthermore, $(X \backslash \Phi) \cup \Phi \vdash \perp$, and there is no $Z \subset (X \backslash \Phi)$ such that $Z \cup \Phi \vdash \perp$. Hence, $\Psi = X \backslash \Phi$. ∎

Proposition 7.2.5 Let $\langle\Phi, \alpha\rangle$ be an argument. For all $X, Y \in \mathsf{MinIncon}(\Delta)$ if $X \cap Y \neq \emptyset$, and $\Phi \subseteq X$, and $Y \cap \Phi \neq \emptyset$, and there is no $Z \in \mathsf{MinIncon}(\Delta)$ such that $(Z \backslash \Phi) \subset (Y \backslash \Phi)$, then $\langle Y \backslash \Phi, \diamond\rangle$ is a canonical undercut for $\langle\Phi, \alpha\rangle$.

Proof Let $\langle\Phi, \alpha\rangle$ be an argument, and let $X, Y \in \mathsf{MinIncon}(\Delta)$ be such that $X \cap Y \neq \emptyset$, and $\Phi \subseteq X$, and $Y \cap \Phi \neq \emptyset$. Therefore $(Y \backslash \Phi) \cup \Phi \vdash \perp$. Also assume there is no $Z \in \mathsf{MinIncon}(\Delta)$ such that $(Z \backslash \Phi) \subset (Y \backslash \Phi)$. Therefore there is no $W \subset (Y \backslash \Phi)$ such that $W \cup \Phi \vdash \perp$. Therefore $\langle Y \backslash \Phi, \diamond\rangle$ is a canonical undercut for $\langle\Phi, \alpha\rangle$. ∎

Proposition 7.2.6 Let $\langle\Phi, \diamond\rangle$ be a canonical undercut. If $\langle\Psi, \diamond\rangle$ is a canonical undercut of $\langle\Phi, \diamond\rangle$, then $\Psi \subset Y$ for some $Y \in \mathsf{MinIncon}(\Delta)$ such that $X \cap Y \neq \emptyset$ for every $X \in \mathsf{MinIncon}(\Delta)$ satisfying $\Phi \subset X$.

Proof Suppose $\langle\Psi, \diamond\rangle$ is a canonical undercut of $\langle\Phi, \diamond\rangle$. Therefore, $\Psi \cup \Phi \vdash \perp$, and there is not a $\Psi' \subset \Psi$ such that $\Psi' \cup \Phi \vdash \perp$. Therefore, there is a $Y \in \mathsf{MinIncon}(\Delta)$ such that $\Psi \subset Y$ and $\Phi \cap Y \neq \emptyset$. Therefore, for every $X \in \mathsf{MinIncon}(\Delta)$ such that $\Phi \subset X$, it immediately follows that $X \cap Y \neq \emptyset$. ∎

Theorem 7.2.1 Let Compilation$(\Delta) = (N, A)$. If $\langle \Phi, \alpha \rangle$ is the root of a complete argument tree T and $\langle \Phi, \alpha \rangle$ is the parent node of $\langle \Psi, \diamond \rangle$, then the subtree of T rooted at $\langle \Psi, \diamond \rangle$ is isomorphic to a subgraph of (N, A).

Proof Consider a branch $\langle \Psi_0, \alpha \rangle, \langle \Psi_1, \diamond \rangle, \ldots, \langle \Psi_m, \diamond \rangle$ in T (i.e., Ψ_0 is Φ). The main step is to show that for $1 \leq i \leq m - 1$, the fact that $\langle \Psi_{i+1}, \diamond \rangle$ is a canonical undercut for $\langle \Psi_i, \diamond \rangle$ entails that there exist X_i and X_{i+1} in N satisfying $(X_i, X_{i+1}) \in A$.

Applying proposition 7.2.4, there exists $X_i \in N$ such that $\Psi_i = X_i \backslash \Psi_{i-1}$ and there exists $X_{i+1} \in N$ such that $\Psi_{i+1} = X_{i+1} \backslash \Psi_i$. Clearly, $X_i = \Psi_i \cup \Psi_{i-1}$ and $X_{i+1} = \Psi_{i+1} \cup \Psi_i$. Since Ψ_i is trivially disjoint from Ψ_{i-1} and Ψ_{i+1}, it then follows that $X_i = X_{i+1}$ implies $\Psi_{i-1} = \Psi_{i+1}$. By the definition of an argument tree, the support of an argument in T cannot be a subset of the support of an ancestor: Here, $\Psi_{i-1} \neq \Psi_{i+1}$. Therefore, $X_i \neq X_{i+1}$. As $X_i \cap X_{i+1} \neq \emptyset$ is trivial, $(X_i, X_{i+1}) \in A$ ensues.

There remains to show that no such branch is accounted for by an unfolding of a cycle in (N, A). Assume $X_j = X_{j+k}$ for some j and k. Similarly to the above observation, $X_j = \Psi_j \cup \Psi_{j-1}$ and $X_{j+k} = \Psi_{j+k} \cup \Psi_{j+k-1}$. Thus, $\Psi_j \cup \Psi_{j-1} = \Psi_{j+k} \cup \Psi_{j+k-1}$. Hence, $\Psi_{j+k} \subseteq \Psi_j \cup \Psi_{j-1}$. However, the definition of an argument tree prohibits the support of an argument to be a subset of the set-theoretic union of the supports of its ancestors. That is, a contradiction arises. Therefore, $X_j = X_{j+k}$ is impossible.

Lastly, it can be seen that two siblings nodes cannot be accounted for by the same element in (N, A). Indeed, consider $\langle \Psi', \diamond \rangle$ and $\langle \Psi'', \diamond \rangle$ in T sharing the same parent node $\langle \Psi, \diamond \rangle$. Assume that proposition 7.2.4 yields the same X in N for $\langle \Psi', \diamond \rangle$ and $\langle \Psi'', \diamond \rangle$. Then, $\Psi' = X \backslash \Psi$ and $\Psi'' = X \backslash \Psi$. Trivially, $\Psi' = \Psi''$ ensues, but no two distinct canonical undercuts for the same argument can have the same support. Therefore, the assumption must be false, and the proof is over. ■

Theorem 7.2.2 For a knowledgebase Δ, let Compilation$(\Delta) = (N, A)$; let $\langle \Gamma, \alpha \rangle$ be an argument. For all $X \in N$ such that $\Gamma \subseteq X$, if $\langle \Psi, \diamond \rangle$ is an undercut for $\langle \Gamma, \alpha \rangle$, then

- either $\Psi = X \backslash \Gamma$
- or $\Psi = Y \backslash \Gamma$ for some $Y \in N$ such that $(X, Y) \in A$ and $Y \cap \Gamma \neq \emptyset$

Proof Let $X \in N$ satisfy $\Gamma \subseteq X$, and let $\langle \Psi, \diamond \rangle$ be an undercut for $\langle \Gamma, \alpha \rangle$. It is sufficient to make the hypothesis $\Psi \neq X \backslash \Gamma$ and then prove $\Psi = Y \backslash \Gamma$ for some $Y \in N$ such that $Y \cap \Gamma \neq \emptyset$ and $(X, Y) \in A$.

