Assumption-based Argumentation Dialogues

Xiuyi Fan

November 2012

Supervised by Francesca Toni

Submitted in part fulfilment of the requirements for the degree of Doctor of Philosophy in Computing of Imperial College London and the Diploma of Imperial College London
Abstract

Formal argumentation based dialogue models have attracted some research interests recently. Within this line of research, we propose a formal model for argumentation-based dialogues between agents, using assumption-based argumentation (ABA). Thus, the dialogues amount to conducting an argumentation process in ABA. The model is given in terms of ABA-specific utterances, debate trees and forests implicitly built during and drawn from dialogues, legal-move functions (amounting to protocols) and outcome functions. Moreover, we investigate the strategic behaviour of agents in dialogues, using strategy-move functions. We instantiate our dialogue model in a range of dialogue types studied in the literature, including information-seeking, inquiry, persuasion, conflict resolution, and discovery. Finally, we prove (1) a formal connection between dialogues and well-known argumentation semantics, and (2) soundness and completeness results for our dialogue models and dialogue strategies used in different dialogue types.
Acknowledgements

I would like to express the deepest appreciation to my supervisor Francesca Toni for her patience, continued support and encouragement during my PhD study. Thanks to Francesca for her advice regarding both the technical training and academic life in general. She has taken a lot of headache in training a student who had little experience in writing mathematical proofs to write a thesis that contains many formal results in three years. Without her guidance and persistent help this thesis would not have been possible.

Thanks to Tom Henderson for his mentoring in the past years. He continually and convincingly conveyed the concept of “discipline and commitment” in research and scholarship. Tom started as my Master’s thesis advisor, then became a sounding board that giving helpful feedback on many naive thoughts I gave him, and my ski coach (with Joelle), poker teacher, .... and, most importantly, a good friend. His kindness and humility is something I try to imitate (albeit not very successfully so far). Without him, I might have long dropped out of graduate school.

Thanks to Susan Eisenbach and Krysia Broda for helping me to get my Rector’s Award scholarship and bursary to study at Imperial, for arranging additional travel grants to support conference trips and for their encouragement and unfailing support in many other occasions. Thanks to Amani El-Kholy, Naranker Dulay, Teresa Ng, Ann Halford and many others for looking after me so well in the department.

Thanks to Bob Kowalski, Henry Prakken, and Sanjay Modgil for their helpful comments and feedback on my work.

Thanks to everyone in my office and many others in the department for helping to create an enjoyable working environment. In particular, I would like to thank Yang Gao for the stimulating discussions. Thanks to Dianhuan Lin for reading an early draft of this thesis and providing helpful feedback. Thanks to Nick Ng for giving me opportunities to present my work at our friendly PhD student seminars.

Finally, thanks to Tony Hunter and Abbas Edalat for taking the time to read this thesis and for examining me.
To my parents and Lisa.
## Contents

1 Introduction ........................................... 12  
  1.1 Overview ........................................... 12  
  1.2 Structure of Thesis ................................. 15  
  1.3 Contributions ...................................... 16  
  1.4 Statement of Originality ........................... 17  
  1.5 Publications ....................................... 17  

2 Background ........................................... 19  
  2.1 Introduction ....................................... 19  
  2.2 Argumentation ...................................... 19  
  2.3 Abstract Argumentation ............................ 22  
  2.4 Assumption-based Argumentation ................... 27  
  2.5 Dialogue ........................................... 32  
  2.6 Conclusion ......................................... 38  

3 Argumentation Dialogues .............................. 40  
  3.1 Introduction ....................................... 40  
  3.2 Generic Dialogues .................................. 40  
  3.3 Related Dialogues .................................. 53  
  3.4 Strategy-move Functions ............................ 54  
  3.5 Related Work ...................................... 55  
  3.6 Conclusion ......................................... 56  

4 Dialogues and Argumentation Semantics .............. 58  
  4.1 Introduction ....................................... 58  
  4.2 Debate Trees ....................................... 58  
  4.3 Focused Dialogues .................................. 69  
  4.3.1 One-way Expansion Dialogue .................... 77  
  4.4 Dialogues and Debate Forests ....................... 79  

5
4.5 Illustration ........................................ 90
4.6 Related Work .................................... 90
4.7 Conclusion ........................................ 91

5 Dialogue Strategies ........................................ 94
5.1 Introduction ..................................... 94
5.2 Preliminaries ..................................... 95
5.3 Classes of Strategy-move Functions .................. 99
5.4 Results ........................................... 100
5.5 Illustration ....................................... 103
5.6 Related Work .................................... 104
5.7 Conclusion ........................................ 104

6 Dialogue Instantiations (I) ......................... 106
6.1 Introduction ..................................... 106
6.2 Information-seeking and Inquiry .................... 106
   6.2.1 Information-seeking Dialogues ............... 107
   6.2.2 Inquiry Dialogues .......................... 109
   6.2.3 Related Work ............................... 111
   6.2.4 Summary .................................... 111
6.3 Persuasion Dialogues with Mechanism Design ........ 111
   6.3.1 Mechanism Design Background ............... 112
   6.3.2 Preliminaries ............................... 113
   6.3.3 Strategy-move Functions used in Persuasion .. 114
   6.3.4 Persuasion Dialogue Results ................. 115
   6.3.5 Related Work ............................... 122
   6.3.6 Summary .................................... 123
6.4 Conclusion ........................................ 124

7 Dialogue Instantiations (II) ......................... 125
7.1 Introduction ..................................... 125
7.2 Preliminaries .................................... 125
7.3 Conflict Resolution Dialogues ..................... 126
   7.3.1 Motivating Example ........................ 126
   7.3.2 Conflict and Conflict Resolution Agents ...... 127
   7.3.3 Conflict Resolution Dialogues and Sequences .. 130
   7.3.4 An Extension ............................... 132
7.3.5 Related Work ........................................ 134
7.3.6 Summary ........................................... 135
7.4 Discovery Dialogues ................................. 135
  7.4.1 Another Movie Example ....................... 136
  7.4.2 Discovery Dialogue Formulation ............. 137
  7.4.3 Debate Trees Revisited ....................... 140
  7.4.4 Related Work ................................... 144
  7.4.5 Summary ....................................... 144
7.5 Conclusion ........................................... 145

8 Conclusion ........................................... 146

Bibliography ............................................ 148
# List of Tables

2.1 Argumentation semantics for Example 1 .......................... 25  
2.2 Argumentation semantics Example 2 .......................... 25  
2.3 Argumentation semantics Example 3 .......................... 26  
2.4 Dialogue types (adapted from [WK95]). .......................... 33  
2.5 Black and Hunter’s inquiry dialogue framework. .......................... 35  
2.6 Prakken’s persuasion dialogue framework. .......................... 37  
3.1 Dialogue $\delta$ for Example 10. .......................... 44  
3.2 A natural language reading of the dialogue in Table 3.1. .......................... 44  
3.3 Summary of concepts introduced in Chapter 3. .......................... 57  
4.1 A dialogue that draws a non-focused debate tree. .......................... 70  
4.2 Another dialogue that draws a non-focused debate tree. .......................... 70  
4.3 A sub-dialogue of the dialogue in Example 31 (part 1). .......................... 86  
4.4 A sub-dialogue of the dialogue in Example 31 (part 2). .......................... 86  
4.5 Summary of types of dialogues defined with legal-move and outcome functions from Chapters 3 and 4. .......................... 92  
4.6 Summary of our dialogue framework. .......................... 92  
5.1 ABA frameworks for Example 33. .......................... 96  
5.2 A dialogue that is constructed with a truthful strategy-move function in Example 33. .......................... 103  
5.3 A dialogue that is constructed with a non-attack thorough strategy-move function in Example 33. .......................... 103  
5.4 Summary of strategy-move functions defined in Chapter 5. .......................... 105  
6.1 Information-seeking and inquiry dialogues (from [WK95]). .......................... 107  
6.2 Two formulations of information-seeking dialogues. .......................... 107  
6.3 Information-seeking dialogue in Example 33. .......................... 109  
6.4 Two formulations of inquiry dialogues. .......................... 110
6.5 An example dialogue constructed with a $\phi_p$ and a $\phi_o$. . . . . . . 115
6.6 Dialogue as mechanism . . . . . . . . . . . . . . . . . . . . . . . . 120
6.7 A persuasion dialogue example. . . . . . . . . . . . . . . . . . . . . . . 121

7.1 Jenny’s and Amy’s Concession and Non-Concession Rules . . . . 129
7.2 A conflict resolution dialogue example (part 1). . . . . . . . . 132
7.3 A conflict resolution dialogue example (part 2). . . . . . . . . 132
7.4 A conflict resolution dialogue example (part 3). . . . . . . . . 133
7.5 A conflict resolution dialogue example (part 4). . . . . . . . . 133
# List of Figures

2.1 Toulmin’s layout of arguments. .................................. 20

2.2 An argumentation graph example. ................................. 23

2.3 Argumentation graph for Example 2. ............................ 25

2.4 Argumentation graph Example 3. ................................. 26

2.5 Arguments in Example 5. ........................................ 30

2.6 An abstract dispute tree for the argument A in Figure 2.2. .... 32

4.1 Nodes in a debate tree connected as in ABA arguments (left) or attacks between ABA arguments (right). .................... 60

4.2 The debate tree drawn from the dialogue in Example 9. ....... 63

4.3 The construction of the debate tree in Figure 4.2. Here the tree is constructed using utterances $u'_{1}, \ldots, u'_{9}$ as given in Example 11. ........................................ 64

4.4 Two proponent arguments (left) and one opponent argument (right) drawn from the debate tree shown in Figure 4.2. .............. 65

4.5 The non-patient debate tree drawn from the dialogue in Example 22. .................................................. 65

4.6 Filtered (left) and not filtered (right) debate trees. .......... 66

4.7 A non-focused tree. Note, in this tree, the opponent argument \{a, b\} ⊨ t s has two proponent nodes as its children: (c, um : P[7]) and (c, um : P[8]). ............................................. 70

4.8 A non-focused tree. Note, in this tree, a proponent node (the root) has two “sets” of proponent children from two different utterances. 70

4.9 The debate tree drawn from the dialogue in Example 26. ........ 72

4.10 The debate tree drawn from the dialogue in Example 19. .... 72

4.11 A non-positive dialogue in Example 28. ........................ 73

4.12 The expanded debate tree of the debate tree shown in Figure 4.10. .................................................. 76

4.13 The (non-focused) debate tree drawn from the (non-focused) dialogue in Example 31. ............................................. 81

4.14 The argument \{s\} ⊨ t s (left) and its attacker (right). ........ 81

4.15 The construction of the debate forest in Example 31 (Part 1). 84
4.16 The construction of the debate forest in Example 31 (Part 2). . . . 85
4.17 The debate tree drawn from the dialogue in Example 10. . . . . . 90

7.1 The construction of the debate tree drawn from the discovery dia-
logue in Example 42. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 143
1 Introduction

1.1 Overview

Argumentation based dialogue systems have attracted some research interests in recent years (e.g. see [Pra06]), largely due to the need for agents to communicate and agree in multi-agent systems. The modern study of formal dialogue systems for argumentation starts from Charles Hamblin’s work [Ham71]. The topic was initially studied within philosophical logic and argumentation theory [Mac90, WK95]. Subsequently, researchers from the field of artificial intelligence & law [Gor94, Pra01] and multi-agent systems [APM00, PWA03] have looked into dialogue systems as well.

Much literature has explained the role agent systems plays in the current computing research (e.g., see [Woo09] for a thorough treatment on this subject). Here, we quote from [JMP05] that gives a nice summary of software agents.

The rise of the Internet, ambient computing, ad-hoc networks and virtual communities have led to a paradigm shift in how we view computer systems and computation. Instead of computer systems being viewed simply as programs which execute some pre-determined method, a better analogy is to view systems as societies of interacting and autonomous entities, or “agents”, who combine together as and when necessary to achieve possibly-conflicting individual objectives. This agent-oriented perspective has become influential within computer science over the last decade, and has made connections with prior work in biology (e.g., ecology, evolutionary theory), physics (statistical mechanics), economics (game theory) and sociology (organisation theory).

Wooldridge and Jennings give the following list of agent properties (quoted verbatim from [WJ95]).
autonomy: agents operate without the direct intervention of humans or others, and have some kind of control over their actions and internal state;

social ability: agents interact with other agents (and possibly humans) via some kind of agent communication language;

reactivity: agents perceive their environment (which may be the physical world, a user via a graphical user interface, a collection of other agents, the Internet, or perhaps all of these combined), and respond in a timely fashion to changes that occur in it;

pro-activeness: agents do not simply act in response to their environment, they are able to exhibit goal-directed behaviour by taking the initiative.

We can think of multi-agent systems as a collection of agents that co-exist in a shared environment. Though it is uncertain if all agents in the environment share some overarching goals or not, each agent certainly has its own goals. Moreover, usually it is the case that agents need to communicate in order to cooperate, coordinate, and negotiate with other agents so as to reach their individual goals. Therefore, it is easy to see the need for effective communication among agents exists.

Much research has been dedicated to low level inter-agent communication, in terms of languages and protocols for autonomous software agents. The most influential example is the development of the FIPA’s Agent Communications Language (FIPA-ACL) [FIP02]. FIPA-ACL relies on speech act theory [Sea70, Aus75] developed by Searle and Austin. It defines a set of performatives, including ask-if, ask-all, tell, deny, insert, etc. It also specifies allowed agent responses upon receiving these performatives. However, the communication framework given by FIPA-ACL, through its performatives, is a low level framework. To capture more sophisticated higher level agent communication behaviours, e.g., inquiry, and information-seeking, dialogue models have been proposed.

Two major questions need to be addressed in a study of dialogue models. Firstly, how to construct “coherent” dialogues? Secondly, how to construct dialogues with specific goals? The first question can be addressed by introducing dialogue protocols; and the second question can be addressed by studying dialogue strategies. This thesis is concerned with answering these two questions. Our goals are to:

- Develop a generic argumentation dialogue framework capable of supporting various types of dialogue.
Study strategic behaviours of agents participating in different types of dialogues.

Hence, we address the high level agent communication need as follows:

**In multi-agent systems, argumentation dialogues are a viable and appropriate means for agents to exchange information and deliberate.**

Specifically, this thesis presents a two-agent argumentation based dialogue model. We use Assumption-based Argumentation (ABA) [DKT09] as the underlying representation for arguments, as ABA is a general purpose, widely used argumentation framework. In ABA, there are strong theoretical results to use, e.g., there are existing work in computing acceptable arguments within an ABA framework with respect to various argumentation semantics [DKT09]. Moreover, there is a clear relation between ABA and Abstract Argumentation (AA) framework, i.e., ABA and AA are instances of each other [DMT07]. Hence all theoretical results obtained in AA are applicable in ABA.

In ABA, we have rules, assumptions, and contraries. Informally, rules and assumptions form deductions (arguments); contraries of assumptions provide a means of specifying “counter-arguments” for arguments composed of rules and assumptions. Within an ABA framework, arguments are deemed “acceptable” if certain properties of these arguments are fulfilled, e.g., an argument does not “attack” itself and attacks all arguments that attack it. In our model, a dialogue is composed of utterances with contents either a rule, an assumption, a contrary, a topic, a goal, or pass. A dialogue starts with an agent posing a topic/goal and completes when both agents utter pass.

To ensure coherence of a dialogue, we introduce a set of legal-move and outcome functions. Legal-move functions are mappings from dialogues to utterances. Hence, given an incomplete dialogue, a legal-move function returns a set of allowed utterances that extends the dialogue. Legal-move functions can then be viewed as dialogue protocols. Outcome functions are mappings from dialogues to true/false. Given a dialogue, an outcome function returns true if a certain property holds within that dialogue.

Through dialogues, the participating agents construct a “joint knowledge base” by pooling all information disclosed in the dialogue to form the *ABA framework, $\mathcal{F}_\delta$, drawn from a dialogue, $\delta$*. Since $\mathcal{F}_\delta$ contains all information that the two
agents have uttered in the dialogue, it gives the context of examining the accept-
ability of the claim of the dialogue. Conceptually, a dialogue is “successful” if its
claim is “acceptable” in $F_\delta$. Rather than checking success retrospectively, once
the dialogue is completed, we define legal-move functions guaranteed to gener-
ate successful dialogues. This main result is obtained by mapping the debate tree
generated from a dialogue to an abstract dispute tree [DKT06].

Some of the earlier studies on dialogue systems have categorised dialogues into
six types: persuasion, negotiation, inquiry, deliberation, information-seeking and
eristics [WK95]. Each of these types of dialogues has its own goals, and agents
participating in different types of dialogue have different interests. Hence different
types of dialogues call for different dialogue strategies.

Building upon the aforementioned dialogue protocols, dialogue strategies can
be formulated via strategy-move functions. These are mappings from dialogues
and legal-move functions to utterances. Hence, given a dialogue and a legal-move
function, a strategy-move function selects a subset of the set of utterances allowed
by the legal-move function such that utterances within this subset advance the
dialogue towards the agents’ and the dialogue’s specific goals.

For instance, in an information-seeking dialogue, a questioner agent poses a
topic and an answerer agent puts forward information that is related to the topic.
The aim of the dialogue is to build proper (arguments) for this topic. The be-
haviours of the questioner and the answerer can be captured by two strategy-
move functions: the pass and the non-attack strategy-move functions, respectively.
Agents (questioners) that use a pass strategy-move function put forward the claim
and no other utterance in a dialogue; agents (answerers) that use a non-attack
strategy-move function only utter rules and assumptions, but not contraries.

To summarise, this thesis studies argumentation based dialogue models from
two angles (1) dialogue protocols, which are used to keep the integrity of dia-
logues and guarantee the acceptability of the claim of a dialogue and (2) dialogue
strategies, which identify appropriate utterances that advance a dialogue towards
its goal and the goals of the agents participating in the dialogue.

1.2 Structure of Thesis

This thesis is structured as follows. Chapter 2 presents background studies on
argumentation frameworks, especially ABA, and argumentation dialogues.

Chapter 3 presents basic notions of our ABA based dialogue model. Core ele-
ments of our model, including legal-move, outcome and strategy-move functions, are introduced.

Chapter 4 discusses our dialogue protocols in detail and links dialogues to argumentation semantics. We present several legal-move functions and prove soundness results of our dialogue model. Basically, we show that for a dialogue that is compatible with certain legal-move functions, the claim of this dialogue is “acceptable” in the ABA framework drawn from the dialogue.

Chapter 5 discusses dialogue strategies that agents use through various dialogues. Several strategy-move functions are presented in detail in this chapter. We prove soundness and completeness results for claims of dialogues.

Chapter 6 instantiates our dialogue framework to model information-seeking and inquiry dialogues. We formally define these two types of dialogues and prove completeness and soundness results with respect to information disclosed throughout dialogues with certain legal-move and strategy-move functions.

Chapter 7 further instantiates and extends our framework to model conflict resolution and discovery dialogues.

We conclude in Chapter 8 with a summary of this thesis and a list of possible future works.

1.3 Contributions

The contributions of the thesis are as follows:

- Presentation of a generic dialogue model (Chapters 3, 4). This sets the foundation for all later formalisations of various types of dialogues.

- Presentation of a catalogue of dialogue strategies for agents (Chapter 5). This gives a way of specifying agent behaviours in dialogues.

- Formalisation of information-seeking and inquiry dialogues (Chapter 6). This illustrates our dialogue model and justifies it by showing various results.