$\langle\Psi, \diamond\rangle$ is an undercut for $\langle\Gamma, \alpha\rangle$; hence, Ψ is a minimal consistent subset of Δ such that $\Psi \vdash \neg(\gamma_1 \wedge \cdots \wedge \gamma_n)$ where $\Gamma = \{\gamma_1, \ldots, \gamma_n\}$. By classical logic, there exists a minimal subset $\Gamma' = \{\gamma'_1, \ldots, \gamma'_m\}$ of Γ that satisfies $\Psi \vdash \neg(\gamma'_1 \wedge \cdots \wedge \gamma'_m)$ for some $m > 0$ (i.e., Γ' is nonempty). Thus, $\Psi \cup \Gamma'$ is minimally inconsistent.

Reasoning by reductio ad absurdum, assume some $\gamma_i \in \Gamma \cap \Psi$. Then, $\langle\Psi, \diamond\rangle$ being an undercut for $\langle\Gamma, \alpha\rangle$ means that $\{\gamma_i\} \cup (\Psi\backslash\{\gamma_i\}) \vdash \neg(\gamma_1 \wedge \cdots \wedge \gamma_n)$. By classical logic, $\Psi\backslash\{\gamma_i\} \vdash \neg(\gamma_1 \wedge \cdots \wedge \gamma_n)$ ensues, and Ψ would not be a minimal subset of Δ entailing $\neg(\gamma_1 \wedge \cdots \wedge \gamma_n)$; a contradiction arises. Therefore, $\Psi \cap \Gamma = \emptyset$.

Reasoning by reductio ad absurdum again, assume $\Psi \subseteq X$. As $\langle\Psi, \diamond\rangle$ is an undercut for $\langle\Gamma, \alpha\rangle$, it follows that $\Psi \cup \Phi$ is inconsistent. $X \in N$ means that X is minimally inconsistent. Hence, $\Gamma \subseteq X$ and $\Psi \subseteq X$ imply $X = \Gamma \cup \Psi$. It has been shown above that Ψ and Γ are disjoint; therefore, $\Psi = X\backslash\Gamma$, and a contradiction arises. Hence, the assumption must be false. Thus, $\Psi \not\subseteq X$. That is, there exists $\psi \in \Psi$ such that $\psi \notin X$.

Take Y to be $\Psi \cup \Gamma'$. (1) As $\Psi \cap \Gamma = \emptyset$ has been shown, $\Psi = Y\backslash\Gamma$. (2) $\Psi \cup \Gamma'$ has been shown to be minimally inconsistent; therefore, $Y \in N$. (3) It has been shown that Γ' is nonempty; hence, $\Gamma' \subseteq \Gamma$ yields $\Gamma \cap Y \neq \emptyset$. (4) Lastly, $\Gamma' \subseteq X \cap Y$ ensues from $\Gamma' \subseteq \Gamma \subseteq X$, and it follows that $X \cap Y$ is not empty because Γ' is not. Now, $\Psi \subseteq Y$, but it has been shown that $\Psi \not\subseteq X$; hence, $X \neq Y$. An easy consequence of $X \cap Y \neq \emptyset$ and $X \neq Y$ is $(X, Y) \in A$. ■

Theorem 7.2.3 For a knowledgebase Δ, a hypergraph $\mathsf{Compilation}(\Delta) = (N, A)$, and an argument $\langle\Phi, \alpha\rangle$, such that $\Phi \subseteq \Delta$, the algorithm `Undercuts`$(\langle\Phi, \alpha\rangle, N, A)$ returns the canonical undercuts for a complete argument tree T where the root of T is $\langle\Phi, \alpha\rangle$.

Proof From proposition 7.2.2, we see that the call to the `FirstLevel` algorithm ensures that all the canonical undercuts for $\langle\Phi, \alpha\rangle$ are returned. From proposition 7.2.2, each of the undercuts obtained via the `FirstLevel` algorithm are indeed canonical undercuts for $\langle\Phi, \alpha\rangle$.

From proposition 7.2.2, we see that each call to the `Subcuts` algorithm returns all canonical undercuts to canonical undercuts, since given an undercut $\langle\Gamma, \diamond\rangle$ where $\Gamma \subset X$ for some $X \in N$, all undercuts $\langle Y\backslash\Gamma, \diamond\rangle$ are returned where $(X, Y) \in A$ and $Y \not\subseteq S$. The `Subcuts` algorithm is called by recursion to obtain all canonical undercuts to each canonical undercut by recursion. Furthermore, from proposition 7.2.5, each of the undercuts obtained via the `Subcuts` algorithm are indeed canonical undercuts in T. ■

Proposition 7.2.7 Let $n = |\Delta|$ and $m = \lfloor n/2 \rfloor$. If $\Delta \vdash \bot$, then $1 \leq |\mathsf{MinIncon}(\Delta)| \leq \mathsf{Combinations}(n, m)$.

Proof For any $\Phi \in \mathsf{MinIncon}(\Delta)$, there is no $\Psi \in \mathsf{MinIncon}(\Delta)$ such that $\Phi \subset \Psi$ or $\Psi \subset \Phi$. The largest subset of $\wp(\Delta)$ for which no element is a subset of any other is the set of subsets of Δ that are of cardinality $\lfloor n/2 \rfloor$. The number of these subsets is given by $\mathsf{Combinations}(n, m)$. ∎

Proposition 7.3.1 For Δ and α, if $|\Delta| = n$, then $|\mathsf{NaiveQuerying}(\Delta, \alpha)| = 2^{n+1}$.

Proof Recall $\mathsf{NaiveQuerying}(\Delta, \alpha) = \{(\Phi?\alpha) \mid \Phi \subseteq \Delta\} \cup \{(\Phi?\bot) \mid \Phi \subseteq \Delta\}$. Since $|\{(\Phi?\alpha) \mid \Phi \subseteq \Delta\}| = 2^n$ and $|\{(\Phi?\bot) \mid \Phi \subseteq \Delta\}| = 2^n$, we get that $|\mathsf{NaiveQuerying}(\Delta, \alpha)| = 2^n + 2^n = 2^{n+1}$. ∎

Proposition 7.3.2 For any $\Phi \subseteq \Delta$, $\Phi \in \mathsf{C}(\alpha)$ iff (1) for all $\Psi \in \mathsf{U}(\alpha)$, $\Psi \not\subset \Phi$, and (2) for all $\Psi \in \mathsf{L}(\alpha)$, $\Phi \not\subseteq \Psi$.

Proof ($\Rightarrow$) If $\Phi \in \mathsf{C}(\alpha)$, then $\Phi \vdash \alpha$ and there is no $\Psi \subset \Phi$ such that $\Psi \vdash \alpha$. Hence, Φ is a minimal element in $\mathsf{U}(\alpha)$, and so for all $\Psi \in \mathsf{U}(\alpha)$, $\Psi \not\subset \Phi$. Furthermore, since for all $\Psi \in \mathsf{L}(\alpha)$, $\Psi \nvdash \alpha$, we have for all $\Psi \in \mathsf{L}(\alpha)$, $\Phi \not\subseteq \Psi$. ($\Leftarrow$) Let $\Phi \subseteq \Delta$ be such that (1) for all $\Psi \in \mathsf{U}(\alpha)$, $\Psi \not\subset \Phi$, and (2) for all $\Psi \in \mathsf{L}(\alpha)$, $\Phi \not\subseteq \Psi$. From (1), Φ is a minimal element in $\mathsf{U}(\alpha)$, or Φ is an element in $\mathsf{L}(\alpha)$. From (2), it is not the case that Φ is an element in $\mathsf{L}(\alpha)$. Hence, $\Phi \vdash \alpha$, and there is no $\Psi \subset \Phi$ such that $\Psi \vdash \alpha$. Therefore $\Phi \in \mathsf{C}(\alpha)$. ∎

Proposition 7.3.3 For any Φ and α, $\Phi \in \mathsf{C}(\alpha) \cap \mathsf{L}(\bot)$ iff $\langle \Phi, \alpha \rangle$ is an argument.

Proof ($\Rightarrow$) From $\Phi \in \mathsf{C}(\alpha)$, we get that $\Phi \vdash \alpha$ and there is no $\Psi \subset \Phi$ such that $\Psi \vdash \alpha$. From $\Phi \in \mathsf{L}(\bot)$, $\Phi \nvdash \bot$. Hence, $\langle \Phi, \alpha \rangle$ is an argument. ($\Leftarrow$) If $\langle \Phi, \alpha \rangle$ is an argument, then (1) $\Phi \vdash \alpha$ and (2) there is no $\Psi \subset \Phi$ such that $\Psi \vdash \alpha$, and (3) $\Phi \nvdash \bot$. Therefore $\Phi \in \mathsf{C}(\alpha)$, and $\Phi \in \mathsf{L}(\bot)$. ∎

Proposition 7.3.4 For any α, β, $\mathsf{C}(\alpha \wedge \beta) \subseteq (\mathsf{C}(\alpha) \oplus \mathsf{C}(\beta)) \subseteq \mathsf{U}(\alpha \wedge \beta)$.

Proof If $\Phi \in \mathsf{C}(\alpha \wedge \beta)$, then $\Phi \in \mathsf{U}(\alpha) \cap \mathsf{U}(\beta)$. Therefore, there is a $\Phi_1 \subseteq \Phi$ such that $\Phi_1 \in \mathsf{C}(\alpha)$, and there is a $\Phi_2 \subseteq \Phi$ such that $\Phi_2 \in \mathsf{C}(\beta)$. Since there is no $\Phi' \subset \Phi$ such that $\Phi' \in \mathsf{C}(\alpha \wedge \beta)$, then $\Phi = \Phi_1 \cup \Phi_2$ for some $\Phi_1 \in \mathsf{C}(\alpha)$ and $\Phi_2 \in \mathsf{C}(\beta)$. Furthermore, by the definition of $\oplus$, $\Phi_1 \cup \Phi_2 \in \mathsf{C}(\alpha) \oplus \mathsf{C}(\beta)$. Therefore, $\Phi \in (\mathsf{C}(\alpha) \oplus \mathsf{C}(\beta))$. Now, if $\Phi \in \mathsf{C}(\alpha) \oplus \mathsf{C}(\beta)$, then there is a $\Phi_1 \in \mathsf{C}(\alpha)$ and there is a $\Phi_2 \in \mathsf{C}(\beta)$ such that $\Phi = \Phi_1 \cup \Phi_2$. Therefore, $\Phi \vdash \alpha \wedge \beta$, and so $\Phi \in \mathsf{U}(\alpha \wedge \beta)$. ∎

Proposition 7.3.6 For any α, $(\mathsf{C}(\neg\alpha) \cap \mathsf{L}(\bot)) \subseteq (\div\mathsf{C}(\alpha) \cap \mathsf{L}(\bot))$.

Proof We show the contrapositive (i.e., $\Phi \notin \div(\mathsf{C}(\alpha) \cap \mathsf{L}(\bot))$, then $\Phi \notin (C(\neg\alpha) \cap \mathsf{L}(\bot))$. If $\Phi \notin \div(\mathsf{C}(\alpha) \cap \mathsf{L}(\bot))$, then exists a $\Psi \in (\mathsf{C}(\alpha) \cap \mathsf{L}(\bot))$, such that $(\Phi \subseteq \Psi$ or $\Psi \subseteq \Phi)$ or $\Phi \vdash \bot$. Since $\Psi \not\vdash \bot$ and $\Psi \vdash \alpha$, it follows that $\Psi \not\vdash \neg\alpha$. Thus, if $\Phi \subseteq \Psi$, $\Phi \not\vdash \neg\alpha$, and if $\Psi \subseteq \Phi$, then either $\Phi \not\vdash \neg\alpha$ or $\Phi \vdash \bot$. Hence, $\Phi \notin \mathsf{C}(\neg\alpha) \cap \mathsf{L}(\bot)$. ■

Proposition 7.3.7 If $\langle\Phi, \alpha\rangle$ is an argument, then $\Phi \in (\mathsf{C}(\Pi, \alpha) \cup \div\mathsf{C}(\Pi, \alpha))$.

Proof Let $\langle\Phi, \alpha\rangle$ be an argument. Therefore, exactly one of the following two cases holds: (1) $(\Phi \vdash \alpha) \in \Pi$, and for all $\phi \in \Phi$, $(\Phi \setminus \{\phi\} \not\vdash \alpha) \in \Pi$; or (2) $(\Phi \vdash \alpha) \notin \Pi$, or there is a $\phi \in \Phi$ such that $(\Phi \setminus \{\phi\} \not\vdash \alpha) \notin \Pi$. If case (1) holds, then $\Phi \in (\mathsf{C}(\Pi, \alpha))$. Now we consider case (2). Since $\langle\Phi, \alpha\rangle$ is an argument, there is no Ψ such that $(\Psi \subset \Phi$ or $\Phi \subset \Psi)$ and $(\Psi \vdash \alpha) \in \Pi$ and for all $\psi \in \Psi$, $(\Psi \setminus \{\psi\} \not\vdash \alpha) \in \Pi$. Hence, if case (2) holds, then $\Phi \in \div\mathsf{C}(\Pi, \alpha)$. Therefore, if $\langle\Phi, \alpha\rangle$ is an argument, then $\Phi \in (\mathsf{C}(\Pi, \alpha) \cup \div\mathsf{C}(\Pi, \alpha))$. ■

Proposition 7.3.9 For any α, if $|\Delta| = n$, then $1 \leq \mathsf{C}(\alpha) \leq \mathsf{MaxWidth}(n)$.

Proof For any $\Phi \in \mathsf{C}(\alpha)$, there is no $\Psi \in \mathsf{C}(\alpha)$ such that $\Phi \subset \Psi$ or $\Psi \subset \Phi$. The largest subset of $\wp(\Delta)$ for which no element is a subset of any other is the set of subsets of Δ that are of cardinality $\lfloor n/2 \rfloor$. The number of these subsets is $\mathsf{MaxWidth}(n)$. ■

Proposition 7.3.10 Let Π be such that $\mathsf{C}(\Pi, \alpha) = \mathsf{C}(\alpha)$ For any k, if $\langle\Phi, \alpha\rangle$ is an argument, then there is a $\Psi \in \mathsf{C}(\Pi, \alpha, k)$ such that $\Phi \subseteq \Psi$.

Proof For each $\Gamma \in \mathsf{C}(\alpha)$, there is a $\Psi \in \mathsf{C}(\Pi, \alpha, k)$ such that $\Gamma \subseteq \Psi$. Thus, if $\langle\Phi, \alpha\rangle$ is an argument, then $\Phi \in \mathsf{C}(\alpha)$, and so there is a $\Psi \in \mathsf{C}(\Pi, \alpha, k)$ such that $\Phi \subseteq \Psi$. ■

Proposition 7.4.1 If T_2 is a refinement of T_1 with refinement function f, then for all $[\Phi_1, \alpha_1] \in T_1$, if $f(\langle\Phi_1, \alpha_1\rangle) = \langle\Phi_2, \alpha_2\rangle$, then $\langle\Phi_2, \alpha_2\rangle$ is more conservative than $\langle\Phi_1, \alpha_1\rangle$.