- Formalisation of persuasion dialogue with mechanism design techniques (Chapter 6). This gives an example on how mechanism design techniques can be applied in argumentation based dialogues. Our results include that: under specified conditions, neither the two agents in a persuasion will lie in dialogues.
• Formalisation of discovery and conflict resolution dialogue (Chapter 7). This further justifies our dialogue model by applying it to more types of dialogues.

1.4 Statement of Originality

I declare that this thesis was composed by myself and that the work it presents is my own, except where otherwise stated.

1.5 Publications

The work in this thesis brings together and builds upon work published as follows:

• (Chapter 4) Xiuyi Fan, Francesca Toni, Assumption-based Argumentation Dialogues, *Twenty-Second International Joint Conference on Artificial Intelligence (IJCAI 2011)*, July 16-22, 2011, Barcelona, Spain [FT11a]

• (Chapter 5) Xiuyi Fan, Francesca Toni, Agent Strategies for ABA based Information-seeking and Inquiry Dialogues, *20th European Conference on Artificial Intelligence (ECAI 2012)*, August 27-31, 2012, Montpellier, France [FT12a]

• (Chapter 6) Xiuyi Fan, Francesca Toni, Mechanism Design for Argumentation based Persuasion Dialogues, *Fourth International Conference on Computational Models of Argument (COMMA 2012)*, September 10-12, 2012, Vienna, Austria [FT12c]

• (Chapter 7) Xiuyi Fan, Francesca Toni, Argumentation Dialogues for Two-Agent Conflict Resolution, *Fourth International Conference on Computational Models of Argument (COMMA 2012)*, September 10-12, 2012, Vienna, Austria [FT12b]

• (Chapter 7) Xiuyi Fan, Francesca Toni, A First Step towards Argumentation Dialogues for Discovery, *First International Workshop on the Theory and Applications of Formal Argumentation (TFA-11)*, July 16-17, 2011, Barcelona, Spain [FT11c]
• (Chapter 7) Xiuyi Fan, Francesca Toni, Two-Agent Conflict Resolution with Argumentation Dialogues (Extended Abstract), *Tenth International Conference on Autonomous Agents and Multiagent System (AAMAS 2011)*, May 2-6, 2011, Taipei, China [FT11b]

The work published in the following paper has also contributed to some early ideas that lead to this thesis, even though not explicitly included.

• Xiuyi Fan, Francesca Toni, Adil Hussain, Multi-Agent Conflict Resolution with Assumption-based Argumentation. *Third International Conference on Computational Models of Argument (COMMA 2010)*, 8th-10th September 2010, Desenzano del Garda, Italy [FTH10]
2 Background

2.1 Introduction

In this chapter, we cover the following topics. Firstly, we look at argumentation from a historical perspective and develop a general understanding of what argumentation is about (Section 2.2). We then give some more formal discussion about the Abstract Argumentation (AA) framework (Section 2.3). After that, we move to the Assumption-based Argumentation (ABA) framework, which serves as the underlying framework for the rest of this thesis (Section 2.4). With a good understanding of argumentation, and ABA in particular, we move to argumentation based dialogues (Section 2.5). We review two existing dialogue frameworks. This review helps us to put the research presented in this thesis into context.

2.2 Argumentation

Argumentation is usually traced back to ancient Greek philosophers [RS09]. In particular, it is often attributed to Aristotle, for his work in formal deductive reasoning and rhetoric [Bil87, BH08].

It is interesting to consider some early development of argumentation. Toulmin, while aiming at the study of “twentieth-century epistemology” (Preface to the Updated Edition, 2003 edition, page vii [Tou58]), gives a description of a semi-formal argumentation model. Toulmin has touched upon many topics that have attracted the attention of several researchers later on.

In [Tou58], Toulmin has described that arguments are performed within a certain “field” with respect to a given “modal”. The “field” can be roughly understood as the scope or domain where an argument takes place. We can view this as defining the scope of a particular argumentation framework. The “Modal” can be viewed as an early attempt to understand argumentation semantics, whether we understand argumentation semantics broadly as criteria of classifying arguments into “acceptable” ones or not.
Furthermore, Toulmin has discussed about uncertainty in arguments. Intuitively, while an argument is constructed, the facts used to support the claim might be uncertain; the rules used to deduct the claim from facts might also be uncertain. While an argument is attacked, the attack relation between arguments might also be uncertain. Recent research has attempted to incorporate these uncertainty into their argumentation frameworks, e.g., [Zuk09] gives a few pointers to more recent research in these directions.

Toulmin’s layout of arguments is shown in Figure 2.1. In this figure, D stands for Data, Q stands for Qualifier, C stands for Claim, W stands for Warrant, B stands for Backing, and R stands for Rebuttal [Tou58, Ver09]. Toulmin’s model is well recognised and acknowledged in later works such as [Dun95]. However, apart from mentioning “modal”, Toulmin has made little progress in formally understanding “argument acceptability” or “argumentation semantics” as we know today.

Another line of influential work on early development of argumentation is by Pollock [Pol94, Pol91, Pol87]. Prakken and Horty [PH12] give a nice summary of Pollock’s contribution in computational argumentation. It is considered that Pollock developed one of the first formal systems for argumentation based inference. Many topics in argumentation were first studied by Pollock, such as argument structure, the nature of defeasible reasons, the interplay between deductive and defeasible reasons, rebutting versus undercutting defeat, argument strength, argument labelings, self-defeat, and resource-bounded argumentation [PH12].

In Pollock’s view, defeasible reasoning is the following [Pol95] (Page 85):

Defeasible reasoning is, a fortiori, reasoning. Reasoning proceeds by constructing arguments, where reasons provide the atomic links in arguments. Conclusive reasons logically entail their conclusions. Defeasibility arises from the fact that not all reasons are conclusive.

\[
\text{D} \rightarrow \text{So, Q, C} \\
\text{Since W} \\
\text{On account of B} \quad \text{Unless R}
\]

Figure 2.1: Toulmin’s layout of arguments.
Those that are not are prima facie reasons. Prima facie reasons create a presumption in favour of their conclusion, but it is defeasible.

Pollock considers arguments as inference trees, where the nodes are statements, with the leaf nodes being premises, and the links are applications of “reasons”. Formally, Pollock considers sequences of lines from an argument. Each line from an argument is a tuple \((\varphi, r, l, s)\), where \(\varphi\) is a proposition, \(r\) is the reason applied to infer \(\varphi\), \(l\) is the set of preceding lines from which \(\varphi\) is inferred, and \(s\) is the line’s strength (a number). A sequence of such lines is a (linear) argument if each line is such that its proposition is either inferred from earlier lines or taken from the knowledge base [PH12].

With his definition of arguments, Pollock defines defeat as follows.

- An argument line \((\varphi, r, l, s)\) defeats an argument line \((\varphi', r', l', s')\) if and only if
  - \(r'\) is a defeasible rule, and
  - \(s \geq s'\), and
  - \(\varphi = \neg \varphi'\) or \(\varphi = \neg r'\) (here \(\neg r\) is shorthand for saying that the antecedents of rule \(r\) do not support its consequent).

- An argument \(A\) defeats an argument \(B\) if and only if a line of \(A\) defeats a line of \(B\).

In addition to argument construction, with argument defeat relation defined, Pollock has also worked on various argumentation semantics. For him, a semantics is an account of how the set of constructed arguments, taken together with their defeat relations, determines what a cogniser should believe. His understanding is the same as today’s view on argumentation semantics. Here we do not repeat Pollock’s two semantics definitions, which can be found in [Pol87] and [Pol95], respectively. We just point out that, as proved in [Dun95] and [JV99], under certain conditions, Pollock’s two semantics are equivalent to Dung’s grounded and preferred semantics [Dun95] (both introduced later), respectively.

From a modern perspective, argumentation is a useful technique in computing. Argumentation theories have been in rapid development in the field of artificial intelligence, and logical computing in particular, e.g., Rahwan and McBurney [RM07] say:
if logic is the means by which computers think, then argumentation is the means by which intelligent computers interact, both with one another and with humans.

Hence, argumentation is a means for intelligent computer interaction.

Applications of argumentation in computing are developed in many areas. In the area of medical application, many argumentation based expert systems and tutorial systems have been developed [FGG+07]. Argumentation has long been used in legal reasoning, e.g. Bench-Capon et al [BCPS09] present a detailed overview of argumentation applications developed for legal reasoning in the past two decades. Argumentation has also been used in recommendation systems [CnMS06]. Here, contrary to machine learning based systems, argumentation provides a white-box approach. Users of such systems not only receive the recommendation, but the reasoning behind it. In the context of Semantic Grid and Semantic Web, argumentation is also an emerging technology [TGK+08]. Supported by ontology frameworks, web services, and Grid computing, argumentation is a natural way for agents to interact with one another [RS09, TSL+07, TGC07, MS07].

A few informal definitions for argumentation are presented to set up the ground for our discussion. These definitions are based on the argumentation framework used throughout this thesis, though they represent similar concepts in most other argumentation frameworks as well. Formal definitions follow later.

**Argument:** An argument is a set of propositions composed of three parts: a claim, some premises, and an inference from the premises to the claim.

**Contradiction:** Two formulas are in contradiction, for example, if one negates the other.

**Attack:** One argument, A, attacks another argument, B, if and only if the conclusion of A is a contradiction with a premise of B.

### 2.3 Abstract Argumentation

Abstract argumentation (AA) [Dun95] is developed as one of the first computational argumentation frameworks. An AA framework is simply a pair $AF = \langle A, R \rangle$, consisting of a set of arguments, $A$, and a binary attack relation, $R$. For example,

$AF = \langle \{A, B, C\}, \{(A, B), (B, A), (B, C), (C, C)\} \rangle$
is an argumentation framework. AA is abstract as its arguments and attacks are both atomic. Moreover, there is no other relation between arguments defined in AA, except attacks.

An abstract argumentation system captures “inconsistency” between arguments, given by the relation of attack, hence not all arguments can co-exist with each other. It is essential to determine which arguments are acceptable. Intuitively, arguments that are not attacked should be accepted; arguments that are attacked and cannot somehow withstand their attacks should not be accepted. As we show in the next section, [Dun95, BG09] have proposed a full suite of policies, which are called semantics, to determine the acceptability of arguments in an abstract argumentation framework. These argumentation semantics have been adopted and used in many other argumentation frameworks.

Abstract argumentation is usually represented as directed graphs. As introduced in [Dun95], arguments are represented as nodes and attacks are represented as directed links in these graphs. Figure 2.2, is the directed graph for the earlier argumentation framework

\[
AF = \langle \{A, B, C\}, \{(A, B), (B, A), (B, C), (C, C)\}\rangle
\]

From Figure 2.2, it can be noticed that in an argumentation framework, two arguments can attack each other (arguments A and B) and an argument can attack itself (argument C).

In the literature, extension is used to describe a subset of arguments. An “acceptable” extension, E, is a set of arguments which jointly withstand from attacks according to a given semantics. Formally, given an abstract argumentation framework \(AF = \langle A, R\rangle\) and a semantics \(S\), the set of extensions prescribed by \(S\) for \(AF\) is denoted as \(E_S(AF) \subseteq 2^A\). The acceptability of an argument \(a \in A\) is now defined as its membership of elements of \(E_S(AF)\).

Intuitively, an extension requires all elements in the extension not to attack each other, hence to be conflict-free.

- Formally, given an argumentation framework \(AF = \langle A, R\rangle\), a set \(S \subseteq A\) is conflict-free if and only if \(\nexists a, b \in S\) such that \((a, b) \in R\).
A further requirement of an extension is that a set of arguments can withstand attacks by firing back counterattacks, hence be acceptable.

- Formally, given an argumentation framework \( AF = \langle A, R \rangle \), an argument \( a \in A \) is acceptable with respect to a set \( S \subseteq A \) if and only if \( \forall b \in A : (b, a) \in R \implies (\exists c \in S : (c, b) \in R) \).

A set of arguments is acceptable if and only if all of its elements jointly defend the set by issuing counterattacks.

- Formally, a set \( S \subseteq A \) is acceptable if and only if \( \forall x \in A, \forall b \in S : (x, b) \in R \implies \exists c \in S(c, x) \in R \).

With conflict-freeness and acceptability defined, we can define admissibility in terms of sets that are both conflict-free and acceptable.

- Admissible: Given an argumentation framework \( AF = \langle A, R \rangle \), an extension \( E \) is admissible if and only if \( E \) is conflict-free and acceptable.

The admissible semantics is the most widely used semantics, both in abstract argumentation and other works [RS09]. Admissibility is easy to understand as a set of arguments which do not attack themselves but attack all arguments that attack this set. Several other semantics have been developed such that each of them is believed to have better use in certain applications.

- Complete: Given an argumentation framework \( AF = \langle A, R \rangle \), an extension \( E \) is complete if and only if \( E \) is admissible and every argument of \( A \) which is acceptable with respect to \( E \) belongs to \( E \).

- Grounded: An extension is grounded if and only if it is the smallest set, with respect to set inclusion, of arguments which is complete.

- Preferred: An extension is preferred if and only if it is a maximal set, with respect to set inclusion, of arguments which is admissible.

- Ideal: An extension is ideal if and only if it is a maximal set, with respect to set inclusion, of arguments which is admissible and a subset of all preferred extensions.

- Stable: An extension is stable if and only if it is conflict-free and attacks all arguments that are not in it.
Example 1. In the argumentation framework:

\[ AF' = \langle \{A, B, C\}, \{(A, B), (B, A), (B, C), (C, C)\} \rangle, \]

shown in Figure 2.2, we can obtain extensions as shown in Table 2.1.

<table>
<thead>
<tr>
<th>Semantics</th>
<th>Sets</th>
</tr>
</thead>
<tbody>
<tr>
<td>Conflict-free</td>
<td>{} , {A} , {B}</td>
</tr>
<tr>
<td>Admissible</td>
<td>{} , {A} , {B}</td>
</tr>
<tr>
<td>Complete</td>
<td>{} , {A} , {B}</td>
</tr>
<tr>
<td>Grounded</td>
<td>{}</td>
</tr>
<tr>
<td>Preferred</td>
<td>{A} , {B}</td>
</tr>
<tr>
<td>Ideal</td>
<td>{}</td>
</tr>
<tr>
<td>Stable</td>
<td>{B}</td>
</tr>
</tbody>
</table>

Table 2.1: Argumentation semantics for Example 1

Example 2. In the argumentation framework:

\[ AF = \langle \{A, B, C, D\}, \{(A, B), (B, C), (C, D)\} \rangle, \]

shown in Figure 2.3, we can obtain extensions as shown in Table 2.2.

<table>
<thead>
<tr>
<th>Semantics</th>
<th>Sets</th>
</tr>
</thead>
<tbody>
<tr>
<td>Conflict-free</td>
<td>{} , {A} , {B} , {C} , {D} , {A, C} , {B, D}</td>
</tr>
<tr>
<td>Admissible</td>
<td>{} , {A} , {A, C}</td>
</tr>
<tr>
<td>Complete</td>
<td>{A, C}</td>
</tr>
<tr>
<td>Grounded</td>
<td>{A, C}</td>
</tr>
<tr>
<td>Preferred</td>
<td>{A, C}</td>
</tr>
<tr>
<td>Ideal</td>
<td>{A, C}</td>
</tr>
<tr>
<td>Stable</td>
<td>{A, C}</td>
</tr>
</tbody>
</table>

Table 2.2: Argumentation semantics Example 2

Example 3. In the argumentation framework,

\[ AF = \langle \{A, B, C, D, E\}, \{(A, B), (B, C), (C, A), (D, E), (E, D), (E, E)\} \rangle, \]

shown in Figure 2.4, we can obtain extensions as shown in Table 2.3.
In addition to the extension based approach introduced previously, the *labelling approach* for computing semantics has been introduced [CG09]. As shown in [MC09], when computing extensions in Abstract Argumentation frameworks, many of the argumentation semantics (admissible, complete, grounded, preferred, and stable) can be computed with the labelling approach. The process works as follows (sketch):

- Each argument in an argumentation framework can be either: IN, OUT or UNDEC.

- Given an AA framework \( \langle A, R \rangle \), for \( x, y \in A \), the *labelling requirements* specify the following:
  
  - \( x \) is IN if and only if every \( y \) that attacks \( x \) is OUT.
  
  - \( x \) is OUT if and only if there is at least one \( y \) that attacks \( x \) such that \( y \) is IN.
  
  - \( x \) is UNDEC if and only if there is no \( y \) that attacks \( x \) such that \( y \) is IN, and it is not the case that: for all \( y \), \( y \) attacks \( x \) implies \( y \) is OUT.

Given a labelling \( \mathcal{L} \), \( \text{in}(\mathcal{L}) \) and \( \text{undec}(\mathcal{L}) \) denote the sets of arguments that is labelled IN and UNDEC, respectively. With the above labelling specification, argument labelling can be mapped to extensions, as follows:
• A labelling is **admissible** if and only if all arguments labelled **IN** and **OUT** satisfy the labelling requirements.

• A labelling is **complete** if and only if all arguments labelled **UNDEC** satisfy the labelling requirements.

• Let \( \mathcal{L} \) be a complete labelling, then:
  - \( \mathcal{L} \) is **grounded** if and only if there does not exist a complete labelling \( \mathcal{L}' \) such that \( \text{in}(\mathcal{L}') \subseteq \text{in}(\mathcal{L}) \).
  - \( \mathcal{L} \) is **preferred** if and only if there does not exist a complete labelling \( \mathcal{L}' \) such that \( \text{in}(\mathcal{L}') \supset \text{in}(\mathcal{L}) \).
  - \( \mathcal{L} \) is **stable** if and only if \( \text{undec}(\mathcal{L}) = \{\} \).

With labelling defined, Modgil and Caminada have shown a result that links labelling with extension semantics. Theorem 6.1 in [MC09] states the following:

Let \( AF = \langle A, R \rangle \) be an argumentation framework, and a set of arguments \( E \subseteq A \). For \( S \in \{\text{admissible, complete, grounded, preferred, stable}\} \), \( E \in E_S(AF) \) if and only if there exists an \( S \) labelling \( \mathcal{L} \) with \( \text{in}(\mathcal{L}) = E \).

Modgil and Caminada state that the labelling approach not only is a viable way of computing argumentation semantics, but also is a useful tool for introducing argumentation semantics to people.

### 2.4 Assumption-based Argumentation

At the same time as the initial development of AA, another argumentation framework, Assumption-based Argumentation (ABA) [BDKT97, BTK93, DKT09], had been developed. Unlike attacks in abstract argumentation, which are abstract and atomic, logic deductions supported by assumptions are used to define arguments. The notion of contrary is introduced to facilitate attacks between arguments. The idea of contrary goes beyond negation. For instance, “it is raining” is not only contrary to “it is not raining” but also to “it is sunny.”

In ABA, arguments are trees, where the root of a tree is the **claim** of the arguments; and leaves are either **assumptions** or **facts**. Using the notion of contrary, an argument \( A \) attacks an argument \( B \) if and only if the claim of \( A \) is in contrary with an assumption of \( B \).
Formally, as in [DKT09], an ABA framework is a tuple \( \langle L, R, A, C \rangle \) where

- \( \langle L, R \rangle \) is a deductive system, with a language \( L \) and a set of inference rules \( R \) of the form \( \beta_0 \leftarrow \beta_1, \ldots, \beta_m (m \geq 0) \) with \( \beta_i \in L \), and if \( m > 1 \), then \( \beta_i \neq \beta_j \) for \( i \neq j, 1 \leq i, j \leq m \);
- \( A \subseteq L \) is a (non-empty) set, whose elements are referred to as assumptions;
- \( C \) is a total mapping from \( A \) into \( 2^L \setminus \{\} \), where each \( c \in C(\alpha) \) is a contrary of \( \alpha \).