Proof Assume $f(\langle\Phi_1, \alpha_1\rangle) = \langle\Phi_2, \alpha_2\rangle$. Hence, $\mathsf{Support}(\Phi_2) \subseteq \mathsf{Support}(\Phi_1)$ and $\alpha_1 \vdash \alpha_1$. Hence, by definition 3.2.2, $\langle\Phi_2, \alpha_2\rangle$ is more conservative than $\langle\Phi_1, \alpha_1\rangle$. ■

Proposition 7.4.2 If T_n is a complete preargument tree for α, then there is a sequence $\langle T_1, \ldots, T_n\rangle$ of approximate trees for α such that T_1 is an altoment tree with just a root node, and for each i, where $i < n$, T_{i+1} is obtained by a revision step from T_i.

Proof Assume T_1 is an altoment tree with just a root node. Resupport can be applied exhaustively to this node. Suppose resupport is applied exclusively and exhautively i times (i.e., T_2 from T_1 is obtained by resupport and ... and T_i from T_{i-1} is obtained by resupport), then T_i is an argument tree with only a root node. Now suppose T_j is such that each node is an argument, and for each $\langle \Phi, \alpha \rangle$ in T_j, if $\langle \Psi, \beta \rangle$ is a child of $\langle \Phi, \alpha \rangle$, then β is $\neg(\wedge\Phi)$. If a new child can be added to T_j using the expansion step, then resupport can be applied exclusively and exhaustively so that after a number of steps, the child has become a canonical undercut or it can be removed by contraction. Thus, by interleaving expansion, with exclusive and exhaustive use of resupport and contraction, a complete argument tree T_n can eventually be obtained. Hence, there a sequence $\langle T_1, \ldots, T_n \rangle$ of approximate trees for α such that T_1 is an altoment tree with just a root node, and for each i, where $i < n$, T_{i+1} is obtained by a revision step from T_i. ■

Proposition 7.4.3 Let $\langle T_1, \ldots, T_n \rangle$ be a sequence of entailment trees for α such that T_1 is an altoment tree with just a root node, and for each i, where $1 \leq i < n$, T_{i+1} is obtained by a revision step from T_i. If T_n is such that no further revision steps are possible on it, then T_n is a complete preargument tree for α.

Proof If T_n is such that no further revision steps are possible, then we can draw the following observations. (Observation 1) Because resupport cannot be applied, for all $\langle \Phi, \alpha \rangle$ in T_n, $\langle \Phi, \alpha \rangle$ is a miniment, and because contraction cannot be applied, for all $\langle \Phi, \alpha \rangle$ in T_n, $\Phi \nvdash \perp$. Hence, for all $\langle \Phi, \alpha \rangle$ in T_n, $\langle \Phi, \alpha \rangle$ is an argument. (Observation 2) Because reconsequent and deflation cannot be applied, for all $\langle \Phi, \alpha \rangle$ in T_n, if $\langle \Psi, \beta \rangle$ is a child of $\langle \Phi, \alpha \rangle$, then β is $\neg(\wedge\Phi)$. (Observation 3) Because expansion cannot be applied, for all $\langle \Phi, \alpha \rangle$ in T_n, there are no further arguments of the form $\langle \Psi, \neg(\wedge\Phi) \rangle$ that can be added to the tree. (Observation 4) Since all nodes (apart from the root) have been added by expansion, if a node $\langle \Psi, \beta \rangle$ has ancestor nodes $\langle \Phi_1, \beta_1 \rangle, \ldots, \langle \Phi_n, \beta_n \rangle$, then Ψ is not a subset of $\Phi_1 \cup \cdots \cup \Phi_n$. These observations imply that T_n is a complete argument tree. ■

Bibliography

[ABGM00] G. Antoniou, D. Billington, G. Governatori, and M. Maher. A flexible framework for defeasible logics. In *Proceedings of the 17th American National Conference on Artificial Intelligence (AAAI 2000)*, pages 405–410. MIT Press, 2000.

[ABM05] K. Atkinson, T. Bench-Capon, and P. McBurney. Persuasive political argument. In *Proceedings of the IJCAI Workshop on Computational Models of Natural Argument (CNMA 2005)*, pages 44–51. IJCAI, 2005.

[ABP05] L. Amgoud, J.-F. Bonnefon, and H. Prade. An argumentation-based approach to multiple criteria decision. In *Proceedings of the 8th European Conference on Symbolic and Quantitative Approaches to Reasoning with Uncertainty (ECSQARU 2005)*, volume 3571 of *LNCS*, pages 269–280. Springer, 2005.

[AC98] L. Amgoud and C. Cayrol. On the acceptability of arguments in preference-based argumentation. In G. Cooper and S. Moral, editors, *Proceedings of the 14th Conference on Uncertainty in Artificial Intelligence (UAI 1998)*, pages 1–7. Morgan Kaufmann, 1998.

[AC02] L. Amgoud and C. Cayrol. A model of reasoning based on the production of acceptable arguments. *Annals of Mathematics and Artificial Intelligence*, 34:197–215, 2002.

[ACL96] L. Amgoud, C. Cayrol, and D. Le Berre. Comparing arguments using preference orderings for argument-based reasoning. In *Proceedings of the 8th IEEE International Conference on Tools with Artificial Intelligence (ICTAI 1996)*, pages 400–403. IEEE Computer Society, 1996.

[ACL04] L. Amgoud, C. Cayrol, and M.-Ch. Lagasquie-Schiex. On the bipolarity in argumentation frameworks. In *Proceedings of the 10th Non-Monotonic Reasoning Workshop (NMR 2004)*, pages 1–10. Simon Fraser University, 2004.

[Amg05] L. Amgoud. A general argumentation framework for inference and decision making. In *Proceedings of the 21st Conference on Uncertainty in Artificial Intelligence (UAI 2005)*, pages 26–33. AAAI Press, 2005.

[AMP00] L. Amgoud, N. Maudet, and S. Parsons. Arguments, dialogue and negotiation. In *Proceedings of the 14th European Conference on Artificial Intelligence (ECAI 2000)*, pages 338–342. IOS Press, 2000.

[Ant97] G. Antoniou. *Non-monotonic Reasoning*. MIT Press, 1997.

[AP05a] L. Amgoud and H. Prade. Formal handling of threats and rewards in a negotiation dialogue. In *Proceedings of the 4th International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS 2005)*, pages 537–543. ACM Press, 2005.

[AP05b] L. Amgoud and H. Prade. Handling threats, rewards and explanatory arguments in a unified setting. *International Journal of Intelligent Systems*, 20(12):1195–1218, 2005.

[AR06] G. Aubry and V. Risch. Managing deceitful arguments with x-logics. In *Proceedings of the 11th International Workshop on Non-Monotonic Reasoning (NMR 2006)*, pages 418–425. Technische Universität Clausthal, 2006.

[AS01] J. Augusto and G. Simari. Temporal defeasible reasoning. *Knowledge and Information Systems*, 3:287–318, 2001.

[ASP04] ASPIC. Review on argumentation technology: State of the art, technical and user requirements. Technical report, ASPIC Project Deliverable, 2004.

[BBF$^+$96] H. Barringer, D. Brough, M. Fisher, D. Gabbay, I. Hodkinson, A. Hunter, P. McBrien, R. Owens, and M. Reynolds. Languages, meta-languages, and metatem: A discussion paper. *Journal of the Interest Group for Pure and Applied Logic*, 4:255–272, 1996.

[BBJ02] G. Boolos, J. Burgess, and R. Jeffrey. *Computability and Logic*. Cambridge University Press, 2002.

[BD04] Ph. Besnard and S. Doutre. Characterization of semantics for argumentation systems. In *Proceedings of the 9th International Conference on Principles of Knowledge Representation and Reasoning (KR 2004)*, pages 183–193. AAAI Press, 2004.

[BDK97] G. Brewka, J. Dix, and K. Konolige. *Non-monotonic reasoning: An overview*. CLSI Publications, 1997.

[BDKT97] A. Bondarenko, P. Dung, R. Kowalski, and F. Toni. An abstract argumentation-theoretic approach to default reasoning. *Artificial Intelligence*, 93:63–101, 1997.

[BDP93] S. Benferhat, D. Dubois, and H. Prade. Argumentative inference in uncertain and inconsistent knowledge bases. In *Proceedings of the 9th Annual Conference on Uncertainty in Artificial Intelligence (UAI 1993)*, pages 1449–1445. Morgan Kaufmann, 1993.

[BDP95] S. Benferhat, D. Dubois, and H. Prade. A logical approach to reasoning under inconsistency in stratified knowledge bases. In Ch. Froidevaux and J. Kohlas, editors, *Symbolic and Quantitative Approaches to Reasoning and Uncertainty (ECSQARU 1995)*, volume 956 of *LNCS*, pages 36–43. Springer, 1995.

[BE04] S. Benferhat and R. El Baida. Handling conflicts in access control models. In *Proceedings of the 16th European Conference on Artificial Intelligence (ECAI 2004)*, pages 961–962. IOS Press, 2004.

[Ben03] T. Bench-Capon. Persuasion in practical argument using value based argumentation frameworks. *Journal of Logic and Computation*, 13(3):429–448, 2003.

[Bes89] Ph. Besnard. *An Introduction to Default Logic*. Springer, 1989.

[BG02] P. Baroni and M. Giacomin. Argumentation through a distributed self-stabilizing approach. *Journal of Experimental and Theoretical Artificial Intelligence*, 14(4):273–301, 2002.

[BH00] Ph. Besnard and A. Hunter. Towards a logic-based theory of argumentation. In *Proceedings of the 17th National Conference on Artificial Intelligence (AAAI 2000)*, pages 411–416. MIT Press, 2000.

[BH01] Ph. Besnard and A. Hunter. A logic-based theory of deductive arguments. *Artificial Intelligence*, 128:203–235, 2001.

[BH05] Ph. Besnard and A. Hunter. Practical first-order argumentation. In *Proceedings of the 20th National Conference on Artificial Intelligence (AAAI 2005)*, pages 590–595. MIT Press, 2005.

[BH06] Ph. Besnard and A. Hunter. Knowledgebase compilation for efficient logical argumentation. In *Proceedings of the 10th International Conference on Knowledge Representation (KR 2006)*, pages 123–133. AAAI Press, 2006.

[Bil87] M. Billig. *Arguing and Thinking: A Rhetorical Approach to Social Psychology*. Cambridge University Press, 1987.

[Bow07] R. Bowerman. *Tracking and Judging Debates Using Argumentation: The Role of Confirmation, Preclusion, and Reflection*. PhD thesis, University College London, 2007.

[Bre89] G. Brewka. Preferred subtheories: An extended logical framework for default reasoning. In *Proceedings of the 11th International Conference on Artificial Intelligence (IJCAI 1989)*, pages 1043–1048. Morgan Kaufmann, 1989.

[Bre91] G. Brewka. *Non-monotonic Reasoning: Logical Foundations of Commonsense*. Cambridge University Press, 1991.

[Bre94] G. Brewka. Reasoning about priorities in default logic. In *Proceedings of the 12th National Conference on Artificial Intelligence (AAAI 1994)*, pages 420–425. MIT Press, 1994.

[Buc07] S. Buckingham Shum. Modelling naturalistic argumentation in research literatures: Representation and interaction design issues. *International Journal of Intelligent Systems*, 22(1):17–47, 2007. Special Issue on Computational Modelling of Naturalistic Argumentation.

[CA05] M. Caminada and L. Amgoud. An axiomatic account of formal argumentation. In *Proceedings of the 20th National Conference on Artificial Intelligence (AAAI 2005)*, pages 608–613, 2005.

[Can93] R. Cann. *Formal Semantics: An Introduction*. Cambridge University Press, 1993.

[CC92] R. Cockcroft and S. Cockcroft. *Persuading People: An Introduction to Rhetoric*. Macmillan Press, 1992.

[CCLR01] T. Cormen, S. Clifford, C. Leiserson, and R. Rivest. *Introduction to Algorithms*. MIT Press, 2nd edition, 2001.

[CCS05] M. Capobianco, C. Chesñevar, and G. Simari. Argumentation and the dynamics of warranted beliefs in changing environments. *International Journal on Autonomous Agents and Multiagent Systems*, 11:127–151, 2005.

[CDK$^+$05] S. Coste-Marquis, C. Devred, S. Konieczny, M.-C. Lagasquie-Schiex, and P. Marquis. Merging argumentation systems. In *Proceedings of the 20th National Conference on Artificial Intelligence (AAAI 2005)*, pages 614–619. MIT Press, 2005.

[CDM01] C. Cayrol, S. Doutre, and J. Mengin. Dialectical proof theories for the credulous preferred semantics of argumentation frameworks. In *Quantitative and Qualitative Approaches to Reasoning with Uncertainty (ECSQARU 2001)*, volume 2143 of *LNCS*, pages 668–679. Springer, 2001.

[CDM03] C. Cayrol, S. Doutre, and J. Mengin. On decision problems related to the preferred semantics for argumentation frameworks. *Journal of Logic and Computation*, 13(3):377–403, 2003.

[CDM05] S. Coste-Marquis, C. Devred, and P. Marquis. Prudent semantics for argumentation frameworks. In *Proceedings of the 17th International Conference on Tools with Artificial Intelligence (ICTAI 05)*, pages 568–572, 2005.

[CDM06] S. Coste-Marquis, C. Devred, and P. Marquis. Constrained argumentation frameworks. In *Proceedings of the 10th International Conference on Principles of Knowledge Representation and Reasoning (KR 2006)*, pages 112–122. AAAI Press, 2006.

[CH93] J. Cussens and A. Hunter. Using maximum entropy in a defeasible logic with probabilistic semantics. In *Proceedings of the 4th International Conference in Information Processing and Management of Uncertainty in Knowledge-based Systems (IPMU 1992)*, volume 682 of *LNCS*, pages 43–52. Springer, 1993.

[Cho95] L. Cholvy. Automated reasoning with merged contradictory information whose reliability depends on topics. In *Symbolic and Quantitative Approaches to Reasoning and Uncertainty (ECSQARU 1995)*, volume 946 of *LNCS*, pages 125–132. Springer, 1995.

[CHS93] J. Cussens, A. Hunter, and A. Srinivasan. Generating explicit orderings for nonmonotonic logics. In *Proceedings of the 11th National Conference on Artificial Intelligence (AAAI 1993)*, pages 420–425. MIT Press, 1993.