Given \( \beta_0 \leftarrow \beta_1, \ldots, \beta_m \), \( \beta_0 \) is referred as the head and \( \beta_1, \ldots, \beta_m \) as the body. We will use the following notation: Head\( (\beta_0 \leftarrow \beta_1, \ldots, \beta_m) = \beta_0 \); Body\( (\beta_0 \leftarrow \beta_1, \ldots, \beta_m) = \beta_1, \ldots, \beta_m \). An ABA framework is flat if and only if no assumption is the head of a rule.

Example 4. Given an ABA framework \( \mathcal{A}F_1 = \langle L, R_1, A_1, C_1 \rangle \) as follows:

- \( L_1 = \{a, b, c, p, q, r\} \),
- \( R_1 = \{p \leftarrow a, q \leftarrow b, r \leftarrow c\} \),
- \( A_1 = \{a, b, c\} \),
- \( C_1 \) is: \( C_1(a) = \{q\}, C_1(b) = \{p\}, C_1(c) = \{r, q\} \).

\( \mathcal{A}F_1 \) is flat.

Let \( \mathcal{A}F_2 = \langle L, R_2, A_2, C_2 \rangle \) be:

- \( L_2 = \{a, p, q\} \),
- \( R_2 = \{p \leftarrow a\} \),
- \( A_2 = \{p, a\} \),

\( \overset{1}{\text{Standard ABA [DKT09] does not require } \beta_i \neq \beta_j, \text{ but this can be imposed with no loss of generality. Suppose there are two otherwise identical ABA frameworks, one with rules such that } \beta_i = \beta_j, \text{ one with rules such that } \beta_i \neq \beta_j (\text{all corresponding rules are otherwise identical}), \text{ then, for any sentence } \beta \text{ in the language of the two ABA frameworks, the acceptability of } \beta \text{ with respect to a given semantics (defined later) are the same in both frameworks, e.g., suppose we have two ABA frameworks that are otherwise identical, except one has a rule } p \leftarrow q, q \text{ whereas the others has a rule } p \leftarrow q \text{, we can show that for any sentence } \beta \text{ in the language, the acceptability of } \beta \text{ is the same in both frameworks.}}}{2\text{Here, as in [GT08], we define the contrary of an assumption as a total mapping from an assumption to a set of sentences, instead of a mapping from an assumption to a sentence as in the original ABA [BDKT97].}} \)

28
• $C_2$ is: $C_2(a) = \{q\}$, $C_2(p) = \{q\}$.

$\mathcal{AF}_2$ is not flat as $p$ is both the head of a rule $(p \leftarrow a)$ and an assumption.

ABA frameworks can be defined for any logic specified by means of inference rules. Sentences in the underlying language are assumptions. Arguments are deductions of a claim supported by a set of assumptions. Attacks are directed at the assumptions in inference rules.

Formally, given a deduction system $\langle L, \mathcal{R} \rangle$, a set of rules $\mathcal{R}$, and a set of assumptions $A \subseteq L$, an argument for $c \in L$ (the claim) supported by $A \subseteq A$ is a tree with nodes labelled by sentences in $L$ or by the symbol $\tau^3$, such that

1. the root is labelled by $c$

2. for every node $N$
   • if $N$ is a leaf then $N$ is labelled either by an assumption or by $\tau$;
   • if $N$ is not a leaf and $\beta_0$ is the label of $N$, then there is an inference rule $\beta_0 \leftarrow \beta_1, \ldots, \beta_m (m \geq 0)$ and either $m = 0$ and the child of $N$ is $\tau$ or $m > 0$ and $N$ has $m$ children, labelled by $\beta_1, \ldots, \beta_m$ (respectively)

3. $A$ is the set of all assumptions labelling the leaves.

We use $A \vdash c$ to denote an argument with assumptions $A$ and claim $c$.

With the notion of arguments and contrary of assumption, we can formally define attack in an ABA framework as [DKT09]:

• an argument $A_1 \vdash c_1$ attacks an argument $A_2 \vdash c_2$ if and only if the claim $c_1$ of the first argument is the contrary of one of the assumptions in the support $A_2$ of the second argument ($\exists \alpha \in A_2$ such that $c_1 \in C(\alpha)$);

• a set of arguments $A_\mathcal{A}$ attacks a set of arguments $B_\mathcal{B}$ if an argument in $A_\mathcal{A}$ attacks an argument in $B_\mathcal{B}$;

• a set of assumptions $A_1$ attacks a set of assumptions $A_2$ if and only if an argument supported by a subset of $A_1$ attacks an argument supported by a subset of $A_2$.

$\tau \notin L$ intuitively represents “true”.

29
Example 5. (Continuation of Example 4) In $\mathcal{A}_1$, we have $\{a\} \vdash p$ attacks $\{b\} \vdash q$, $\{b\} \vdash q$ attacks $\{a\} \vdash p$, $\{b\} \vdash q$ attacks $\{c\} \vdash r$, and $\{c\} \vdash r$ attacks $\{c\} \vdash r$. We also say that $\{a\}$ attacks $\{b\}$, $\{b\}$ attacks $\{a\}$, $\{b\}$ attacks $\{c\}$, and $\{c\}$ attacks $\{c\}$.

Figure 2.5 illustrates arguments as trees.

$$\{a\} \vdash p \quad \{b\} \vdash q \quad \{c\} \vdash r$$

\[
\begin{array}{ccc}
\{a\} & p & \{b\} \\
\{a\} & \{b\} & q \\
\{a\} & \{b\} & \{c\} \\
\{a\} & \{b\} & \{c\} & r \\
\{a\} & \{b\} & \{c\} & \{c\} & r
\end{array}
\]

Figure 2.5: Arguments in Example 5.

With argument and attack defined, all semantics introduced in abstract argumentation can be applied in ABA. These semantics can also be defined for assumptions, as follows [BTK93, BDKT97].

Formally, given $\mathcal{F} = \langle \mathcal{L}, \mathcal{R}, \mathcal{A}, \mathcal{C} \rangle$:

- *a set of assumptions is admissible* (in $\mathcal{F}$) if and only if it does not attack itself and it attacks all $A \subseteq \mathcal{A}$ that attack it;

- *a set of assumptions is complete* (in $\mathcal{F}$) if and only if it is admissible and contains all assumptions it *defends*, where a set of assumptions $A$ *defends* another set of assumptions $A'$ if and only if $A$ attacks all sets of assumptions that attack $A'$;

- *a set of assumptions is grounded* (in $\mathcal{F}$) if and only if it is the least set (with respect to $\subseteq$) that is complete;

- *a set of assumptions is preferred* (in $\mathcal{F}$) if and only if it is maximally admissible (with respect to $\subseteq$);

- *a set of assumptions is ideal* (in $\mathcal{F}$) if and only if it is the largest (with respect to $\subseteq$) admissible set contained in all maximally (with respect to $\subseteq$) admissible sets;

We can also define for individual argument and sentence to be acceptable, as follows:
• an argument $A \vdash \beta$ is admissible (complete, grounded, preferred, ideal) (in $\mathcal{F}$) supported by $A' \subseteq A$ if and only if $A \subseteq A'$ and $A'$ is admissible (complete, grounded, preferred, ideal, respectively);

• a sentence is admissible (complete, grounded, preferred, ideal) (in $\mathcal{F}$) if and only if it is the claim of an argument that is admissible (complete, grounded, preferred, ideal, respectively) supported by some $A \subseteq A$.

Furthermore, it can be seen that ABA is an instance of AA [DMT07], as illustrated in Example 6.

**Example 6.** (Continuation of Example 5) In $\mathcal{AF}_1$, we have arguments $\{a\} \vdash p$, $\{b\} \vdash q$ and $\{c\} \vdash r$. If we name these three arguments as $A$, $B$, $C$, respectively, then we have the argument-attack relation presented in Example 1.

Later on, we will use the term *frameworks* to describe tuples of the form $\langle \mathcal{L}, \mathcal{R}, A, C \rangle$ but where $C$ is a mapping from $A$ into $2^\mathcal{L}$. Obviously, ABA frameworks are frameworks but not vice versa.

We will use the abstract dispute trees of [DKT06] to prove some of our results later, where an abstract dispute tree for an argument $A$ is a (possibly infinite) tree $T^a$ ($a$ for abstract):

1. every node of $T^a$ is labelled by an argument and is assigned the status of either a *proponent* (P) node or an *opponent* (O) node, but not both;

2. the root of $T^a$ is a P node labelled by $A$;

3. for every P node $n$ labelled by an argument $B$, and for every argument $C$ that attacks $B$, there exists a child of $n$, which is an O node labelled by $C$;

4. for every O node $n$ labelled by an argument $B$, there exists exactly one child of $n$ which is a P node labelled by an argument which attacks some assumption $\alpha$ in the set supporting $B$. $\alpha$ is said to be the *culprit* in $B$;

5. there are no other nodes in $T^a$ except those given by 1-4 above.

The set of all assumptions in (the support of arguments of) the proponent nodes in $T^a$ is called the *defence set* of $T^a$.

An abstract dispute tree is admissible if and only if no culprit in the argument of an opponent node belongs to the defence set of $T^a$. The defence set of an admissible abstract dispute tree $T^a$ for an argument $A$ is admissible (Theorem 5.1
An abstract dispute tree is grounded if and only if it is admissible and finite. The defence set of a grounded abstract dispute tree $T^a$ for an argument $A$ is grounded.

An abstract dispute tree is ideal if and only if for no opponent node $O$ in the tree there exists an admissible tree with root $O$. The defence set of an ideal abstract dispute tree for an argument is ideal (Theorem 3.4 in [DMT07]).

An abstract dispute tree example is shown in Figure 2.6. We can see that this tree is admissible, but not ideal.

![Figure 2.6: An abstract dispute tree for the argument $A$ in Figure 2.2.](image)

### 2.5 Dialogue

In this thesis, we consider that a dialogue is composed of a sequence of utterances made by the participating agents. In [Car83], the author nicely summarises the use of dialogue systems:

[In agent systems] logic is used to define the condition under which a proposition is true and dialogue systems define the conditions under which an utterance is appropriate.

A well recognised dialogue taxonomy is given in [WK95]. Walton and Krabbe divide dialogues into six different types (see Table 2.4, adapted from [WK95]). Later parts of this thesis study information-seeking, inquiry, and persuasion dialogues in some depth. For these three dialogue types, Walton and Krabbe state [WK95]:

The information-seeking dialogue arises from an initial situation where one participant has some knowledge, or is in a position to know something, and the other party both lacks and needs that information. Thus
Table 2.4: Dialogue types (adapted from [WK95]).

<table>
<thead>
<tr>
<th>Type</th>
<th>Initial Situation</th>
<th>Main Goal</th>
<th>Participants’ Aims</th>
</tr>
</thead>
<tbody>
<tr>
<td>Persuasion</td>
<td>Conflicting Point of View</td>
<td>Resolution of Such Conflicts by Verbal Means</td>
<td>Persuade the Other(s)</td>
</tr>
<tr>
<td>Negotiation</td>
<td>Conflict of Interests &amp; Need</td>
<td>Making a Deal</td>
<td>Get the Best out of it for Oneself</td>
</tr>
<tr>
<td>Inquiry</td>
<td>General Ignorance</td>
<td>Growth of Knowledge &amp; Agreement</td>
<td>Find a “Proof” or Destroy One</td>
</tr>
<tr>
<td>Deliberation</td>
<td>Need for Action</td>
<td>Reach a Decision</td>
<td>Influence Outcome</td>
</tr>
<tr>
<td>Information-Seeking</td>
<td>Personal Ignorance</td>
<td>Spreading Knowledge &amp; Revealing Positions</td>
<td>Gain, Pass on, Show or Hide Personal Knowledge</td>
</tr>
<tr>
<td>Eristics</td>
<td>Conflict &amp; Antagonism</td>
<td>Reaching a (Provisional) Accommodation in a Relationship</td>
<td>Strike the Others Party &amp; Win in the Onlookers</td>
</tr>
</tbody>
</table>

the initial situation for information-seeking dialogue is asymmetrical.
The goal is some kind of spreading of knowledge.
The inquiry is a type of dialogue that strives to establish or “prove” propositions in order to answer a question (solve a problem) in such a way that a stable and general agreement on the matter at issue results.
The Persuasion dialogue (critical discussion) always arises from a conflict of opinions, and its goal is to resolve the conflict, to arrive at a final outcome of stable agreement in the end.

Walton and Krabbe’s taxonomy gives a summary of dialogue goals and participants’ aims for different type of dialogues. Dialogue systems provide a computational means of realising these goals and aims.

From the computational perspective, dialogue systems has been studied, e.g. in [Pra06]. In such systems, two or more participants exchange statements for a purpose, e.g., seeking information, resolving a difference of opinion, etc. Computationally, a complete dialogue system usually includes the following elements [Lou98]:

33
• Contents of utterances during the dialogue.

• Occasions of making any utterance (conditions on when an utterance can be made).

• Consequences of receiving an utterance.

• Mechanisms of determining the outcome of a dialogue.

The list above gives the principle elements of dialogue frameworks. An important factor is that an utterance is appropriate if it furthers the goal of the dialogue in which it is made. Carlson [Car83] has proposed a game-theoretic approach to dialogues, in which speech acts are viewed as moves in a game and rules for their appropriateness are formulated as rules of the game. Such treatment has since been the standard approach of dialogue systems [Pra06].

Prakken has further formalised dialogue systems [Pra06]. He proposes to analyse dialogue systems with a number of elements as follows.

• Dialogue goal: purpose of the dialogue.

• Topic language: the language that describes the content of each utterance.

• Communication language: the language that facilitates the discussion.

• Context: the background and the scope of the dialogue.

• Protocol: the rules defining the legal moves in a dialogue.

• Effect rules: the rules that define the effect of an utterance.

• Outcome rules: the rules that define the outcome of the dialogue.

This list is general enough that many dialogue systems follow suit. Chapter 3 and 4 of this thesis are dedicate to the presentation of our ABA based dialogue system. Though we do not explicitly follow this list while defining our terms, much of our effort can be mapped to this list.

Black and Hunter [BH09] present a formal dialogue system for inquiry dialogues using Defeasible Logic Programming (DeLP) [GS04] as its underlying argumentation framework. Using Prakken’s framework as a guideline, we can view that dialogue system as shown in Table 2.5.
Black and Hunter have considered two types of inquiry dialogues in [BH09], argument inquiry and warrant inquiry. In both dialogues, the two participating agents jointly inquire about topics. Argument inquiry dialogues allow two agents to share knowledge to jointly construct arguments (roughly speaking, in their frameworks, arguments are deductions of topics); whereas warrant dialogues allow two agents to share knowledge to jointly construct dialectical trees. The dialectical tree defined by Black and Hunter resemble the abstract dispute trees we have presented earlier in that “the dialectical tree can be considered a tree with an argument at each node in which a child node is a counter argument to its parent”.

Black and Hunter have defined their framework in steps. Firstly, they define that their framework contains three moves (type of utterances): open, close and assert. Then, they define their dialogues as sequences of moves satisfying three requirements: (1) starting with an open move, (2) every move must be made to a participant of the dialogue, and (3) maintaining a strict interleaving between the two agents. A dialogue terminates when both agents make the close move. A dialogue proceeds by agents contributing sentences as the content of their moves. In argument inquiry dialogues, only sentences that “expand” the deduction of the topic are allowed in the dialogue; whereas in warrant dialogues sentences that are both expanding and in conflict with the topic can be uttered. In both cases, there is a “relevance” requirement such that all utterances are “related” to the claim in both dialogues.

Black and Hunter have used the “union” of the two agents’ knowledge as the benchmark for evaluation. They have proved that their framework is both sound and complete, i.e., for their argument inquiry dialogues, if there is an argument in the union of the two agents’ knowledge base, then such argument can be constructed in their dialogues, and if their dialogue finds an argument, then this argu-
ment exists in the union of the two agents’ knowledge base. Similar results have been shown for their warrant inquiry dialogue as well. These results hold mostly because of the exhaustive strategy they enforce, which defines that during a dialogue, if the protocol allows making an utterance, then the agent has to make such utterance, i.e., agents cannot hide any information that is “relevant” to the topic in discussion.

Compare with their work, the work presented in this thesis have a few advantages. Namely,

1. by using a structured argumentation framework, i.e., ABA, we allow dialogues to be computed with different argumentation semantics, e.g., admissibility and grounded, whereas their system uses DeLP and does not support this computation;

2. our framework is generic and supports many types of dialogues whereas their system is only used for inquiry.

Prakken [Pra05] defines a formal dialogue system for persuasion dialogues. Properties of that system are summarised in Table 2.6. Similarly to Table 2.5, much of the detail of Prakken’s system could not be captured in a compact table. However, it can be seen that these two systems serve as examples of dialogue systems and they share the common framework described in [Pra06].

Prakken has introduced two notions of dialogues in [Pra05], liberal and relevant. Both liberal and relevant dialogues are supported with moves: claim, why, argue, concede and retract. Prakken’s liberal dialogues are somewhat similar to Black and Hunter’s dialogues, in that the dialogue starts with a topic, and all moves need to be related to some earlier moves. In Prakken’s framework, an argument is a deduction with a conclusion (conc) and premises (prem). Prakken defines extends as “An argument B extends an argument A if \( \text{conc}(B) = \varphi \) and \( \varphi \in \text{prem}(A) \) (for example, \( r \) since \( s \) extends \( p \) since \( q, r \)).” Unlike Black and Hunter, Prakken requires no explicit interleaving between the two agents and he has specifically defined a turn-taking function to decide which agent makes the next move. Another difference between the two frameworks is that Prakken’s system does not exploit internal structures of arguments. Therefore moves are operated at the level of arguments rather than at the level of logic sentences. This is less of a concern for Prakken as he has focused on persuasion dialogues and there is little need for the two agents to jointly construct arguments in persuasion.
Also similarly to Black and Hunter, Prakken represents his dialogues as trees. Nodes in trees are labelled proponent and opponent meaning they support and attack the topic, respectively. To determining the outcome of a dialogue, Prakken has used an in/out labelling. Conceptually, a node is in if it withstands its attacks; otherwise, it is out. Since the root of a dialogue tree is the topic, the proponent wins the dialogue if the root node is in.

Prakken’s relevant dialogue is his liberal dialogue with certain moves disallowed under specific conditions. Namely, moves that do not change the acceptability status of the topic (root) node is disallowed in relevant dialogues. This refinement ensures more compact and efficient dialogues.

Compare with Prakken’s work, the work presented in this thesis differs such that,

1. our framework is generic and supports many types of dialogues whereas his system is used for persuasion;

2. our dialogue protocol does not pre-assign the proponent or opponent roles to agents and allows agents to construct arguments jointly whereas his system assigns the roles to agents and does not allow joint construction of arguments.

Parsons et al. [PWA03] have presented a study on two-agent information-seeking, inquiry, and persuasion dialogues. This research is performed in an agent-centric view. Though it defines a few utterances, e.g., question, assert, challenge, etc., their definition of dialogue protocol specifies agent behaviours in a global view, not unlike a two-agent interaction algorithms, as shown in Example 7.

**Example 7.** This example presents the information-seeking dialogue protocol shown in [PWA03].
1. \( a_1 \) asks \textit{question}(p).

2. \( a_2 \) replies with either \textit{assert}(p) or \textit{assert}(\neg p) if it can, and \textit{assert}(U) if it cannot. Which response is given will depend upon the contents of its knowledge base and its assertion attitude. \( U \) indicates that, for whatever reason, \( a_2 \) cannot give an answer.

3. \( a_1 \) either \textit{accepts} \( a_2 \)'s response, if its acceptance attitude allows, or \textit{challenges}. \( U \) cannot be \textit{challenged} and as soon as it is asserted, the dialogue terminates without the question being resolved.

4. \( a_2 \) replies to a \textit{challenge} with an \textit{assert}(S), where \( S \) is the support of an argument for the last proposition challenged by \( a_1 \).

5. Go to 3 for each proposition in \( S \) in turn.

6. \( a_1 \) accepts \( p \) if its acceptance attitude allows.

Compare to our work, Parsons's results are not linked to any argumentation semantics and honesty of agents has been unconditionally assumed, whereas ours supports semantics and does not assume honesty in the level of dialogue protocol. Since their protocols are given as algorithms, it is feasible to compute the complexity of their algorithms, which they have studied. Hence, though [PWA03] have touched upon some aspects of agent dialogues, their research differs from that of Prakken or Black and Hunter and is orthogonal to ours.

Applications of dialogue systems are abundant. Some of the early work in dialogue systems include [Mac90, WW78, WK95]. Using dialogue systems in supporting resource-bounded reasoning is seen in [Lou98, Ger00]. In artificial intelligence and law, persuasion dialogue systems that promote fairness and effective procedure in legal reasoning have been studied [Gor94, HLL94, BC98, Pra01]. In the field of multi-agent systems, dialogue systems have been introduced to support agent interaction [KSE98, PSJ98, APM00, PWA03].

### 2.6 Conclusion

In this chapter, we briefly introduce the origin of argumentation, applications of computer based argumentation and basic argumentation concepts. A large portion of this chapter is devoted to introducing two well-developed argumentation
frameworks: abstract argumentation and ABA. We have also presented some background on argumentation dialogues and introduced two dialogue frameworks.

In the next chapter, we will start to introduce our dialogue model, key ingredients such as legal-move functions, strategy-move functions, and outcome functions will be presented. We will also compare our model with the two main dialogue frameworks presented in this chapter.
3 Argumentation Dialogues

3.1 Introduction

This chapter presents the basis of our model of argumentation based dialogues between (two) agents, using ABA. In our dialogue model, agents can utter claims (to be debated), rules, assumptions and contraries or pass. Thus, dialogues “build” an ABA framework shared between the agents.

The model is given in terms of (various kinds of) legal-move functions, to determine which moves agents can make during dialogues, and outcome functions, to determine whether dialogues have been successful. While participating in dialogues, agents use strategy-move functions to select utterances that fulfil their goals and/or the goal of the dialogue, amongst utterances allowed by legal-move functions.

We illustrate our dialogue framework with an example dialogue using elements from the classic movie *Twelve Angry Men* throughout this chapter (and use this example till Chapter 6).

The chapter is organised as follows. Section 3.2 presents the dialogue model, including legal-move functions and outcome functions. Section 3.3 presents a refined type of dialogue where utterances are related. Section 3.4 introduces strategy-move functions. Section 3.5 discusses a few related work. Section 3.6 concludes.

3.2 Generic Dialogues

We define dialogues as sequences of utterances between two agents $a_1$ and $a_2$ sharing a common language $L$. We also assume there is a (non-empty, possibly infinite) set $ID$ that:

- is totally ordered, with the ordering described by the symbol “$<$”;
- contains a special element $ID_0$ which is the least element with respect to $<$. 


Example 8. For instance, we can define $\mathcal{ID}$ as the set of nonnegative integers, namely, $\mathbb{N}$. The total order relation $<$ is defined in the classic sense that for $a, b \in \mathbb{N}$, $a < b$ if and only if there exists some $c \in \mathbb{N}$ such that $a + c = b$. $ID_0$ is 0.

Utterances are defined as follows:

**Definition 1.** An utterance from agent $a_i$ to agent $a_j$ ($i, j = 1, 2, i \neq j$) is a tuple $\langle a_i, a_j, T, C, ID \rangle$ where:

- $C$ (the content) is of one of the following forms:
  - $claim(\chi)$ for some $\chi \in \mathcal{L}$ (a claim),
  - $rl(s_0 \leftarrow s_1, \ldots, s_m)$ for some $s_0, \ldots, s_m \in \mathcal{L}$ with $m \geq 0$ (a rule),
  - $asm(\alpha)$ for some $\alpha \in \mathcal{L}$ (an assumption),
  - $ctr(\alpha, \beta)$ for some $\alpha, \beta \in \mathcal{L}$ (a contrary),
  - a pass sentence $\pi$, such that $\pi \notin \mathcal{L}$.

- $ID \in \mathcal{ID} \setminus \{ID_0\}$ (the identifier)

- $T \in \mathcal{ID}$ (the target); we impose that $T < ID$.

We refer to an utterance with content $\pi$ as a pass-utterance. We also use $PASS$ to denote the set of all utterances of the form $\langle \_\_\_, \_\_\_, \pi, \_\_\_ \rangle$\(^1\). We refer to an utterance with content other than $\pi$ or $claim(\chi)$ as a regular-utterance. In the remainder $\mathcal{U}$ will stand for the set of all possible utterances as in Definition 1. We will also adopt the following notation: $\mathcal{U}^i$ stands for all utterances from $a_i$ in $\mathcal{U}$, namely of the form $\langle a_i, \_\_\_, \_\_\_ \rangle$, and, for any utterance $\langle a_i, \_\_\_, \_\_\_ \rangle = u$, we also say that $u$ is made by $a_i$.

Intuitively, when the content of an utterance is $\pi$, the utterance indicates that the agent making it does not have or want to contribute any information (i.e. claim, rule, assumption, contrary) that can be added to the dialogue, either because no such information is in the agent’s possession or because the agent chooses not to disclose such information. The target of an utterance is the identifier of some earlier utterance in the dialogue, as we will see below.

Note that until Chapter 7, we will assume that the identifier of an utterance in all of our dialogues, for simplicity, is a non-negative integer, with $\mathcal{ID} = \mathbb{N}$ and $ID_0 = 0$, as seen in Example 8.

\(^1\)Throughout, $\_\_$ stands for an anonymous variable as in Prolog.
Definition 2. A dialogue \(D_{a_i a_j}^\chi(\chi)\) (between agents \(a_i\) and \(a_j\), \(i, j \in \{1, 2\}, i \neq j\) for \(\chi \in \mathcal{L}\)), is a finite sequence \(\langle u_1, \ldots, u_n \rangle\), \(n \geq 0\), where each \(u_l, l = 1, \ldots, n\), is an utterance from \(a_i\) or \(a_j\), \(u_1\) is an utterance from \(a_i\), and:

1. \(u_1 = \langle a_i, a_j, ID_0, claim(\chi), id \rangle\), for some \(id \in ID\);
2. the content of \(u_l\) is \(claim(\chi)\) if and only if \(l = 1\);
3. the target of pass- and claim utterances is \(ID_0\); the target of regular-utterance is not \(ID_0\);
4. for every utterance \(u_i = \langle \_, \_, T, \_, \_ \rangle\), such that \(i > 1\) and \(T \neq ID_0\), there exists a \(u_k = \langle \_, \_, C, T \rangle\), such that \(C \neq \pi\) and \(k < i\);
5. no two consecutive utterances are pass-utterances, other than possibly the last two utterances, \(u_{n-1}\) and \(u_n\);
6. for \(0 \leq i, j \leq n\), if \(i < j\), then the identifier of \(u_i\) is less than the identifier of \(u_j\).

If the last two utterances are pass-utterances, then \(D_{a_i a_j}^\chi(\chi)\) is referred to as completed. If \(n = 0\), \(D_{a_i a_j}^\chi(\chi)\) is referred to as empty. Given a dialogue \(\delta = \langle u_1, \ldots, u_n \rangle\) and an utterance \(u\), we define \(\delta \circ u = \langle u_1, \ldots, u_n, u \rangle\).

Given \(\delta = \langle u_1, \ldots, u_n \rangle\), we say that each \(u_i, 1 \leq i \leq n\), is in \(\delta\).

In the remainder \(D\) will stands for the set of all dialogues as in Definition 2.

Note that condition 4 in Definition 2 enforces a basic notion of “relevance”, expressed by the following.

Definition 3. For any two utterances \(u_i, u_j \in U, u_i \neq u_j, u_j\) is related to \(u_i\) if and only if \(u_i = \langle \_, \_, \_, ID \rangle, u_j = \langle \_, \_, ID, \_ \rangle\).

The following example illustrates our notion of dialogue.

Example 9. Given \(\mathcal{L} = \{s, a, b, c, d, g, q, r\}\), and \(ID = N\) with \(ID_0 = 0\), a possible (completed) dialogue \(D_{a_1 a_2}^s(s)\) is as follows:
Note that in this example (and in general) $a_1$, $a_2$ may or may not be equipped with an ABA framework. ABA is used as a lingua franca in the spirit of the Argument Interchange Format (AIF) [RR09], in the sense that internally, agents can represent their knowledge in representations other than ABA; while communicating, agents convert their internal representation into ABA. Even if $a_1$, $a_2$ are equipped with ABA frameworks, the agents may or may not be truthful, in that, for example, an agent may utter “made-up” rules which do not exist in its ABA framework.

We will illustrate the dialogue notions in the context of the following example, adapted from the movie *Twelve Angry Men*, an example of argumentation based collaborative reasoning [AB11]. Here, we focus on the reasoning of two of the jurors: juror 8, played by Henry Fonda ($a_1$), and juror 9, played by Joseph Sweeney ($a_2$). These agents need to decide whether to condemn a boy, accused of murder, or acquit him, after a trial where two witnesses have provided evidence against the boy. According to the law, the jurors should acquit the boy if they do not believe that the trial has proven him guilty convincingly.

**Example 10.** In this example, two agents perform a dialogue $\delta$ to decide whether to condemn the boy. The example dialogue, $\delta = D_{a_2}^{a_1}(\text{boy innocent})$, is shown in Table 3.1. Utterances in this dialogue should be self-explanatory. For example,

$$\text{boy innocent} \leftarrow \text{boy not proven guilty}$$

says that the boy should be deemed to be innocent if he cannot be proven guilty.

In this example, it can be seen that $a_1$ starts the dialogue by putting forward the claim: $\text{claim(boy innocent)}$. Then both agents contribute rules, assumptions and contraries for and against the claim (directly or indirectly). A natural language reading of this dialogue is in Table 3.2. Utterances in this dialogue are (top-down) related and the dialogue completes with two consecutive passes.

Our dialogue model allows pass-utterances being uttered at any moment throughout a dialogue, but we are able to extract a “sub-dialogue” which contains no pass-

<table>
<thead>
<tr>
<th>$a_1$</th>
<th>$a_2$</th>
</tr>
</thead>
<tbody>
<tr>
<td>\langle a_1, a_2, 0, \text{claim(s)}, 1 \rangle</td>
<td>\langle a_2, a_1, 0, \pi, 2 \rangle</td>
</tr>
<tr>
<td>\langle a_1, a_2, 1, rl(s \leftarrow a), 3 \rangle</td>
<td>\langle a_2, a_1, 3, asm(a), 4 \rangle</td>
</tr>
<tr>
<td>\langle a_1, a_2, 4, ctr(a, q), 5 \rangle</td>
<td>\langle a_2, a_1, 5, rl(q \leftarrow b), 6 \rangle</td>
</tr>
<tr>
<td>\langle a_1, a_2, 6, asm(b), 7 \rangle</td>
<td>\langle a_2, a_1, 7, ctr(b, c), 8 \rangle</td>
</tr>
<tr>
<td>\langle a_1, a_2, 8, asm(c), 9 \rangle</td>
<td>\langle a_2, a_1, 9, ctr(c, r), 10 \rangle</td>
</tr>
<tr>
<td>\langle a_1, a_2, 0, \pi, 11 \rangle</td>
<td>\langle a_2, a_1, 0, \pi, 12 \rangle</td>
</tr>
</tbody>
</table>
Table 3.1: Dialogue $\delta$ for Example 10.

\[
\begin{align*}
&\langle a_1, a_2, 0, \text{claim(boy\_innocent)}, 1 \rangle \\
&\langle a_2, a_1, 1, rl(\text{boy\_innocent} \leftarrow \text{boy\_not\_proven\_guilty}), 2 \rangle \\
&\langle a_1, a_2, 2, asm(\text{boy\_not\_proven\_guilty}), 3 \rangle \\
&\langle a_2, a_1, 3, ctr(\text{boy\_not\_proven\_guilty}, \text{boy\_proven\_guilty}), 4 \rangle \\
&\langle a_1, a_2, 4, rl(\text{boy\_proven\_guilty} \leftarrow w_1\_is\_believable), 5 \rangle \\
&\langle a_2, a_1, 5, asm(w_1\_is\_believable), 6 \rangle \\
&\langle a_1, a_2, 6, ctr(w_1\_is\_believable, w_1\_not\_believable), 7 \rangle \\
&\langle a_1, a_2, 7, rl(w_1\_not\_believable \leftarrow w_1\_contradicted\_by\_w_2), 8 \rangle \\
&\langle a_1, a_2, 8, rl(w_1\_contradicted\_by\_w_2), 9 \rangle \\
&\langle a_2, a_1, 4, rl(\text{boy\_proven\_guilty} \leftarrow w_2\_is\_believable), 10 \rangle \\
&\langle a_1, a_2, 10, asm(w_2\_is\_believable), 11 \rangle \\
&\langle a_2, a_1, 11, ctr(w_2\_is\_believable, w_2\_not\_believable), 12 \rangle \\
&\langle a_1, a_2, 12, rl(w_2\_not\_believable \leftarrow w_2\_has\_poor\_eyesight), 13 \rangle \\
&\langle a_2, a_1, 13, rl(w_2\_has\_poor\_eyesight), 14 \rangle \\
&\langle a_1, a_2, 0, \pi, 15 \rangle \\
&\langle a_2, a_1, 0, \pi, 16 \rangle
\end{align*}
\]

Table 3.2: A natural language reading of the dialogue in Table 3.1.

<table>
<thead>
<tr>
<th>$a_1$</th>
<th>$a_2$</th>
<th>$a_1$</th>
<th>$a_2$</th>
</tr>
</thead>
<tbody>
<tr>
<td>OK</td>
<td>OK</td>
<td>The boy is innocent.</td>
<td>The boy is innocent if he is not proven guilty.</td>
</tr>
<tr>
<td>OK</td>
<td>OK</td>
<td>We assume the boy is not proven guilty.</td>
<td>We assume the boy is innocent if he is not proven guilty.</td>
</tr>
<tr>
<td>OK</td>
<td>OK</td>
<td>The boy cannot be proven not guilty if he is guilty.</td>
<td>The boy is innocent if he is not proven guilty.</td>
</tr>
<tr>
<td>OK</td>
<td>OK</td>
<td>We assume the boy is guilty if witness 1 is believable.</td>
<td>We assume the boy is innocent if he is not proven guilty.</td>
</tr>
<tr>
<td>OK</td>
<td>OK</td>
<td>Witness 1 is not believable if it cannot be believed.</td>
<td>Witness 1 is not believable if it cannot be believed.</td>
</tr>
<tr>
<td>OK</td>
<td>OK</td>
<td>Witness 1 cannot be believed if it is contradicted by witness 2.</td>
<td>Witness 1 cannot be believed if it is contradicted by witness 2.</td>
</tr>
<tr>
<td>OK</td>
<td>OK</td>
<td>Witness 1 is indeed contradicted by witness 2.</td>
<td>Witness 1 is indeed contradicted by witness 2.</td>
</tr>
<tr>
<td>OK</td>
<td>OK</td>
<td>The boy is guilty if witness 2 is believable.</td>
<td>The boy is guilty if witness 2 is believable.</td>
</tr>
<tr>
<td>OK</td>
<td>OK</td>
<td>We assume the witness 2 is believable.</td>
<td>We assume the witness 2 is believable.</td>
</tr>
<tr>
<td>OK</td>
<td>OK</td>
<td>Witness 2 is not believable if it cannot be believed.</td>
<td>Witness 2 is not believable if it cannot be believed.</td>
</tr>
<tr>
<td>OK</td>
<td>OK</td>
<td>Witness 2 cannot be believed as it has a poor eyesight.</td>
<td>Witness 2 cannot be believed as it has a poor eyesight.</td>
</tr>
<tr>
<td>OK</td>
<td>OK</td>
<td>Witness 2 indeed has a poor eyesight.</td>
<td>Witness 2 indeed has a poor eyesight.</td>
</tr>
<tr>
<td>OK</td>
<td>OK</td>
<td>OK.</td>
<td>OK.</td>
</tr>
</tbody>
</table>
utterance. This is defined using the notion of \( \pi \)-pruned sequence obtained from a dialogue, consisting of all regular-utterances in a dialogue. Note that, since no regular-utterance has a pass-utterance as its target (by condition 3 in Definition 2), the target of every utterance in a \( \pi \)-pruned sequence is guaranteed to be in this sequence. Also, since the first utterance in a dialogue can never be a pass utterance (by condition 2 of Definition 2), the first utterance in a \( \pi \)-pruned sequence is always the same as the first utterance in the original dialogue. Moreover, it is trivially true that for any utterance \( u \) that if \( u \) is not in a dialogue \( \delta \), \( u \) is not in the \( \pi \)-pruned sequence of \( \delta \).

**Example 11.** The \( \pi \)-pruned sequence obtained from the dialogue \( \delta \) in Example 9 is the following \( \delta' \):

\[
\begin{array}{c|c}
\hline
a_1 & a_2 \\
\hline
\langle a_1, a_2, 0, \text{claim}(s), 1 \rangle & \langle a_2, a_1, 3, \text{asm}(a), 4 \rangle \\
\langle a_1, a_2, 1, rl(s \leftarrow a), 3 \rangle & \langle a_2, a_1, 5, rl(q \leftarrow b), 6 \rangle \\
\langle a_1, a_2, 4, ctr(a, q), 5 \rangle & \langle a_2, a_1, 7, ctr(b, c), 8 \rangle \\
\langle a_1, a_2, 6, \text{asm}(b), 7 \rangle & \langle a_2, a_1, 9, ctr(c, r), 10 \rangle \\
\langle a_1, a_2, 8, \text{asm}(c), 9 \rangle & \\
\hline
\end{array}
\]

Note that \( \delta' = (u_1', \ldots, u_9') \) where \( u_1' = u_1, u_2' = u_3, u_3' = u_4, u_4' = u_5, u_5' = u_6, u_6' = u_7, u_7' = u_8, u_8' = u_{10}, \) for \( u_i = \langle \ldots, \ldots, i \rangle \) in the original \( \delta \).

In Example 9, we see the two agents taking turns in making utterances. A strict interleaving is enforced between these two agents. In general, such requirement is unnecessary, i.e., an agent is allowed to make a few consecutive utterances before the other agent makes any. We define a (generic) turn-making function to specify the agents that makes the next utterance.