[CL05a] C. Cayrol and M.-C. Lagasquie-Schiex. Graduality in argumentation. *Journal of Artificial Intelligence Research*, 23:245–297, 2005.

[CL05b] C. Cayrol and M.-C. Lagasquie-Schiex. Gradual valuation for bipolar argumentation frameworks. In *Proceedings of the 8th European Conference on Symbolic and Quantitative Approaches to Reasoning and Uncertainty (ECSQARU 2005)*, volume 3571 of *LNCS*, pages 366–377. Springer, 2005.

[CM04a] C. Chesñevar and A. Maguitman. Arguenet: An argument-based recommender system for solving Web search queries. In *Proceedings of the 2nd International Conference on Intelligent Systems (IS 2004)*, pages 282–287. IEEE Press, 2004.

[CM04b] C. Chesñevar and A. Maguitman. An argumentative approach to assessing natural language usage based on the Web corpus. In *Proceedings of the 16th European Conference on Artificial Intelligence (ECAI 2004)*, pages 581–585. IOS Press, 2004.

[CM05] C. Chesñevar and A. Maguitman. Combining argumentation and Web search technology: Towards a qualitative approach for ranking results. *Journal of Advanced Computational Intelligence and Intelligent Informatics*, 9:53–60, 2005.

[CML00] C. Chesñevar, A. Maguitman, and R. Loui. Logical models of argument. *ACM Computing Surveys*, 32:337–383, 2000.

[CMS06] C. Chesñevar, A. Maguitman, and G. Simari. Argument-based critics and recommenders: A qualitative perspective on user support systems. *Data and Knowledge Engineering*, 59:293–319, 2006.

[CRL00] D. Carbogim, D. Robertson, and J. Lee. Argument-based applications to knowledge engineering. *Knowledge Engineering Review*, 15:119–149, 2000.

[CRS93] C. Cayrol, V. Royer, and C. Saurel. Management of preferences in assumption based reasoning. In *Proceedings of the 4th International Conference on Information Processing and Management of Uncertainty in Knowledge based Systems (IPMU 1992)*, volume 682 of *LNCS*, pages 13–22. Springer, 1993.

[CSG05] C. Chesñevar, G. Simari, and L. Godo. Computing dialectical trees efficiently in possibilistic defeasible logic programming. In *Proceedings of the 8th International Logic Programming and Nonmonotonic Reasoning Conference (LPNMR 2005)*, volume 3662 of *LNCS*, pages 158–171. Springer, 2005.

[Dal88] M. Dalal. Investigations into a theory of knowledge base revision: Preliminary report. In *Proceedings of the 7th National Conference on Artificial Intelligence (AAAI 1988)*, pages 3–7. MIT Press, 1988.

[DB02] P. Dunne and T. Bench-Capon. Coherence in finite argument systems. *Artificial Intelligence*, 141:187–203, 2002.

[DKT06] P. Dung, R. Kowalski, and F. Toni. Dialectical proof procedures for assumption-based admissible argumentation. *Artificial Intelligence*, 170:114–159, 2006.

[DM04] S. Doutre and J. Mengin. On sceptical versus credulous acceptance for abstract argumentation systems. In *Proceedings of the 9th European Conference on Logics in Artificial Intelligence (JELIA 2004)*, volume 3229 of *LNCS*, pages 462–473. Springer, 2004.

[DMT02] P. Dung, P. Mancarella, and F. Toni. Argumentation-based proof procedures for credulous and sceptical non-monontonic reasoning. In *Computational Logic: Logic Programming and Beyond: Essays in Honour of Robert A. Kowalski*, volume 2408 of *LNCS*, pages 289–310. Springer, 2002.

[DN99] B. Duval and P. Nicolas. Learning default theories. In *Symbolic and Quantitative Approaches to Reasoning and Uncertainty (ECSQARU 1999)*, volume 1638 of *LNCS*, pages 148–159. Springer, 1999.

[DNT02] Y. Dimopoulos, B. Nebel, and F. Toni. On the computational complexity of assumption-based argumentation for default reasoning. *Artificial Intelligence*, 141:57–78, 2002.

[Dun95] P. Dung. On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming, and n-person games. *Artificial Intelligence*, 77:321–357, 1995.

[EG95] T. Eiter and G. Gottlob. The complexity of logic-based abduction. *Journal of the ACM*, 42:3–42, 1995.

[EGH95] M. Elvang-Goransson and A. Hunter. Argumentative logics: Reasoning from classically inconsistent information. *Data and Knowledge Engineering*, 16:125–145, 1995.

[FD00] J. Fox and S. Das. *Safe and Sound: Artificial Intelligence in Hazardous Applications*. MIT Press, 2000.

[Fis88] A. Fisher. *The Logic of Real Arguments*. Cambridge University Press, 1988.

[FJR98] J. Fox, N. Johns, and A. Rahmanzadeh. Disseminating medical knowledge: The PROforma approach. *Artificial Intelligence in Medicine*, 14:157–182, 1998.

[FKEG93] J. Fox, P. Krause, and M. Elvang-Gøransson. Argumentation as a general framework for uncertain reasoning. In *Proceedings of the 9th Conference on Uncertainty in Artificial Intelligence (UAI 1993)*, pages 428–434. Morgan Kaufmann, 1993.

[GH06] J. Grant and A. Hunter. Measuring inconsistency in knowledgebases. *Journal of Intelligent Information Systems*, 27:159–184, 2006.

[GHR94] D. Gabbay, I. Hodkinson, and M. Reynolds. *Temporal Logic: Mathematical Foundations and Computational Aspects*. Oxford University Press, 1994.

[GHR95] D. Gabbay, C. Hogger, and J. Robinson, editors. *Handbook of Logic in Artificial Logic and Logic Programming (Volume 4: Epistemic and Temporal Reasoning)*. Oxford University Press, 1995.

[GHR98] D. Gabbay, C. Hogger, and J. Robinson, editors. *Handbook of Logic in Artificial Intelligence and Logic Programming (Volume 3: Nonmonontonic Reasoning and Uncertainty Reasoning)*. Oxford University Press, 1998.

[GJ79] M. Garey and D. Johnson. *Computers and Intractability*. W. H. Freeman, 1979.

[GK97] T. Gordon and N. Karaçapilidis. The Zeno argumentation framework. In *Proceedings of the 6th International Conference on Artificial Intelligence and Law (ICAIL 1997)*, pages 10–18. ACM Press, 1997.

[GK98] J. Gebhardt and R. Kruse. Background and perspectives of possibilistic graphical models. In A. Hunter and S. Parsons, editors, *Applications of Uncertainty Formalisms*. Springer, 1998.

[GM89] G. Gazdar and C. Mellish. *Natural Language Processing in Prolog*. Addison-Wesley, 1989.

[Gra78] J. Grant. Classifications for inconsistent theories. *Notre Dame Journal of Formal Logic*, 19:435–444, 1978.

[GS04] A. García and G. Simari. Defeasible logic programming: An argumentative approach. *Theory and Practice of Logic Programming*, 4:95–138, 2004.

[GS05] M. Guerini and O. Stock. Towards ethical persuasive agents. In *Proceedings of the IJCAI Workshop on Computational Models of Natural Argument (CNMA 2005)*. IJCAI, 2005.

[Gul06] R. Gula. *Nonsense: Red Herrings, Straw Men, and Sacred Cows: How We Abuse Logic in Everyday Language*. Axios Press, 2006.