**Definition 4.** A turn-making function \( \gamma : \mathcal{D} \mapsto \{a_1, a_2 \} \) such that, given \( \delta = \mathcal{D}^{n_{a_j}}(\chi) = \langle u_1, \ldots, u_n \rangle, i, j \in \{1, 2\}, i \neq j \):\[
\gamma(\delta) = \begin{cases} 
a_i & \text{if } n = 0, \\
a_x \in \{a_i, a_j \} & \text{otherwise.}
\end{cases}
\]

A dialogue \( \langle u_1, \ldots, u_n \rangle (n > 0) \) is compatible with a turn-making function \( \gamma \) if and only if for each \( l = 1, \ldots, n \), if \( u_l = \langle a_x, \ldots, \ldots \rangle \) then \( \gamma(\langle u_1, \ldots, u_{l-1} \rangle) = a_x \).

Our definition of turn-making function is very liberal, in that it states that \( a_i \) starts \( \mathcal{D}^{n_{a_j}}(\chi) \) and all subsequent utterances are made by any of the agents as dictated by \( \gamma \). As observed earlier, \( \gamma(\delta) = \mathcal{D}^{n_{a_j}}(\chi) \) in Example 9 forces a strict
interleaving between $a_1$ and $a_2$. The following example shows a different kind of $\gamma$.

**Example 12.** Given $L, ID, ID_0$ as in Example 9, a possible (completed) dialogue $D_{a_2}^a(s)$ is as follows:

<table>
<thead>
<tr>
<th>$a_1$</th>
<th>$a_2$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\langle a_1, a_2, 0, claim(s), 1 \rangle$</td>
<td>$\langle a_1, a_2, 1, rl(s \leftarrow a, b), 2 \rangle$</td>
</tr>
<tr>
<td>$\langle a_1, a_2, 1, rl(s \leftarrow a, b), 2 \rangle$</td>
<td>$\langle a_2, a_1, 2, asm(a), 3 \rangle$</td>
</tr>
<tr>
<td>$\langle a_2, a_1, 3, ctr(a, c), 5 \rangle$</td>
<td>$\langle a_2, a_1, 0, \pi, 6 \rangle$</td>
</tr>
<tr>
<td>$\langle a_2, a_1, 0, \pi, 7 \rangle$</td>
<td></td>
</tr>
</tbody>
</table>

By means of dialogues agents jointly construct a shared framework, as follows:

**Definition 5.** The framework drawn from a dialogue $\delta = \langle u_1, \ldots, u_n \rangle$ is $F_\delta = \langle L, R_\delta, A_\delta, C_\delta \rangle$ where:

- $R_\delta = \{ \rho \mid rl(\rho) \text{ is the content of some } u_i \text{ in } \delta \}$;
- $A_\delta = \{ \alpha \mid asm(\alpha) \text{ is the content of some } u_i \text{ in } \delta \}$;
- $C_\delta$ is a mapping such that, for any $\alpha \in A_\delta$, $C_\delta(\alpha) = \{ \beta \mid ctr(\alpha, \beta) \text{ is the content of some } u_i \text{ in } \delta \}$.

Clearly, the ABA framework drawn from a dialogue represents all information that has been disclosed by the two agents in the dialogue.

**Example 13.** The framework drawn from the dialogue $\delta$ in Example 9 is $F_\delta = \langle L, R_\delta, A_\delta, C_\delta \rangle$, in which

- $A_\delta = \{ a, b, c \}$;
- $R_\delta = \{ s \leftarrow a, q \leftarrow b \}$;
- $C_\delta$ is such that $C_\delta(a) = \{ q \}$, $C_\delta(b) = \{ c \}$, $C_\delta(c) = \{ r \}$.

Note that $F_\delta$ in Example 13 is a flat ABA framework, but, in general, the framework drawn from a dialogue may not be an ABA framework, since $C$ may be empty as in the following example.
Example 14. Given $L = \{s, a\}$, let $\delta$ be:

<table>
<thead>
<tr>
<th></th>
<th>$a_1$</th>
<th>$a_2$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>$\langle a_1, a_2, 0, claim(s), 1 \rangle$</td>
<td>$\langle a_2, a_1, 1, rl(s \leftarrow a), 2 \rangle$</td>
</tr>
<tr>
<td>2</td>
<td>$\langle a_1, a_2, 2, asm(a), 3 \rangle$</td>
<td>$\langle a_2, a_1, 2, rl(s \leftarrow a), 2 \rangle$</td>
</tr>
</tbody>
</table>

The framework drawn from the dialogue, $F_\delta = \langle L, R_\delta, A_\delta, C_\delta \rangle$ has:

- $A_\delta = \{a\}$;
- $R_\delta = \{s \leftarrow a\}$;
- $C_\delta$ is such that $C_\delta(a) = \{\}.$

$F_\delta$, is not an ABA framework as the contrary of assumption $a$ in this framework is the empty set.

Even when the framework drawn from a dialogue is an ABA framework, it may not be flat, as the agents may disagree on the assumptions, as in the following example.

Example 15. Given $L = \{s, a, q, r\}$, let $\delta$ be:

<table>
<thead>
<tr>
<th></th>
<th>$a_1$</th>
<th>$a_2$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>$\langle a_1, a_2, 0, claim(s), 1 \rangle$</td>
<td>$\langle a_2, a_1, 1, rl(s \leftarrow a), 2 \rangle$</td>
</tr>
<tr>
<td>2</td>
<td>$\langle a_1, a_2, 2, asm(a), 3 \rangle$</td>
<td>$\langle a_2, a_1, 2, rl(s \leftarrow a), 2 \rangle$</td>
</tr>
<tr>
<td>3</td>
<td>$\langle a_1, a_2, 3, asm(a), 3 \rangle$</td>
<td>$\langle a_2, a_1, 2, rl(a \leftarrow q), 4 \rangle$</td>
</tr>
<tr>
<td>4</td>
<td>$\langle a_1, a_2, 3, ctr(a, r), 5 \rangle$</td>
<td>$\langle a_2, a_1, 2, rl(a \leftarrow q), 4 \rangle$</td>
</tr>
</tbody>
</table>

The framework $F_\delta = \langle L, R_\delta, A_\delta, C_\delta \rangle$ has:

- $A_\delta = \{a\}$;
- $R_\delta = \{s \leftarrow a, a \leftarrow q\}$;
- $C_\delta$ is such that $C_\delta(a) = \{r\}$.

In this example $a$ is both an assumption and the head of a rule, hence the ABA framework, $F_\delta$, is not flat.

We refine our dialogues so that the frameworks drawn from them are guaranteed to be flat ABA frameworks. This refinement builds upon the notions of legal-move function (with respect to a turn-making function) and outcome function to (with respect to a legal-move function) restrict the kind of utterances allowed in “successful” dialogues.
Definition 6. A legal-move function (with respect to $\gamma$) is a mapping $\lambda : D \mapsto 2^U$ such that, given $\delta = \langle u_1, \ldots, u_n \rangle \in D$, for all $u \in \lambda(\delta)$:

1. $\delta \circ u$ is a dialogue;
2. $\delta \circ u$ is compatible with $\gamma$ if $\delta$ is;
3. if $u = \langle \ldots, T, C, \ldots \rangle$, then there exists no $i$, $1 \leq i \leq n$, such that $u_i = \langle \ldots, T, C, \ldots \rangle$.

Given $\delta = \langle u_1, \ldots, u_n \rangle$, if $u_{m+1} \in \lambda(\langle u_1, \ldots, u_m \rangle)$ for all $m$ such that $0 \leq m < n$, we say that $\delta$ is compatible with $\lambda$.

We use $\Lambda$ to denote the set of all legal-move functions. Also, we often omit to mention the turn-making function when describing a legal-move function.

Given $\lambda_1, \lambda_2 \in \Lambda$, $\lambda = \lambda_1 \cdot \lambda_2$ is defined as follows: for all $u \in U$, $u \in \lambda(\delta)$ if and only if $u \in \lambda_1(\delta) \cap \lambda_2(\delta)$.

Given $\lambda_1, \ldots, \lambda_n \in \Lambda$, $\lambda_1 \cdot \ldots \cdot \lambda_n = \lambda_1 \cdot (\lambda_2 \cdot (\ldots (\lambda_{n-1} \cdot \lambda_n) \ldots))$.

Condition 3 in this definition regulates that there is no repeated utterance to the same target in a dialogue. However, the definition of legal-move function does not impose any “mentalistic” requirement on agents, such as that they utter information they hold true, similar to communication protocols in multi-agent systems [Woo09].

The operator “$\cdot$” defines the composition of legal-move functions such that the resulting legal-move function allows utterances that are in the intersection of all utterances allowed by the legal-move functions in the composition. Naturally, given $\lambda = \lambda_1 \cdot \lambda_2$, if $\delta$ is compatible with $\lambda$, then $\delta$ is also compatible with $\lambda_1$ and $\lambda_2$, respectively. Also, trivially, if $\lambda_1, \lambda_2 \in \Lambda$, $\lambda_1 \cdot \lambda_2 \in \Lambda$.

From now on we will assume a generic true-making function $\gamma$ and we will omit to mention it in an definitions, assuming implicitly that all our dialogues are compatible with a given $\gamma$.

We then ensure the flatness of the framework drawn from a dialogue using a flat legal-move function, as follows.

Definition 7. A flat legal-move function is a legal-move function $\lambda \in \Lambda$ such that, given a dialogue $\delta = \langle u_1, \ldots, u_n \rangle \in D$, for all $u = \langle \ldots, C, \ldots \rangle \in \lambda(\delta)$, then

- $C = asm(\alpha)$ only if there exists no $u_i$ for $1 \leq i \leq n$ with content $rl(\rho)$ and $Head(\rho) = \alpha$;
\( C = rl(\rho) \) only if there exists no \( u_i \) for \( 1 \leq i \leq n \) with content \( asm(\alpha) \) and \( Head(\rho) = \alpha. \)

We use \( \lambda_{fl} \) to denote a generic flat legal-move function. A flat dialogue is a dialogue compatible with a flat legal-move function \( \lambda_{fl}. \)

The dialogue \( \delta \) in Example 15 is not compatible with a flat legal-move function \( \lambda_{fl}. \)

It is easy to see that the framework drawn from a dialogue compatible with a flat legal-move function is flat.

**Lemma 1.** Given a dialogue \( \delta \), if \( \delta \) is compatible with a flat legal-move function \( \lambda_{fl} \), and the framework drawn from \( \delta \), \( F_\delta = \langle L_\delta, R_\delta, A_\delta, C_\delta \rangle \) is an ABA framework, then it is a flat ABA framework.

**Proof.** By Definition 7, for any \( \beta \in L_\delta \), if \( \beta \in A_\delta \) then \( \beta \leftarrow \ldots \notin R_\delta \) and if \( \beta \leftarrow \ldots \in R_\delta \) then \( \beta \notin A_\delta \). Therefore, \( F_\delta \) is flat.

In order to guarantee that the framework drawn from a dialogue is an ABA framework, we will use a specific kind of outcome function imposing each assumption has a non-empty set as its contrary:

**Definition 8.** An outcome function is a mapping \( \omega : D \times \Lambda \mapsto \{true, false\} \).

We use \( \Omega \) to denote the set of all outcome functions.

**Definition 9.** The ABA outcome function, \( \omega_{ABA} \in \Omega \), is such that given a dialogue \( \delta \) and a legal-move function \( \lambda \), \( \omega_{ABA}(\delta, \lambda) = \text{true} \) if and only if \( \delta \) is compatible with \( \lambda \) and the framework \( \langle L_\delta, R_\delta, A_\delta, C_\delta \rangle \) drawn from \( \delta \) is such that for all \( \alpha \in A_\delta \), \( C_\delta(\alpha) \neq \{\}. \)

An ABA dialogue is a dialogue \( \delta \in D \) compatible with a flat \( \lambda_{fl} \in \Lambda \) such that \( \omega_{ABA}(\delta, \lambda_{fl}) = \text{true}. \)

**Example 16.** We revisit the Twelve Angry Men example in Example 10. The ABA framework drawn from the dialogue \( \delta \) shown in Table 3.1 is \( F_\delta = \langle L_\delta, R_\delta, A_\delta, C_\delta \rangle \), as follows.

- \( A_\delta = \{\text{boy\_not\_proven\_guilty}, w1\_is\_believable, w2\_is\_believable\}; \)
- \( R_\delta = \{\text{boy\_innocent} \leftarrow \text{boy\_not\_proven\_guilty} \)
  \text{boy\_proven\_guilty} \leftarrow w1\_is\_believable \)
  \text{boy\_proven\_guilty} \leftarrow w2\_is\_believable \)
\[ w1_{\text{not believable}} \leftarrow w1_{\text{contradicted by w2}} \]
\[ w1_{\text{contradicted by w2}} \leftarrow \]
\[ w2_{\text{not believable}} \leftarrow w2_{\text{has poor eyesight}} \]
\[ w2_{\text{has poor eyesight}} \leftarrow \}\]

- \( C_\delta \) is such that
  - \( C(\text{boy not proven guilty}) = \{\text{boy proven guilty}\} \),
  - \( C(w1_{\text{is believable}}) = \{w1_{\text{is not believable}}\} \),
  - \( C(w2_{\text{is believable}}) = \{w2_{\text{is not believable}}\} \).

We can see that the ABA framework drawn the dialogue is flat hence this dialogue is ABA and flat.

From now on, unless otherwise specified, all dialogues are ABA dialogues and thus the frameworks \( F_\delta \) drawn from dialogues \( \delta \) are flat ABA frameworks.

We aim at using dialogues to successfully determine the acceptability of their claims (with respect to several argumentation semantics). We define (several kinds of) successful dialogues as follows.

**Definition 10.** Given an ABA dialogue \( D_{\alpha_\delta}(\chi) = \delta \), let \( F_\delta \) be the ABA framework drawn from \( \delta \), then \( \delta \) is:

- **a-successful** if and only if \( \chi \) is admissible in \( F_\delta \).
- **g-successful** if and only if \( \chi \) is grounded in \( F_\delta \).
- **i-successful** if and only if \( \chi \) is ideal in \( F_\delta \).

For simplicity, we use admissibility, grounded and ideal as our semantics throughout this thesis. These semantics have been studied in ABA.

**Proposition 1.** Given a dialogue \( \delta \in D \), if \( \delta \) is g-successful, then \( \delta \) is also a-successful and i-successful.

**Proof.** Since \( \delta \) is g-successful, \( \chi \) is grounded. By the definition of the grounded semantics (see Chapter 2 and definition 2.2 of [DMT07]), \( \chi \) is also complete and admissible. Hence g-successful implies a-successful.

By theorem 2.1 (iii) of [DMT07], which states that an ideal set of arguments is a superset of the grounded set of arguments, we have g-successful implies i-successful.
The following examples illustrate the notions of a-/g-/i-successful dialogues.

**Example 17.** Let the dialogue $\delta$ be:

<table>
<thead>
<tr>
<th>$a_1$</th>
<th>$a_2$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\langle a_1, a_2, 0, claim(a), 1 \rangle$</td>
<td>$\langle a_2, a_1, 1, asm(a), 2 \rangle$</td>
</tr>
<tr>
<td>$\langle a_1, a_2, 2, ctr(a, c), 3 \rangle$</td>
<td>$\langle a_2, a_1, 3, asm(c), 4 \rangle$</td>
</tr>
<tr>
<td>$\langle a_1, a_2, 4, ctr(c, a), 5 \rangle$</td>
<td>$\langle a_2, a_1, 5, asm(a), 6 \rangle$</td>
</tr>
<tr>
<td>$\langle a_1, a_2, 0, \pi, 7 \rangle$</td>
<td>$\langle a_2, a_1, 0, \pi, 8 \rangle$</td>
</tr>
</tbody>
</table>

The ABA framework drawn from this dialogue, $\mathcal{F}_\delta = \langle \mathcal{L}, \mathcal{R}_\delta, \mathcal{A}_\delta, \mathcal{C}_\delta \rangle$ has:

- $\mathcal{A}_\delta = \{a, c\};$
- $\mathcal{R}_\delta = \{};$
- $\mathcal{C}_\delta$ is such that $\mathcal{C}_\delta(a) = \{c\}, \mathcal{C}_\delta(c) = \{a\}.$

In this example, we have two arguments $\{a\} \vdash a$ and $\{c\} \vdash c.$ Since $\{a\} \vdash a$ attacks $\{c\} \vdash c$ that attacks $\{a\} \vdash a,$ $\{a\} \vdash a$ defends itself hence is admissible. Therefore $a$ is admissible. Since $\{c\} \vdash c$ is also admissible, $\{a\} \vdash a$ is not grounded nor ideal.

Since $a$ is admissible, but not grounded, nor ideal, in $\mathcal{F}_\delta,$ $\delta$ is a-successful, but not g-/i-successful.

**Example 18.** Let the dialogue, $\delta$ be:

<table>
<thead>
<tr>
<th>$a_1$</th>
<th>$a_2$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\langle a_1, a_2, 0, claim(a), 1 \rangle$</td>
<td>$\langle a_2, a_1, 1, asm(a), 2 \rangle$</td>
</tr>
<tr>
<td>$\langle a_1, a_2, 2, ctr(a, c), 3 \rangle$</td>
<td>$\langle a_2, a_1, 3, asm(c), 4 \rangle$</td>
</tr>
<tr>
<td>$\langle a_1, a_2, 4, ctr(c, b), 5 \rangle$</td>
<td>$\langle a_2, a_1, 5, rl(b \leftarrow), 6 \rangle$</td>
</tr>
<tr>
<td>$\langle a_1, a_2, 0, \pi, 7 \rangle$</td>
<td>$\langle a_2, a_1, 0, \pi, 8 \rangle$</td>
</tr>
</tbody>
</table>

The ABA framework drawn from this dialogue, $\mathcal{F}_\delta = \langle \mathcal{L}, \mathcal{R}_\delta, \mathcal{A}_\delta, \mathcal{C}_\delta \rangle$ has:

- $\mathcal{A}_\delta = \{a, c\};$
- $\mathcal{R}_\delta = \{b \leftarrow\};$
- $\mathcal{C}_\delta$ is such that $\mathcal{C}_\delta(a) = \{c\}, \mathcal{C}_\delta(c) = \{b\}.$
In this example, we have three arguments: \(\{a\} \vdash a\), \(\{c\} \vdash c\) and \(\{} \vdash b\). \(\{c\} \vdash c\) attacks \(\{a\} \vdash a\) and \(\}\vdash b\) defends \(\{a\} \vdash a\) by attacking \(\{c\} \vdash c\). There is no argument attacks \(\}\vdash b\). Hence, \(\{a\} \vdash a\) is admissible, grounded and ideal. So is \(a\).

Since \(a\) is admissible, grounded, and ideal in \(F_\delta\), \(\delta\) is a-/g-/i-successful.

**Example 19.** Let the dialogue, \(\delta\) be:

<table>
<thead>
<tr>
<th>(a_1)</th>
<th>(a_2)</th>
</tr>
</thead>
<tbody>
<tr>
<td>(\langle a_1, a_2, 0, claim(a), 1\rangle)</td>
<td>(\langle a_2, a_1, 1, asm(a), 2\rangle)</td>
</tr>
<tr>
<td>(\langle a_1, a_2, 2, ctr(a, b), 3\rangle)</td>
<td>(\langle a_2, a_1, 3, rl(b \leftarrow c, d), 4\rangle)</td>
</tr>
<tr>
<td>(\langle a_1, a_2, 4, asm(c), 5\rangle)</td>
<td>(\langle a_2, a_1, 4, asm(d), 6\rangle)</td>
</tr>
<tr>
<td>(\langle a_1, a_2, 5, ctr(c, a), 7\rangle)</td>
<td>(\langle a_2, a_1, 6, ctr(d, c), 8\rangle)</td>
</tr>
<tr>
<td>(\langle a_1, a_2, 8, asm(c), 9\rangle)</td>
<td>(\langle a_2, a_1, 7, asm(a), 10\rangle)</td>
</tr>
<tr>
<td>(\langle a_1, a_2, 10, ctr(a, b), 11\rangle)</td>
<td>(\langle a_2, a_1, 9, ctr(c, a), 12\rangle)</td>
</tr>
<tr>
<td>(\langle a_1, a_2, 0, \pi, 13\rangle)</td>
<td>(\langle a_2, a_1, 0, \pi, 14\rangle)</td>
</tr>
</tbody>
</table>

The ABA framework drawn from this dialogue, \(F_\delta = \langle L, R_\delta, A_\delta, C_\delta\rangle\), has:

- \(A_\delta = \{a, c, d\}\);
- \(R_\delta = \{b \leftarrow c, d\}\);
- \(C_\delta\) is such that \(C_\delta(a) = \{b\}\), \(C_\delta(c) = \{a\}\), \(C_\delta(d) = \{c\}\).

The admissible set of assumptions is \(\{a\}\), so is the preferred set of assumptions. Thus, \(a\) is ideal (and admissible), but not grounded, in \(F_\delta\). Hence, the dialogue \(\delta\) is i-successful (and a-successful but not g-successful).

In the context of the Twelve Angry Men example, the claim of the dialogue in Table 3.1, boy innocent, is admissible, grounded, and ideal in \(F_\delta\). Hence, \(\delta\) in Table 3.1 is a-,g-,i-successful.

The definitions in this section jointly establish the foundations of a generic dialogue framework for determining acceptability of claims (in the flat ABA framework drawn from the dialogue). In subsequent sections we build on these foundation notions and show how (a-/g-) successful dialogues can be constructed. But first, we need to ensure that some basic properties of our dialogues are met. Specifically, we specialise our definition of legal-move function (Definition 6) to construct “related” dialogues, as shown in the next section.
3.3 Related Dialogues

Definition 2 defines a class of generic dialogues. Little control is enforced on constructing such dialogues, for example, dialogues such as the following are allowed.

Example 20. This example illustrates a dialogue in that utterances have no semantic relation.

\[
\begin{array}{c|c}
 a_1 & a_2 \\
\hline
\langle a_1, a_2, 0, \text{claim}(s), 1 \rangle & \langle a_2, a_1, 1, rl(p \leftarrow q), 2 \rangle \\
\langle a_1, a_2, 1, \text{asm}(a), 3 \rangle & \langle a_1, a_2, 1, \text{asm}(a), 3 \rangle \\
\end{array}
\]

Here, the second and third utterances are related to their target in the sense of Definition 3. However, there is no “conceptual” relation between these utterances and their targets, e.g., there is no relation between the sentence \( s \) and the rule \( p \leftarrow q \). To have more purposeful dialogues, where utterances are “semantically” related, we define a form of relevance between utterances within a dialogue.

Definition 11. For any two utterances \( u_i, u_j \in U \), \( u_i = \langle \_, \_, \_, C_i, \_ \rangle \), \( u_j = \langle \_, \_, \_, C_j, \_ \rangle \), \( u_j \) is top-down related to \( u_i \) if and only if \( u_j \) is related to \( u_i \) (as in Definition 3) and one of the following cases holds:

1. \( C_j = rl(\rho_j) \), \( \text{Head}(\rho_j) = \beta \) and either \( C_i = rl(\rho_i) \) with \( \beta \in \text{Body}(\rho_i) \), or \( C_i = ctr(\_, \beta) \), or \( C_i = \text{claim}(\beta) \);

2. \( C_j = \text{asm}(\alpha) \) and either \( C_i = rl(\rho) \) with \( \alpha \in \text{Body}(\rho) \), or \( C_i = ctr(\_, \alpha) \), or \( C_i = \text{claim}(\alpha) \);

3. \( C_j = ctr(\alpha, \_) \) and \( C_i = \text{asm}(\alpha) \).

Intuitively, an utterance is top-down related to another if it contributes to expanding an argument (case 1), identifies an assumption in the support of an argument (case 2) or starts the construction of a counter-argument (case 3). Note that an utterance may be top-down related to an utterance from the same agent or not. Also, no pass-utterance can be top-down related to an utterance and no utterance can be top-down related to a pass-utterance.

We use the notion of top-down related-ness to define a new class of legal-move functions:
Definition 12. A top-down related legal-move function $\lambda \in \Lambda$ is such that for $\delta = \langle u_1, \ldots, u_n \rangle$, $\delta \in D$ and for all $u \in \lambda(\delta)$ such that $u$ is a regular-utterance, then $u$ is top-down related to some $u_m$ in $\delta$, $1 \leq m \leq n$.

We use $\lambda_{rt}$ to denote a generic top-down related legal-move function. A top-down related dialogue is a dialogue compatible with a top-down related legal-move function $\lambda_{rt}$.

For simplicity, we use “related” to mean “top-down related” for the rest of this thesis, when there is no ambiguity. We will introduce a notion of bottom-up related in Chapter 7 as another form of semantic relevance.

The dialogue in Example 9 is related (and also flat); the dialogue in Example 20 is not. Until Chapter 7, we only consider top-down related dialogues.

To ease the presentation, we will also define coherent dialogues, as follows.

Definition 13. Given a dialogue $\delta \in D$, a flat legal-move function $\lambda_{fl}$, a related legal-move function $\lambda_{rt}$, we say $\delta$ is coherent if and only if $\delta$ is compatible with $\lambda_{fl} \cdot \lambda_{rt}$, for some flat legal-move function $\lambda_{fl}$ and some related legal-move function $\lambda_{rt}$.

Since both flatness and related-ness are fundamental to any dialogues, in the rest of this thesis, unless otherwise specified, all dialogues are coherent.

3.4 Strategy-move Functions

In ABA dialogues, agents make utterances that contain rules, assumptions, contraries and passes. The selection of utterances must satisfy the integrity of the dialogue and the aims of the agents. Legal-move functions are used to keep the integrity of dialogues. As presented in Chapter 2, there are several types of dialogues, where each type has its own goals and agents have different aims in different types of dialogues. Here, we introduce strategy-move functions to specify agents’ behaviours that are suitable for their aims and for the goals of the dialogues they are engaged in.

Definition 14. A strategy-move function for agent $a_i$ ($i = 1, 2$) is a mapping $\phi : D \times \Lambda \mapsto 2^{U_i}$, such that, given $\lambda \in \Lambda$ and $\delta \in D$: $\phi(\delta, \lambda) \subseteq \lambda(\delta)$.

Given a dialogue $D_{a_j}(\chi) = \delta = \langle u_1, \ldots, u_n \rangle$ compatible with a legal-move function $\lambda$ and a strategy-move function $\phi$ for $a_k$ ($k = i, j$), if, for all $u_m = \langle a_k, \ldots, u_n \rangle$, $1 < m \leq n$, $u_m \in \phi(\langle u_1, \ldots, u_{m-1} \rangle, \lambda)$, then we say that $\delta$ is
constructed with $\phi$ with respect to $a_k$ and $a_k$ uses $\phi$ in $\delta$. Furthermore, if $a_i$ and $a_j$ both use $\phi$ in some $\delta$, then we say that $\delta$ is constructed with $\phi$.

Given strategy-move functions $\phi_1, \phi_2$, $\phi = \phi_1 \cdot \phi_2$ is defined as follows: for all $u \in U$, $u \in \phi(\delta, \lambda)$ if and only if $u \in \phi_1(\delta, \lambda) \cap \phi_2(\delta, \lambda)$.

We use $\Phi$ to denote the set of all strategy-move functions.

We will study strategy-move functions in more detail in Chapter 5. There, we will also discuss the strategy-move functions used by $a_1$ and $a_2$ in the dialogue shown in Table 3.1.

### 3.5 Related Work

McBurney and Parsons [MP09] give an overview of dialogue games for argumentation. In that work, the authors explain the syntax and semantics of dialogue protocols. Our work can be seen as providing a novel dialogue game for ABA, hence the syntax of our dialogue is based on ABA and the semantics of our dialogues are standard ABA semantics.

Prakken in [Pra06] reviews dialogue systems for persuasion. Our dialogue model is generic hence it is not restricted to persuasion. As we will illustrate in Chapters 6 and 7, our dialogue model can be applied in information-seeking, inquiry, persuasion, conflict-resolution, and discovery dialogues.

Parsons et al [PMSW07] examine three notions of relevance in dialogues where utterances are arguments and attacks:

- **R1** (every new utterance has a direct impact on the claim),
- **R2** (every new utterance directly or indirectly impacts the claim), and
- **R3** (every new utterance has a direct impact on the previous one).

Our utterances are at a finer granularity level, as they correspond to rules, assumptions, and contraries, whereas in [PMSW07] utterances are at the argument level, i.e., each utterance contains an argument. Thus, there is no direct mapping between our work and their relevance.

On the other hand, we are going to introduce several other legal-move functions in Chapter 4, and some of these legal-move functions will resemble Parsons’ R1 and R2 relevances.

Black and Hunter [BH09] present a formal system for inquiry dialogues based on DeLP [GS04] as the underlying argumentation framework. Our work differs
from theirs as (1) it defines a mechanism for any type of dialogue whereas they focus on inquiry dialogues; (2) it uses ABA whereas they use DeLP; (3) it does not force an agent to disclose all knowledge whereas they force complete full disclosure of all relevant knowledge for the purpose of inquiry; (4) it does not force a strict interleaving whereas they do.

Prakken [Pra05] defines a formal system for persuasion. The main differences with our work are: (1) since that work focuses on persuasion dialogues, proponent and opponent roles are pre-assigned to agents before the dialogue whereas in our work agents can play both roles within the same dialogue; (2) Prakken focuses the grounded semantics, whereas we allow admissibility, grounded, and ideal semantics; (3) his set of utterances refers to arguments and attacks, as in the case of [PMSW07]; (4) he forces the support of arguments to be minimal, whereas we do not, in the spirit of [DTM10]; (5) he does not allow agents to jointly construct arguments whereas we do.

An early version of this Chapter has been published in [FT11a]. Several improvements have been implemented here. Firstly, unlike [FT11a], we render explicit the turn making function in Definition 4. Whereas the dialogue model in [FT11a] implicitly forces a strict interleaving between agents, construction given here does not. Secondly, though the legal-move function definition (Definition 6) is similar to Definition 5 in [FT11a], we define $\lambda : \mathcal{D} \rightarrow 2^{\mathcal{U}}$ instead of $\lambda : \mathcal{D} \rightarrow \mathcal{U}$ there. This new definition is more precise as it indicates that there might be more than one utterance that satisfies a legal-move function.

### 3.6 Conclusion

In this chapter, we have presented the basis of our dialogue system. We have defined legal-move, outcome, and strategy-move functions, playing a specific role in the generation of dialogues. Basically, legal-move functions work like dialogue protocols and underpin construction of dialogues. Outcome functions are used to ensure that certain dialogue properties are fulfilled, so some conclusions about the claim of the dialogue can be drawn. Strategy-move functions are used to model agent behaviours such that certain agent goals will be reached if a dialogue is constructed with some strategy-move functions (as we will further explore later).

A summary of the main concepts introduced in this chapter is given in Table 3.3. A summary of the types of dialogues introduced in this chapter can be seen in Table 4.5 from Page 92.
Table 3.3: Summary of concepts introduced in Chapter 3.

<table>
<thead>
<tr>
<th>Turn-making ($\gamma$):</th>
<th>Returns the agent who makes the next utterance</th>
</tr>
</thead>
<tbody>
<tr>
<td>Legal-move ($\lambda$):</td>
<td>Returns utterances that satisfy dialogue protocols</td>
</tr>
<tr>
<td>Outcome ($\omega$):</td>
<td>Tests if certain dialogue property holds</td>
</tr>
<tr>
<td>Strategy-move ($\phi$):</td>
<td>Returns utterances that fulfill agents’ goals</td>
</tr>
</tbody>
</table>

In the next Chapter, we will study our dialogue protocol (legal-move functions and outcome functions) in detail. We will show how dialogues with various level of “generality” can be constructed with different legal-move functions and how to construct dialogues which have their claims being acceptable with respect to information disclosed through the dialogue. We will formally link argumentation semantics with dialogues.
4 Dialogues and Argumentation Semantics

4.1 Introduction

In Chapter 3, we have introduced some basic concepts, e.g., legal-move functions, outcome functions, and strategy-move functions, that are used to define our dialogue model. In this chapter, we study specific classes of legal-move functions and outcome functions in detail. We show how legal-move functions can be used to construct dialogues.

Since our aim is to construct dialogues with “acceptable” claims, we link dialogues with argumentation semantics. Generally speaking, by ensuring that a dialogue being compatible with some specifically defined legal-move functions, we can guarantee that the claim of dialogues is acceptable with respect to given argumentation semantics, notably the admissibility and grounded semantics. This result is obtained using debate trees and debate forests.

The chapter is organised as follows. Section 4.2 presents our notion of debate tree. Section 4.3 presents focused dialogues that are constructed using debate trees. Section 4.4 presents debate forests and dialogues constructed using them. Section 4.5 illustrates the concepts we introduced in this chapter. Section 4.6 compares our refined dialogue model, as defined in this chapter, with a few other models. Section 4.7 concludes.

4.2 Debate Trees

We will see below how utterances can be decided in terms of debate trees drawn from a dialogue. Informally, each node of debate trees

1. contains one sentence,

2. is tagged as either unmarked (um), marked-rule (mr) or marked-assumption (ma),
3. is labelled either proponent or opponent as in the abstract dispute trees of [DKT06] (see Chapter 2), and

4. has an ID taken from an utterance in the dialogue.

The sentence of each node in a debate tree represents an argument’s claim or (element of a) support. A node is tagged unmarked if its sentence is only mentioned in the claim or the body of a rule or contrary of an assumption, but without any further examination, marked-rule if its sentence is the head of an uttered rule, and marked-assumption if its sentence has been explicitly uttered as an assumption. A node is labelled proponent (opponent) if it is (directly or indirectly) for (against) the claim of the dialogue. The ID is used to identify the node’s corresponding utterance in the dialogue.

As an illustration, possible nodes from utterances in Example 9 are:

\[
(s, \text{um}: P[1]),
(s, \text{mr}: P[1]),
(a, \text{um}: P[3]),
(a, \text{ma}: P[4]),
(q, \text{um}: O[5]).
\]

Note that since debate trees are constructed in steps as dialogues proceed, not all nodes in the above list are in a debate tree at every step, namely, certain nodes tagged as um, such as \((s, \text{um}: P[1])\) and \((q, \text{um}: O[5])\) are replaced by nodes with “updated” information about the same sentences, i.e., \((s, \text{mr}: P[1])\) and \((q, \text{mr}: O[5])\), respectively, after new utterances are inserted into the dialogue.

For instance, the content of utterance 5 in the dialogue in Example 9 is \(\text{ctr}(a,q)\), so by this utterance, \(q\) is only mentioned as a contrary of \(a\), it is uncertain whether \(q\) is an assumption or the head of a rule, hence the tag in \((q, \text{um}: O[5])\) is um.

However, after utterance 6, which has the content \(\text{rl}(q \leftarrow b)\), it is known that \(q\) is the head of the rule \(q \leftarrow b\), hence \((q, \text{um}: O[5])\) is replaced by \((q, \text{mr}: O[5])\).

Thus nodes tagged um may be replaced by other nodes during the construction of a debate tree.

In a debate tree, nodes are connected in two cases:

1. they belong to the same ABA argument, and

2. they form attacks between two ABA arguments.
Figure 4.1: Nodes in a debate tree connected as in ABA arguments (left) or attacks between ABA arguments (right).

Figure 4.1 (left) gives an example of case (1), again, the dialogue in Example 9. Here, the left column contains connected nodes in a debate tree; the right column shows the corresponding ABA argument. In case (2), two nodes $n_1 = (\alpha, ma : L[id])$ and $n_2 = (\beta, : L'[\_])$, $L, L' \in \{P, O\} L' \neq L$, are connected if there is an utterance $u = (\_ , id, ctr(\alpha, \beta), \_)$ in the dialogue. (e.g., see Figure 4.1 right).

Hence, the two nodes are connected if the parent node contains an assumption and the child node contains a contrary of that assumption.

We give the formal definition of a debate tree as follows.

**Definition 15.** Given $D^\chi_{hi}(\chi) = \delta$, the debate tree drawn from $\delta$ is a tree $\mathcal{T}(\delta)$, where:

1. nodes of $\mathcal{T}(\delta)$ can be characterised as follows:
   a) nodes are tuples $(S, F : L[U])$ where
      - $S$ is a sentence in $L$,
      - $F$ (the tag) is either $um$ (unmarked), $mr$ (marked-rule) or $ma$ (marked-assumption),
      - $L$ (the label) is either $P$ (proponent) or $O$ (opponent),
      - $U \in ID$ (the ID);
   b) $(\beta, : \_[,\_])$ is a node in $\mathcal{T}(\delta)$ if and only if there is an utterance $(\_ , id, C, \_)$ in $\delta$ such that $C$ is either:
      i. $claim(\beta)$, or
      ii. $s_0 \leftarrow s_1, \ldots, s_m$ where $\beta = s_i$ for some $i$ with $0 < i \leq m$, or
      iii. $asm(\beta)$, or
      iv. $ctr(\_ , \beta)$;
   c) $(\beta, um : \_[,\_])$ is a node in $\mathcal{T}(\delta)$ if and only if there is no utterance $(\_ , t, rl(\beta \leftarrow \_), \_)$ or $(\_ , t, asm(\beta), \_)$ in $\delta$;
Lemma 2. Given a dialogue $\mathcal{D}_{\pi}^{a}(\chi) = \delta$, the debate tree $\mathcal{T}(\delta)$ is $\mathcal{T}^{m}(\delta)$ in the sequence $\mathcal{T}^{0}(\delta), \mathcal{T}^{1}(\delta), \ldots, \mathcal{T}^{m}(\delta)$ constructed inductively from the $\pi$-pruned sequence $\delta' = \langle u'_{1}, \ldots, u'_{m} \rangle$ obtained from $\delta$, as follows ($L, L' \in \{P, O\}, L \neq L'$):

1. $\mathcal{T}^{0}(\delta)$ is empty;

2. $\mathcal{T}^{1}(\delta)$ contains a single node:

$$\langle \chi, um : P[id_{1}] \rangle,$$
where $id_1$ is the identifier of $u'_1 = u_1$.