[Haa78] S. Haack. *Philosophy of Logics*. Cambridge University Press, 1978.

[Hae98] R. Haenni. Modelling uncertainty with propositional assumptions-based systems. In A. Hunter and S. Parsons, editors, *Applications of Uncertainty Formalisms*, pages 446–470. Springer, 1998.

[Hae01] R. Haenni. Cost-bounded argumentation. *Approximate Reasoning*, 26:101–127, 2001.

[Ham78] A. Hamilton. *Mathematical Logic for Mathematicians*. Cambridge University Press, 1978.

[HB05] T. Hollihan and K. Baaske. *Arguments and Arguing*. Waveland Press, 2005.

[HK05] A. Hunter and S. Konieczny. Approaches to measuring inconsistent information. In *Inconsistency Tolerance*, volume 3300 of *LNCS*, pages 189–234. Springer, 2005.

[HK06] A. Hunter and S. Konieczny. Shapley inconsistency values. In *Proceedings of the 10th International Conference on Knowledge Representation (KR 2006)*, pages 249–259. AAAI Press, 2006.

[HKL00] R. Haenni, J. Kohlas, and N. Lehmann. Probabilistic argumentation systems. In D. Gabbay and Ph Smets, editors, *Handbook of Defeasible Reasoning and Uncertainty Management Systems, Volume 5*, pages 221–288. Kluwer, 2000.

[HMLN05] C. Haley, J. Moffett, R. Laney, and B. Nuseibeh. Arguing security: Validating security requirements using structured argumentation. In *Proceedings of the 3rd Symposium on Requirements Engineering for Information Security (SREIS 2005)*, pages 1–8. IEEE, 2005.

[HN98] A. Hunter and B. Nuseibeh. Managing inconsistent specifications: Reasoning, analysis and action. *ACM Transactions on Software Engineering and Methodology*, 7:335–367, 1998.

[Hod77] W. Hodges. *Logic: An Introduction to Elementary Logic*. Penguin, 1977.

[HP06] S. Holbech Nielson and S. Parsons. Computing preferred extensions for argumentation systems with sets of attacking arguments. In P. Dunne and T. Bench-Capon, editors, *Computational Models of Argument (COMMA 2006)*, pages 97–108. IOP Press, 2006.

[Hun01] A. Hunter. Ramification analysis with structured news reports using temporal argumentation. In *Proceedings of the Adventures in Argumentation Workshop (part of the 6th European Conference on Symbolic and Quantitative Approaches to Reasoning with Uncertainty)*, pages 23–28. Institut de Recherche en Informatique de Toulouse, 2001.

[Hun02] A. Hunter. Measuring inconsistency in knowledge via quasi-classical models. In *Proceedings of the 18th National Conference on Artificial Intelligence (AAAI 2002)*, pages 68–73. MIT Press, 2002.

[Hun03] A. Hunter. Evaluating the significance of inconsistency. In *Proceedings of the 18th International Joint Conference on Artificial Intelligence (IJCAI 2003)*, pages 468–473. Morgan Kaufmann, 2003.

[Hun04a] A. Hunter. Making argumentation more believable. In *Proceedings of the 19th National Conference on Artificial Intelligence (AAAI 2004)*, pages 269–274. MIT Press, 2004.

[Hun04b] A. Hunter. Towards higher impact argumentation. In *Proceedings of the 19th National Conference on Artificial Intelligence (AAAI 2004)*, pages 275–280. MIT Press, 2004.

[Hun05] A. Hunter. Presentation of arguments and counterarguments for tentative scientific knowledge. In *2nd International Workshop on Argumentation in Multi-agent Systems (ArgMAS'2005)*, volume 4049 of *LNCS*, pages 245–263. Springer, 2005.

[Hun06a] A. Hunter. Approximate arguments for efficiency in logical argumentation. In *Proceedings of the 11th International Workshop on Non-Monotonic Reasoning (NMR 2006)*, pages 383–389. Technische Universität Clausthal, 2006.

[Hun06b] A. Hunter. Contouring of knowledge for intelligent searching for arguments. In *Proceedings of the 17th European Conference on Artificial Intelligence (ECAI 2006)*, pages 250–254. IOS Press, 2006.

[Hun06c] A. Hunter. How to act on inconsistent news: Ignore, resolve, or reject. *Data and Knowledge Engineering*, 57:221–239, 2006.

[JV00] H. Jakobovits and D. Vermeir. Robust semantics for argumentation frameworks. *Journal of Logic and Computation*, 9:215–261, 2000.

[KAEF95] P. Krause, S. Ambler, M. Elvang-Gorannson, and J. Fox. Is there a role for qualitative risk assessment? In *Proceedings of the 11th Conference on Uncertainty in Artificial Intelligence (UAI 1995)*, pages 386–393. Morgan Kaufmann, 1995.

[KBC03] P. Kirschner, S. Buckingham Shum, and C. Carr, editors. *Visualizing Argumentation: Software Tools for Collaborative and Educational Sense-Making*. Springer, 2003.

[KC93] P. Krause and D. Clark. *Representing Uncertain Knowledge: An Artificial Intelligence Approach*. Intellect, 1993.

[KFJP98] P. Krause, J. Fox, P. Judson, and M. Patel. Qualitative risk assessment fulfils a need. In *Applications of Uncertainty Formalisms*, volume 1455 of *Lecture Notes in Artificial Intelligence*, pages 138–156. Springer, 1998.

[KLM03] S. Konieczny, J. Lang, and P. Marquis. Quantifying information and contradiction in propositional logic through epistemic actions. In *Proceedings of the 18th International Joint Conference on Artificial Intelligence (IJCAI 2003)*, pages 106–111. Morgan Kaufmann, 2003.

[KMD94] A. Kakas, P. Mancarella, and P. Dung. The acceptability semantics for logic programming. In *International Conference on Logic Programming (ICLP 1994)*, pages 504–519. MIT Press, 1994.

[KMG01] R. Kempson, W. Meyer-Viol, and D. Gabbay. *Dynamic Syntax: The Flow of Language Understanding*. Blackwell, 2001.

[Kni01] K. Knight. Measuring inconsistency. *Journal of Philosophical Logic*, 31:77–98, 2001.

[Kni03] K. Knight. Two information measures for inconsistent sets. *Journal of Logic, Language and Information*, 12:227–248, 2003.

[Kor01] F. Koriche. On anytime coherence-based reasoning. In *Proceedings of the 6th European Conference on Quantitative and Qualitative Approaches to Reasoning with Uncertainty (ECSQARU 2001)*, volume 2143 of *LNCS*, pages 556–567. Springer, 2001.

[Kor02] F. Koriche. Approximate coherence-based reasoning. *Journal of Applied Non-classical Logics*, 12:239–258, 2002.

[KR93] H. Kamp and U. Reyle. *From Discourse to Logic: Introduction to Model-theoretic Semantics of Natural Language, Formal Logic and Discourse Representation Theory*. Kluwer, 1993.

[KT99] A. Kakas and F. Toni. Computing argumentation in logic programming. *Journal of Logic and Computation*, 9:515–562, 1999.

[Lev84] H. Levesque. A logic of implicit and explicit belief. In *Proceedings of the 4th National Conference on Artificial Intelligence (AAAI 1984)*, pages 198–202. AAAI Press, 1984.

[LNA+97] R. Loui, J. Norman, J. Alpeter, D. Pinkard, D. Craven, J. Linsday, and M. Foltz. Progress on Room 5: A testbed for public interactive semi-formal legal argumentation. In *Proceedings of the 6th International Conference on Artificial Intelligence and Law (ICAIL 1997)*, pages 207–214. ACM Press, 1997.