3. let $T^i(\delta)$ be the $i$-th tree, for $0 < i < m$, let $u'_{i+1} = \langle _, _, t, C, id \rangle$, and let $\langle _, _, C_t, t \rangle$ be the target utterance of $u'_{i+1}$; then $T^{i+1}(\delta)$ is obtained as follows:
   
a) If $C = rl(\beta_0 \leftarrow \beta_1, \ldots, \beta_l)$ then $T^{i+1}(\delta)$ is $T^i(\delta)$ with additional $l$ nodes:
      $$\langle \beta_1, um : L[id] \rangle, \ldots, \langle \beta_l, um : L[id] \rangle$$
      as children of the node $\langle \beta_0, _ : L[t] \rangle$, and the node $\langle \beta_0, _ : L[t] \rangle$ is replaced by $\langle \beta_0, mr : L[t] \rangle$;
   
b) If $C = asm(\alpha)$ then $T^{i+1}(\delta)$ is $T^i(\delta)$ with the node $\langle \alpha, um : L[t] \rangle$ replaced by $\langle \alpha, ma : L[id] \rangle$.
   
c) If $C = ctr(\alpha, \beta)$ then $T^{i+1}(\delta)$ is $T^i(\delta)$ with an additional node: $\langle \beta, um : L[id] \rangle$ child of $\langle \alpha, ma : L'[t] \rangle$.

Proof. We show, given a dialogue $\delta$, that this inductive process constructs a debate tree $T(\delta)$ as in Definition 15.

Condition 1(a) and 1(b) in Definition 15 are trivially true since nodes in $T^0(\delta), \ldots, T^m(\delta)$ are nodes inserted in accordance with utterances in $\delta$ hence if a sentence $\beta$ is in a node, $\beta$ must be in some utterance in the dialogue. Condition 1(c) in definition 15 is true as 3(a) and 3(b) in the lemma specify how nodes tagged $um$ are replaced by nodes tagged $ma$ and $mr$. Condition 2(a) through 2(f) in Definition 15 are jointly met by 3(a) through 3(c) in the lemma.

Hence, this inductive process yields a debate tree as per Definition 15.

The debate tree $T(\delta)$ for $\delta$ in Example 9 is shown in Figure 4.2. The construction of this tree is shown in Figure 4.3.

It is easy to see that debate trees for related dialogues are always guaranteed to be well-formed, in that each non-root node in them has exactly one parent.\(^1\)

Given the construction of debate trees, arguments can be drawn from a debate tree, as follows:

**Definition 16.** An argument drawn from a debate tree $T(\delta)$ is a sub-tree $T$ of $T(\delta)$ such that:

\(^1\)Note that, as stated in Chapter 3, we assume all dialogues in discussion are coherent, hence flat and related.
1. all nodes in $\mathcal{T}$ have the same label (either P or O);

2. if there is an utterance $\langle \cdot, \cdot, \cdot, rl(\beta_0 \leftarrow \beta_1, \ldots, \beta_m), t \rangle$ in $\mathcal{F}$ and $(\beta_0, mr : L[t])$ is in $\mathcal{T}$, then $(\beta_1, \cdot : L[\cdot]), \ldots, (\beta_m, \cdot : L[\cdot])$ are in $\mathcal{T}$;

3. there does not exist a node $n'$ in $\mathcal{T}(\delta)$ such that $n'$ is parent or child of some node $n_i$ in $\mathcal{T}$ and $n_i, n'$ have the same label.

An argument $\mathcal{T}$ drawn from a debate tree is actual if for all nodes $\langle \cdot, F : \cdot, \cdot \rangle$ in $\mathcal{T}$, $F$ is either mr or ma; if there is at least one node of the form $\langle \cdot, um : \cdot, \cdot \rangle$ in $\mathcal{T}$, then $\mathcal{T}$ is potential.

An argument is a proponent argument if its nodes are labelled P; and an opponent argument if its nodes are labelled O.

Given a debate tree $\mathcal{T}(\delta)$, we say that a node is in an argument (in $\mathcal{T}(\delta)$) if and only if it is a node in some argument drawn from $\mathcal{T}(\delta)$.

The sentence $\chi$ in the root of an argument $\mathcal{T}$ drawn from a debate tree is the claim of $\mathcal{T}$. If $\mathcal{T}$ is actual, and $\chi$ is its claim, $\mathcal{T}$ is written as $S \vdash^t \chi$, where $S = \{\alpha| (\alpha, ma : \cdot, \cdot) \text{ is a node in } \mathcal{T}\}$.

**Example 21.** In the debate tree shown in Figure 4.2, two (actual) proponent arguments and one (actual) opponent argument can be drawn from this tree, as shown in Figure 4.4.

Definition 16 is very useful in that it gives a means of talking about arguments in the context of a debate tree. This notion will be repeatedly used later on, as supported by the following lemma: actual arguments can be mapped to equivalent ABA arguments.
\[ T_1(\delta) \quad T_2(\delta) \quad T_3(\delta) \quad T_4(\delta) \quad T_5(\delta) \]
\[
(\delta) \quad (\delta) \quad (\delta) \quad (\delta) \quad (\delta)
\]
\[
(\delta) \quad (\delta) \quad (\delta) \quad (\delta) \quad (\delta)
\]
\[
(\delta) \quad (\delta) \quad (\delta) \quad (\delta) \quad (\delta)
\]
\[
(\delta) \quad (\delta) \quad (\delta) \quad (\delta) \quad (\delta)
\]
\[
(\delta) \quad (\delta) \quad (\delta) \quad (\delta) \quad (\delta)
\]
\[
(\delta) \quad (\delta) \quad (\delta) \quad (\delta) \quad (\delta)
\]

\[ T_6(\delta) \quad T_7(\delta) \quad T_8(\delta) \quad T_9(\delta) \]
\[
(\delta) \quad (\delta) \quad (\delta) \quad (\delta)
\]
\[
(\delta) \quad (\delta) \quad (\delta) \quad (\delta)
\]
\[
(\delta) \quad (\delta) \quad (\delta) \quad (\delta)
\]
\[
(\delta) \quad (\delta) \quad (\delta) \quad (\delta)
\]
\[
(\delta) \quad (\delta) \quad (\delta) \quad (\delta)
\]

Figure 4.3: The construction of the debate tree in Figure 4.2. Here the tree is constructed using utterances \(u_1', \ldots, u_9'\) as given in Example 11.

**Lemma 3.** For each actual argument \(S \vdash \beta\) drawn from a debate tree \(T(\delta)\), there exists an ABA argument \(S \vdash \beta\) in the ABA framework drawn from \(\delta\).

This lemma is trivially true as a node in an actual argument can be mapped to a node in an ABA argument by 1) dropping the tag and the ID, and 2) adding nodes \(\tau\) as children of leaf nodes of the form \((\_\_ \_ \_ \_ : \_\_ \_ \_\_\_\_\_\_)\) (as each of these node represents a rule with an empty body).

The ABA arguments corresponding to the arguments drawn from the debate tree drawn from the dialogue in Example 9 (see Figure 4.2) are: \(\{a\} \vdash s\), \(\{b\} \vdash q\), and \(\{c\} \vdash c\).

We consider now restricted forms of debate trees, that we then use below to refine our notion of legal-move function.

**Definition 17.** A debate tree \(T(\delta)\) is **patient** if and only if for all nodes \(n = (\_\_ \_ \_ \_ : \_\_ \_ \_\_\_\_)\) in \(T(\delta)\) such that \(n\) has a child, then \(n\) is in an actual argument
Figure 4.4: Two proponent arguments (left) and one opponent argument (right) drawn from the debate tree shown in Figure 4.2.

drawn from $T(\delta)$.

Arguments in a patient tree are fully expanded (cf. actual) before being attacked. The tree in Figure 4.2 is patient. The debate tree in the following example is not.

**Example 22.** Let a dialogue $\delta$ be follows:

<table>
<thead>
<tr>
<th>$a_1$</th>
<th>$a_2$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$(a_1, a_2, 0, claim(s), 1)$</td>
<td>$(a_2, a_1, 1, rl(s \leftarrow a, b), 2)$</td>
</tr>
<tr>
<td>$(a_1, a_2, 2, asm(a), 3)$</td>
<td>$(a_2, a_1, 3, ctr(a, c), 4)$</td>
</tr>
</tbody>
</table>

The debate tree drawn from this dialogue is in Figure 4.5. This debate tree is not patient as the argument $\{a\} \vdash s$ is not actual, since the node $(b, um : P[2])$ has the flag $um$, yet the assumption node $(a, ma : P[3])$ already has a child $(c, um : O[4])$.

Figure 4.5: The non-patient debate tree drawn from the dialogue in Example 22.

The restricted form of legal-move function we consider is guaranteed to generate patient trees, as follows.

**Definition 18.** A legal-move function $\lambda \in \Lambda$ is patient if and only if for every $\delta \in D$ such that $T(\delta)$ is patient, for every $u \in \lambda(\delta)$, $T(\delta \circ u)$ is still patient.

We use $\lambda_p$ to denote a generic patient legal-move function. A patient dialogue is a dialogue compatible with a patient legal-move function $\lambda_p$.

In the rest of this chapter, all dialogues are patient.
This definition requires agents to consult the debate tree before making utterances. Thus, when an agent decides what to utter, it needs to take the current debate tree into account and make sure that its new utterance will keep the tree patient. Thus, the debate tree drawn from a dialogue can be seen as a commitment store [WK95] holding information that agents disclose and share using the dialogue.

Another legal-move function that relies on the debate tree as a commitment store is the filtered legal-move function, defined as follows.

**Definition 19.** A debate tree $T(\delta)$ is filtered if and only if for any two nodes $n_1 = (\beta, ma : L[id_1]), n_2 = (\beta, ma : L[id_2]), L \in \{P, O\}, n_1 \neq n_2$, if $n_1$ has a child $n'_1$ in an actual argument $T_1$ and $n_2$ has a child $n'_2$ in an actual argument $T_2$, then $T_1 \neq T_2$.

Figure 4.6 shows two debate trees. The one on the left is a filtered debate tree whereas the one on the right is not. In the filtered debate tree (left), since the node $(a, ma : P[2])$ has a child $(b, ma : O[4])$, so the node $(a, ma : P[6])$ cannot be attacked by the same argument $\{b\} \vdash t$, hence $(b, um : O[7])$ is an unmarked node, rather than a marked assumption node.

Note, when comparing two arguments for equality, we only consider the sentence and the tag in each argument, while ignoring the label and the ID.

**Definition 20.** A legal-move function $\lambda \in \Lambda$ is filtered if and only if for every $\delta \in \mathcal{D}$ such that $T(\delta)$ is filtered, for every $u \in \lambda(\delta)$, $T(\delta \circ u)$ is still filtered.

We use $\lambda_{fi}$ to denote a generic filtered legal-move function. A filtered dialogue is a dialogue compatible with a filtered legal-move function $\lambda_{fi}$.

Filtered legal-move functions bring efficiency to dialogues. For dialogues that are compatible with a filtered legal-move function, any assumption is attacked
at most once by the same argument. The following two examples show the two dialogues that draw the two debate trees in Figure 4.6.

**Example 23.** A dialogue $\delta$ that is compatible with a filtered legal-move function is shown in the following table. We can see that the argument $\{a\} \vdash a$ is attacked by $\{b\} \vdash b$ once and only once. The debate tree drawn from this dialogue is shown in Figure 4.6 (left).

<table>
<thead>
<tr>
<th>$a_1$</th>
<th>$a_2$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$(a_1, a_2, 0, \text{claim}(a), 1)$</td>
<td>$(a_2, a_1, 1, \text{asm}(a), 2)$</td>
</tr>
<tr>
<td>$(a_1, a_2, 2, \text{ctr}(a, b), 3)$</td>
<td>$(a_2, a_1, 3, \text{asm}(b), 4)$</td>
</tr>
<tr>
<td>$(a_1, a_2, 4, \text{ctr}(b, a), 5)$</td>
<td>$(a_2, a_1, 5, \text{asm}(a), 6)$</td>
</tr>
<tr>
<td>$(a_1, a_2, 6, \text{ctr}(a, b), 7)$</td>
<td>$(a_2, a_1, 0, \pi, 8)$</td>
</tr>
<tr>
<td>$(a_2, a_1, 0, \pi, 9)$</td>
<td></td>
</tr>
</tbody>
</table>

**Example 24.** A dialogue $\delta$ that is not compatible with a filtered legal-move function is shown in the following table. In this dialogue, the argument $\{a\} \vdash a$ is attacked by $\{b\} \vdash b$ twice, repeatedly. The debate tree drawn from this dialogue is shown in Figure 4.6 (right).

<table>
<thead>
<tr>
<th>$a_1$</th>
<th>$a_2$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$(a_1, a_2, 0, \text{claim}(a), 1)$</td>
<td>$(a_2, a_1, 1, \text{asm}(a), 2)$</td>
</tr>
<tr>
<td>$(a_1, a_2, 2, \text{ctr}(a, b), 3)$</td>
<td>$(a_2, a_1, 3, \text{asm}(b), 4)$</td>
</tr>
<tr>
<td>$(a_1, a_2, 4, \text{ctr}(b, a), 5)$</td>
<td>$(a_2, a_1, 5, \text{asm}(a), 6)$</td>
</tr>
<tr>
<td>$(a_1, a_2, 6, \text{ctr}(a, b), 7)$</td>
<td>$(a_2, a_1, 0, \pi, 8)$</td>
</tr>
<tr>
<td>$(a_2, a_1, 7, \text{asm}(b), 8)$</td>
<td></td>
</tr>
</tbody>
</table>

The next notion we introduce is the **exhaustiveness**, which gives sense of a dialogue “completeness”.

**Definition 21.** The exhaustive outcome function $\omega_{ex}$ is such that, given $\delta \in \mathcal{D}$, $\lambda \in \Lambda$ and $(\mathcal{L}, \mathcal{R}_\delta, A_\delta, C_\delta)$ the framework drawn from $\delta$, $\omega_{ex}(\delta, \lambda) = \text{true}$ if and only if $\omega_{ABA}(\delta \circ u', \lambda) = \text{true}$ and $\not\exists u' \in \lambda(\delta)$ with content either:

- $\text{asm}(\alpha)$, for $\alpha \in A_\delta$, or
- $r l(\rho)$, for $\rho \in \mathcal{R}_\delta$, or
- $\text{ctr}(\alpha, \beta)$, for $\beta = C_\delta(\alpha)$,

such that $\omega_{ABA}(\delta \circ u', \lambda) = \text{true}$.

We refer to dialogues for which $\omega_{ex}$ is true as **exhaustive**.
Note that exhaustiveness does not force agents to contribute to dialogues with all relevant information they hold. Rather, it enforces that if an utterance \( u \) with a rule or an assumption as its content has been made in accordance to a certain \( \lambda \), then the pass utterance is not allowed to be uttered where it is possible to make another utterance with the same content as \( u \), if such \( u \) is allowed by \( \lambda \). The following example illustrates this notion of exhaustiveness.

**Example 25.** Given a patient legal-move function \( \lambda_p \), the following dialogue \( \delta \) compatible with \( \lambda_p \) is not exhaustive as

\[
\langle a_2, a_1, 5, rl(c \leftarrow), 6 \rangle
\]

can be uttered in place for

\[
\langle a_2, a_1, 0, \pi, 6 \rangle,
\]

since \( rl(c \leftarrow) \) has been uttered in:

\[
\langle a_2, a_1, 3, rl(c \leftarrow), 4 \rangle.
\]

<table>
<thead>
<tr>
<th>( a_1 )</th>
<th>( a_2 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \langle a_1, a_2, 0, \text{claim}(s), 1 \rangle )</td>
<td>( \langle a_2, a_1, 1, rl(s \leftarrow a, b), 2 \rangle )</td>
</tr>
<tr>
<td>( \langle a_1, a_2, 2, rl(a \leftarrow c), 3 \rangle )</td>
<td>( \langle a_2, a_1, 3, rl(c \leftarrow), 4 \rangle )</td>
</tr>
<tr>
<td>( \langle a_1, a_2, 2, rl(b \leftarrow c), 5 \rangle )</td>
<td>( \langle a_2, a_1, 0, \pi, 6 \rangle )</td>
</tr>
<tr>
<td>( \langle a_1, a_2, 0, \pi, 7 \rangle )</td>
<td></td>
</tr>
</tbody>
</table>

Similarly to [DKT06], we introduce the defence set and the culprits of a debate tree. We use these two concepts to prove our soundness result in the next section.

**Definition 22.** Given a debate tree \( T(\delta) \),

- the defence set \( \mathcal{DEF}(T(\delta)) \) is the union of all assumptions \( \alpha \) in proponent nodes of the form \( n = (\alpha, ma : P[\_]) \), such that \( n \) is in an actual argument;

- the culprits \( \mathcal{CUL}(T(\delta)) \) are given by the set of all assumptions \( \alpha \) in opponent nodes \( n = (\alpha, ma : O[\_]) \) such that the child of \( n \) in \( T(\delta) \) is \( n' = (\_, \_ : P[\_]) \) and both \( n \) and \( n' \) are in actual arguments.

In this section, we have introduced some basic notions used in dialogue construction. In the next two sections, we are going to formally link dialogues with argumentation semantics and show how a-/g-successful dialogues can be constructed.
4.3 Focused Dialogues

With debate tree defined, in this section we link dialogues and argumentation semantics. We first define the notion of a node being properly attacked in a debate tree, as follows.

**Definition 23.** Given a debate tree $T(\delta)$, a node $n$ in $T(\delta)$ is properly attacked if and only if $n$ is of the form $(\_, ma : O[\_])$ and $n$ has a child $n'$, such that $n'$ is in an actual argument.

Thus, an assumption node is properly attacked if and only if there is an actual argument attacking the assumption in the node.

Then we introduce the notion of focused, as follows.

**Definition 24.** A debate tree $T(\delta)$ is focused if and only if

1. for all arguments $A$ drawn from $T(\delta)$, if $A$ contains a node $(\_, ma : O[\_])$, then there is at most one node of the form $(\_, ma : O[\_])$ in $A$ such that $n$ has any child, and such node $n$ has a single child;

2. for all nodes of the form $(\beta_0, mr : P[\_])$ with children $(\beta_1, \_ : P[\_]), \ldots, (\beta_n, \_ : P[\_])$ there must be an utterance in $\delta$ of the form $(\_, \_, t, rl(\beta_0 \leftarrow \beta_1, \ldots, \beta_n), \_)$.

In focused trees, no alternative ways to defend claims are considered simultaneously, i.e., an opponent argument is only attacked by a single proponent argument whereas a proponent argument can be attacked in as many ways as the number of its assumptions. Moreover, the claim is supported by a single set of proponents. The tree in Figure 4.2 is focused. (We call trees and dialogues that are not focused as non-focused trees and dialogues, respectively.) Figures 4.7 and 4.8 show two non-focused debate trees drawn from dialogues in Tables 4.1 and 4.2, respectively, violating condition 1 and 2 in Definition 24, respectively.

**Definition 25.** A legal-move function $\lambda \in \Lambda$ is focused if and only if for every $\delta \in D$ such that $T(\delta)$ is focused, for every $u \in \lambda(\delta)$, $T(\delta \circ u)$ is still focused.

We use $\lambda_f$ to denote a generic focused legal-move function. A focused dialogue is a dialogue compatible with a focused legal-move function $\lambda_f$. 
Table 4.1: A dialogue that draws a non-focused debate tree.