[Loz94] E. Lozinskii. Information and evidence in logic systems. *Journal of Experimental and Theoretical Artificial Intelligence*, 6:163–193, 1994.

[Man07] N. Mann. *Logical Argumentation with Generalized Knowledge*. PhD thesis, University College London, 2007.

[MH69] J. McCarthy and P. Hayes. Some philosophical problems from the standpoint of artificial intelligence. In B. Meltzer and D. Michie, editors, *Machine Intelligence 4*, pages 463–502. Edinburgh University Press, 1969.

[MR70] R. Manor and N. Rescher. On inferences from inconsistent information. *Theory and Decision*, 1:179–219, 1970.

[Nut88] D. Nute. Defeasible reasoning and decision support systems. *Decision Support Systems*, 4:97–110, 1988.

[Nut94] D. Nute. Defeasible logics. In *Handbook of Logic in Artificial Intelligence and Logic Programming, Volume 3: Nonmonotonic Reasoning and Uncertainty Reasoning*, pages 355–395. Oxford University Press, 1994.

[Par96] S. Parsons. Defining normative systems for qualitative argumentation. In *Proceedings of the International Conference on Formal and Applied Practical Reasoning*, volume 1085 of *LNCS*, pages 449–463. Springer, 1996.

[Par98] S. Parsons. A proof theoretic approach to qualitative probabilistic reasoning. *International Journal of Approximate Reasoning*, 19:265–297, 1998.

[Par01] S. Parsons. *Qualitative Approaches to Reasoning Under Uncertainty*. MIT Press, 2001.

[Per80] C. Perelman. *Justice, Law, and Argument*. Reidel, 1980.

[Per82] C. Perelman. *The Realm of Rhetoric*. University of Notre Dame, 1982.

[PH98] S. Parsons and A. Hunter. *Applications of Uncertainty Formalisms*, volume 1455 of *LNCS*, chapter A review of uncertainty handling formalisms, pages 8–37. Springer, 1998.

[Pol87] J. Pollock. Defeasible reasoning. *Cognitive Science*, 11:481–518, 1987.

[Pol92] J. Pollock. How to reason defeasibly. *Artificial Intelligence*, 57:1–42, 1992.

[Pol95] J. Pollock. *Cognitive Carpentry*. MIT Press, 1995.

[Poo88] D. Poole. A logical framework for default reasoning. *Artificial Intelligence*, 36:27–47, 1988.

[Pra97] H. Prakken. *Logical Tools for Modelling Legal Argument*. Kluwer, 1997.

[Pra02] H. Prakken. Intuitions and the modelling of defeasible reasoning: Some case studies. In *Proceedings of the 9th International Workshop on Nonmonotonic Reasoning (NMR 2002)*, pages 91–99. IRIT, 2002.

[PS97] H. Prakken and G. Sartor. Argument-based extended logic programming with defeasible priorities. *Journal of Applied Non-classical Logic*, 7:25–75, 1997.

[PSJ98] S. Parsons, C. Sierra, and N. Jennings. Agents that reason and negotiate by arguing. *Journal of Logic and Computation*, 8:261–292, 1998.

[PV02] H. Prakken and G. Vreeswijk. Logical systems for defeasible argumentation. In D. Gabbay, editor, *Handbook of Philosophical Logic*, pages 219–318. Kluwer, 2002.

[PWA03] S. Parsons, M. Wooldridge, and L. Amgoud. Properties and complexity of formal inter-agent dialogues. *Journal of Logic and Computation*, 13(3):347–376, 2003.

[Rea89] S. Read. *Relevant Logic*. Blackwell, 1989.

[Rei80] R. Reiter. A logic for default logic. *Artificial Intelligence*, 13:81–132, 1980.

[RR04] C. Reed and G. Rowe. Araucaria: Software for argument analysis, diagramming and representation. *International Journal of Artificial Intelligence Tools*, 14:961–980, 2004.

[RRJ+03] I. Rahwan, S. Ramchurn, N. Jennings, P. McBurney, S. Parsons, and L. Sonenberg. Argumentation-based negotiation. *Knowledge Engineering Review*, 18:343–375, 2003.

[SC95] M. Schaerf and M. Cadoli. Tractable reasoning via approximation. *Artificial Intelligence*, 74:249–310, 1995.

[Sch98] T. Schaub. *The Automation of Reasoning with Incomplete Information: From Semantic Foundation to Efficient Computation*. Springer, 1998.

[Sho88] Y. Shoham. *Reasoning about Change*. MIT Press, 1988.

[Sil96] J. Sillince. Would electronic argumentation improve your ability to express yourself. In B. Gorayska and J. May, editors, *Cognitive Technology: In Search of a Humane Interface*, pages 375–387. Elsevier, 1996.

[Sil97] J. Sillince. Intelligent argumentation systems: Requirements, models, research agenda, and applications. In A. Kent, editor, *Encyclopedia of Library and Information Science*, pages 176–217. Marcel Dekker, 1997.

[SL92] G. Simari and R. Loui. A mathematical treatment of defeasible reasoning and its implementation. *Artificial Intelligence*, 53:125–157, 1992.

[STT01] F. Sadri, F. Toni, and P. Torroni. Dialogues for negotiation: Agent varieties and dialogue sequences. In *Pre-proceedings of the 8th International Workshop on Agent Theories, Architectures, and Languages (ATAL 2001)*, volume 2333 of *LNCS*, pages 69–84. Springer, 2001.

[Tou58] S. Toulmin. *The Use of Argument*. Cambridge University Press, 1958.

[van80] D. van Dalen. *Logic and Structure*. Springer, 1980.

[Ver99] B. Verheij. Automated argument assistance for lawyers. In *Proceedings of the 7th International Conference on Artificial Intelligence and Law (ICAIL 1999)*, pages 43–52. ACM Press, 1999.

[Ver05] B. Verheij. *Virtual Arguments: On the Design of Argument Assistants for Lawyers and Other Arguers*. Cambridge University Press, 2005.

[vGK87] F. van Eemeren, R. Grootendorst, and T. Kruiger. *Handbook of Argumentation Theory*. Foris, 1987.

[VL90] D. Vermeir and E. Laenens. A fixpoint semantics of ordered logic. *Journal of Logic and Computation*, 1:159–185, 1990.

[VLG98] D. Vermeir, E. Laenens, and P. Geerts. Defeasible logics. In *Handbook of Defeasible Reasoning and Uncertainty Management, Volume 2*. Kluwer, 1998.

[Vre97] G. Vreeswijk. Abstract argumentation systems. *Artificial Intelligence*, 90:225–279, 1997.

[Vre06] G. Vreeswijk. An algorithm to compute minimally grounded defence sets in argument systems. In P. Dunne and T. Bench-Capon, editors, *Computational Models of Argumentation (COMMA 2006)*, pages 109–120. IOS Press, 2006.

[Wal89] D. Walton. *Informal Logic: A Handbook for Critical Argumentation*. Cambridge University Press, 1989.

[WB01] P. Wong and Ph. Besnard. Paraconsistent reasoning as an analytic tool. *Journal of the Interest Group in Propositional Logic*, 9:233–246, 2001.

[Woo01] M. Wooldridge. *An Introduction to Multi-agent Systems*. John Wiley, 2001.

Index