<table>
<thead>
<tr>
<th>$a_1$</th>
<th>$a_2$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\langle a_1, a_2, 0, \text{claim}(\chi), 1 \rangle$</td>
<td>$\langle a_2, a_1, 1, \text{asm}(\chi), 2 \rangle$</td>
</tr>
<tr>
<td>$\langle a_1, a_2, 2, \text{ctr}(\chi, s), 3 \rangle$</td>
<td>$\langle a_2, a_1, 3, \text{rl}(s \leftarrow a, b), 4 \rangle$</td>
</tr>
<tr>
<td>$\langle a_1, a_2, 4, \text{asm}(a), 5 \rangle$</td>
<td>$\langle a_2, a_1, 4, \text{asm}(b), 6 \rangle$</td>
</tr>
<tr>
<td>$\langle a_1, a_2, 5, \text{ctr}(a, c), 7 \rangle$</td>
<td>$\langle a_2, a_1, 6, \text{ctr}(b, c), 8 \rangle$</td>
</tr>
</tbody>
</table>

Table 4.2: Another dialogue that draws a non-focused debate tree.

<table>
<thead>
<tr>
<th>$a_1$</th>
<th>$a_2$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\langle a_1, a_2, 0, \text{claim}(s), 1 \rangle$</td>
<td>$\langle a_2, a_1, 1, \text{rl}(s \leftarrow a), 2 \rangle$</td>
</tr>
<tr>
<td>$\langle a_1, a_2, 1, \text{rl}(s \leftarrow b), 3 \rangle$</td>
<td></td>
</tr>
</tbody>
</table>

Figure 4.7: A non-focused tree. Note, in this tree, the opponent argument $\{a, b\} \vdash s$ has two proponent nodes as its children: $(c, \text{um}: P[7])$ and $(c, \text{um}: P[8])$.

Figure 4.8: A non-focused tree. Note, in this tree, a proponent node (the root) has two “sets” of proponent children from two different utterances.
With all needed ingredients, we are ready to show how a-/g-successful dialogues can be constructed. We first define two specific legal-move functions by composing several legal-move functions that we have defined so far.

**Definition 26.** Given a flat legal-move function \( \lambda_{fl} \), a related legal-move function \( \lambda_{rt} \), a patient legal-move function \( \lambda_{p} \), a focused legal-move function \( \lambda_{f} \), and a filtered legal-move \( \lambda_{fi} \),

- a *(focused) admissible legal-move function* is defined as:
  \[
  \lambda = \lambda_{fl} \cdot \lambda_{rt} \cdot \lambda_{p} \cdot \lambda_{f} \cdot \lambda_{fi};
  \]

- a *(focused) grounded legal-move function* is defined as:
  \[
  \lambda = \lambda_{fl} \cdot \lambda_{rt} \cdot \lambda_{p} \cdot \lambda_{f}.
  \]

We use \( \lambda_{ADM} \) and \( \lambda_{GND} \) to denote generic (focused) admissible and grounded legal-move functions, respectively.

**Example 26.** A sample dialogue compatible with a grounded legal-move function \( \lambda_{GND} \) is:

The debate tree drawn from this dialogue is in Figure 4.9.

**Example 27.** The dialogue \( \delta \) in Example 19 is compatible with an admissible legal-move function \( \lambda_{ADM} \), but not a grounded legal-move function \( \lambda_{GND} \). The debate tree \( T(\delta) \) drawn from \( \delta \) is shown in Figure 4.10. As illustrated in this example, the filtering has been applied to both:

\( (b, um : O[11]) \) and \( (a, um : O[12]) \).
Hence, there is no repeated attacks to these two assumptions.

Actual arguments drawn from the dialogue in Example 26 are: 
\{s\} ⊢ t, \{q\} ⊢ t, \{p\} ⊢ t, \{b\} ⊢ t, and \{\} ⊢ t. Notice that there are two arguments \{q\} ⊢ t and \{p\} ⊢ t for the same claim \(a\), as both \(rl(a ← q)\) and \(rl(a ← p)\) are in \(δ\). Even though the debate tree drawn from this dialogue (shown in Figure 4.9) is similar to the one in Figure 4.7, this dialogue is still focused as these two rules represent two different ways of attacking the claim, rather than supporting / proving the claim.

We refine the outcome function guaranteeing that the “proponent” has the last word in the dialogue, namely all leaves in the debate tree are proponent nodes or “dead-end” opponent nodes (not corresponding to any unattacked actual arguments).

**Definition 27.** The last word outcome function \(ω_{lw}\) is such that, given \(δ \in D\) such that \(δ\) is compatible with a grounded or admissible legal-move function \(λ\), and the debate tree, \(T(δ)\), drawn from \(δ\), then \(ω_{lw}(δ, λ) = true\) if and only if \(ω_{ex}(δ, λ) = true\) and one of the following two cases holds:

1. for all leaf nodes \(n\) in \(T(δ)\), \(n\) is either \(\_mr : P[\_]\) or \(\_ma : P[\_]\);
2. if a leaf node $n$ is of the form $(\ldots, \cdot : O[\cdot])$, then either
   a) $n$ is in a potential argument, or
   b) $n$ is in an actual argument that contains one node $n'$ of the form $(\alpha, ma : O[\cdot])$ such that there is another node $n''$ in $T(\delta)$ of the form $(\alpha, ma : O[\cdot]), n'' \neq n$, and $n''$ is properly attacked.

We refer to exhaustive dialogues for which $\omega_{lw}$ is true as positive.

The last word outcome function specifies a winning condition for the (fictitious) proponent such that: either the proponent finishes the dialogue with rules and unattacked assumptions (condition 1), or the (fictitious) opponent does not pose any valid attacks via actual argument (condition 2a), or any valid attacks posed by the opponent has been answered with valid counter attacks (condition 2b).

Example 28. This example illustrates a non-positive dialogue $\delta$.

<table>
<thead>
<tr>
<th>$a_1$</th>
<th>$a_2$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$(a_1, a_2, 0, claim(s), 1)$</td>
<td>$(a_2, a_1, 1, rl(s \leftarrow a, b), 2)$</td>
</tr>
<tr>
<td>$(a_1, a_2, 0, \pi, 3)$</td>
<td>$(a_2, a_1, 0, \pi, 4)$</td>
</tr>
</tbody>
</table>

The debate tree $T(\delta)$ drawn from this example is in Figure 4.11. We can see this dialogue is not positive as there are two unmarked leaf nodes:

$$(a, um : P[2])$$ and $$(b, um : P[2])$$,

in $T(\delta)$.

Figure 4.11: A non-positive dialogue in Example 28.

The dialogues in Examples 9 and 26 are positive. The dialogue in Example 28 is not. Positive dialogues give debate trees corresponding to abstract dispute trees [DKT06] (see Chapter 2) with the same defence set and culprits. Formally:

Lemma 4. Given a positive dialogue $\delta$ compatible with some grounded legal-move function $\lambda_{GND}$, let $T(\delta)$ be the debate tree drawn from $\delta$ and $\chi$ be the sentence in the root node of $T(\delta)$. Then there is an abstract dispute tree $T^a$ for $S \vdash \chi$ for some $S$, such that $\mathcal{DEF}(T(\delta)) = \mathcal{DEF}(T^a)$ and $\mathcal{CUL}(T(\delta)) = \mathcal{CUL}(T^a)$. 

73
**Proof.** We can transform debate trees into abstract dispute trees. Given a debate tree \( T(\delta) \), its equivalent abstract dispute tree \( T^a \) is constructed as follows.

1. Delete all nodes \( n \) from \( T(\delta) \) where \( n \) is in a potential argument. Let \( T'(\delta) \) be the result.

2. Modify \( T'(\delta) \) by appending a new flag field \( Z = \{0, 1\} \) to each remaining node in \( T(\delta) \) and initialise \( Z \) to 0 for all nodes, i.e., a node now looks like \( \langle \_ , \_ : \_ , \_ \rangle[0] \). Let \( T''(\delta) \) be the result.

3. \( T^a \) is \( T^a_m \) in the sequence \( T^a_1, \ldots, T^a_m \) constructed inductively as follows:
   - \( T^a_0 \) is empty;
   - let \( A \) be the argument in \( T''(\delta) \) that contains the root of \( T(\delta) \). Set the flags of all nodes in \( T''(\delta) \) that are also in \( A \) to 1. Let \( T^a_1 \) be the result; then \( T^a_1 \) contains a single node that is labelled by \( A \) and is a P node;
   - let \( T^a_i \) be the \( i \)-th tree, for \( 0 < i < m \), then \( T^a_{i+1} \) is \( T^a_i \) with an additional node \( (B, L) \), where \( B \) is an argument drawn from \( T''(\delta) \), child of \( B' \), another argument drawn from \( T''(\delta) \), such that:
     - the flag of at least one node in \( B \) is 0;
     - the root node of \( B \) has a parent node \( p \) in \( T''(\delta) \) with flag equal to 1 and such that \( p \) is in \( B' \);
     - \( L \) is P if the root node of \( B \) is a proponent node, otherwise \( L \) is O;
     - set the flags of all nodes in \( T''(\delta) \) that are also in \( B \) to 1, let \( T''_{i+1}(\delta) \) be the result.
4. \( m \) is the smallest index such that there is no node in \( T''_m(\delta) \) with its flag equal to 0.

\( T^a \) constructed above is an abstract dispute tree as follows.

1. Every node of \( T^a = T^a_m \) contains a single argument as there is no potential argument in \( T'(\delta) \) and \( T''(\delta) \). For each argument, there is a unique node in \( T^a \). Each node in \( T^a \) is labelled either \( P \) or \( O \) as arguments drawn from \( T(\delta) \) are labelled either \( P \) or \( O \).
2. The root node of \( T^a \) contains the argument that has the claim of the dialogue. The root is labelled \( P \) by construction of \( T(\delta) \).
3. By definition of exhaustive outcome function, since \( \delta \) is exhaustive, every assumption is attacked in as many ways as possible. Hence a \( P \) node in \( T^a \) has as many children as its attacks by actual arguments.

4. By the definition of patient and focused legal-move function, since \( \delta \) is patient and focused, there is only one way of attacking an argument labelled by a \( O \) node. By the last word outcome function, since \( \delta \) is positive, there is no un-attacked argument labelled by a \( O \) node. Therefore, every \( O \) node in \( T^a \) with an assumption has one and only one \( P \) node as its child.

Since \( T^a \) contains the same actual arguments as \( T(\delta) \) and arguments have the same P/O labelling in both \( T^a \) and \( T(\delta) \), we have \( D E F(T(\delta)) = D E F(T^a) \) and \( C U L(T(\delta)) = C U L(T^a) \).

Lemma 4 shows the connection between the debate trees drawn from dialogues compatible with \( \lambda_{GND} \) with abstract dispute trees. Similar result can be obtained for dialogues compatible with some admissible legal-move function \( \lambda_{ADM} \). Since \( \lambda_{fi} \) performs the “filtering”, this filtering can be retracted by inserting nodes that have been filtered back to a debate tree. We introduce the expanded debate tree as follows.

**Definition 28.** Given a debate tree \( T(\delta) \) where \( \delta \) is positive and compatible with some \( \lambda_{ADM} \), we can construct a (possibly infinite) sequence \( T^E_0, \ldots, T^E_i, \ldots \) of trees such that:

- Delete all nodes \( n \) from \( T(\delta) \) where \( n \) is in a potential argument. Let \( T'(\delta) \) be the result.
- \( T^E_0 = T'(\delta) \).
- Suppose \( T^E_i \), for \( i \geq 0 \), has been constructed; then \( T^E_{i+1} \) is obtained by adding arguments \( T_1, \ldots, T_k \) simultaneously to leaf nodes \( n_1, \ldots, n_k \) of \( T^E_i \), as children, respectively, where \( T_j \) and \( n_j, 1 \leq j \leq k \) are such that:
  1. \( n_j \) is of the form \( (\alpha, ma : L_0[\_]) \) (\( L_0 \in \{P,O\} \)),
  2. there is \( n^0_j \) of the form \( (\alpha, ma : \_[\_]) \) in \( T'(\delta) \), \( n^0_j \neq n_j \),
  3. \( n^0_j \) has a child \( n'_j \) in \( T'(\delta) \), and \( n'_j \) is in an actual argument \( T''_j \) in \( T'(\delta) \).
  4. modify all nodes \( (\beta, F : \_[i\_]) \) in \( T''_j \) to \( (\beta, F : P[i\_]) \) if \( L_0 = O \), or \( (\beta, F : O[i\_]) \) if \( L_0 = P \); let the result be \( T_j \).
The expanded debate tree $T^E(\delta)$ of $T(\delta)$ is the limit\(^2\) of this sequence. Note that, $T^E(\delta)$ of $T(\delta)$ is the last element of this sequence, if the sequence is finite (when no leaf node is an assumption node).

**Example 29.** We illustrate the notion of the expanded debate tree with the debate tree given in Figure 4.10. The expanded debate tree is shown in Figure 4.12.

![Figure 4.12: The expanded debate tree of the debate tree shown in Figure 4.10.](image)

With the notion of the expanded debate tree defined, we have results for dialogues that are compatible with $\lambda_{ADM}$ as well.

**Lemma 5.** Given a positive dialogue $\delta$ compatible with some admissible legal-move function $\lambda_{ADM}$, let $T(\delta)$ be the debate tree drawn from $\delta$ and $\chi$ be the sentence in the root node of $T(\delta)$. Then there is an abstract dispute tree $T^a$ for $S \vdash \chi$ for some $S$, such that $DEF(T(\delta)) = DEF(T^a)$ and $CUL(T(\delta)) = CUL(T^a)$.

**Proof.** We can transform debate trees into abstract dispute trees using the procedure as shown in the proof of Lemma 4 with a modification. After deleting all nodes $n$ from $T(\delta)$ where $n$ is in a potential argument (step 1), we replace $T(\delta)$ with $T^E(\delta)$, the expanded debate tree of $T(\delta)$. The rest of the construction of $T^a$ remain unchanged.

\(^2\)The limit of a sequence of debate trees is a (possibly infinite) tree $T$ such that every tree in the sequence is a top-portion of $T$, and every finite top-portion of $T$ is a sub-tree of some tree in the sequence.
It is easy to see that $T^a$ constructed with the modified $T^E(\delta)$ is an abstract dispute tree and $\mathcal{DF}(T(\delta)) = \mathcal{DF}(T^a)$ and $\mathcal{UL}(T(\delta)) = \mathcal{UL}(T^a)$, as shown in the proof of Lemma 4.

As in the case of abstract dispute trees, the defence set of a debate tree may not be admissible, as it may attack itself. We refine the notion of outcome function by enforcing that this set does not attack itself, as follows:

**Definition 29.** The successful outcome function $\omega_{scc}$ is such that, given $\delta \in D$, compatible with some admissible legal-move function $\lambda_{ADM}$ or grounded legal-move function $\lambda_{GND}$, $\omega_{scc}(\delta, \lambda) = true$ if and only if $\omega_{lw}(\delta, \lambda) = true$ and $\mathcal{DF}(T(\delta)) \cap \mathcal{UL}(T(\delta)) = \{\}$.

**Theorem 1.** Given a dialogue $D_{a_i}(\chi) = \delta \in D$ compatible with some admissible legal-move function $\lambda_{ADM}$, if $\omega_{scc}(\delta, \lambda_{ADM}) = true$, then $\delta$ is a-successful and $\chi$ is supported by $\mathcal{DF}(T(\delta))$.

**Proof.** If $\omega_{scc}(\delta, \lambda) = true$, by Lemma 5 there exists an abstract dispute tree $T^a$ such that $\mathcal{DF}(T(\delta)) = \mathcal{DF}(T^a)$ and $\mathcal{UL}(T(\delta)) = \mathcal{UL}(T^a)$. By Theorem 5.1 of [DKT06] (see Chapter 2), the theorem holds.

**Theorem 2.** Given a dialogue $D_{a_j}(\chi) = \delta \in D$ compatible with some $\lambda_{GND}$, if $\omega_{scc}(\delta, \lambda_{GND}) = true$, then $\delta$ is g-successful and $\chi$ is supported by $\mathcal{DF}(T(\delta))$.

**Proof.** Similar to the previous proof, if $\omega_{scc}(\delta, \lambda) = true$, then there exists an abstract dispute tree $T^a$ such that $\mathcal{DF}(T(\delta)) = \mathcal{DF}(T^a)$ and $\mathcal{UL}(T(\delta)) = \mathcal{UL}(T^a)$. As shown in [DKT09], as a direct consequence of Theorem 3.7 in [KT99], we have the defence set of a grounded abstract dispute tree is a subset of the grounded set of arguments. Hence $\delta$ is g-successful and $\chi$ is supported by $\mathcal{DF}(T(\delta))$.

**4.3.1 One-way Expansion Dialogue**

We also introduce a special class of focused dialogues compatible with one-way expansion legal-move functions. Dialogues compatible with one-way expansion legal-move functions are simpler than focused dialogues. We show how such dialogues can be used to model a simplified instance of discovery dialogues in Section 7.4.
Definition 30. A one-way expansion legal-move function, $\lambda \in \Lambda$, is such that, for $\delta \in \mathcal{D}$, for all $u = \langle \_, \_, C, \_ \rangle \in \lambda(\delta)$, then:

1. $C = rl(\rho)$ only if $\exists \ u' = \langle \_, \_, rl(\rho'), \_ \rangle \in \delta$ such that $\text{Head}(\rho) = \text{Head}(\rho')$;

2. $C = ctr(\alpha, \beta)$ only if $\exists \ u' = \langle \_, \_, ctr(\alpha, \beta'), \_ \rangle \in \delta$ such that $\beta = \beta'$.

We use $\lambda_{\text{owe}}$ to denote a generic one-way expansion legal-move function.

A one-way expansion dialogue is a dialogue compatible with some $\lambda_{\text{owe}}$.

This definition implies that, in a single dialogue (1) there is only one unique way of expanding a rule; and (2) every assumption has one and only one contrary. Although condition (2) is not very restrictive (indeed it is imposed in the original ABA), condition (1) imposes a restrictive limitation on ABA framework drawn from dialogues, as illustrated by the following example.

Example 30. The following dialogue is a one-way expansion dialogue.

<table>
<thead>
<tr>
<th>$a_1$</th>
<th>$a_2$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\langle a_1, a_2, 0, \text{claim}(s), 1 \rangle$</td>
<td>$\langle a_2, a_1, 1, \text{asm}(s), 2 \rangle$</td>
</tr>
<tr>
<td>$\langle a_1, a_2, 2, \text{ctr}(s, a), 3 \rangle$</td>
<td>$\langle a_2, a_1, 3, rl(a \leftarrow b), 4 \rangle$</td>
</tr>
</tbody>
</table>

The following dialogue is not a one-way expansion dialogue, as there are two rules $rl(a \leftarrow b)$ and $rl(a \leftarrow c)$ in this dialogue, though it is focused.

<table>
<thead>
<tr>
<th>$a_1$</th>
<th>$a_2$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\langle a_1, a_2, 0, \text{claim}(s), 1 \rangle$</td>
<td>$\langle a_2, a_1, 1, \text{asm}(s), 2 \rangle$</td>
</tr>
<tr>
<td>$\langle a_1, a_2, 2, \text{ctr}(s, a), 3 \rangle$</td>
<td>$\langle a_2, a_1, 3, rl(a \leftarrow b), 4 \rangle$</td>
</tr>
<tr>
<td>$\langle a_2, a_1, 3, rl(a \leftarrow c), 5 \rangle$</td>
<td></td>
</tr>
</tbody>
</table>

As we can see from Definition 30, one-way expansion legal-move functions are (special cases of) focused legal-move functions. Hence we have the following lemma.

Lemma 6. Given a dialogue $\delta$, for any one-way expansion legal-move function $\lambda_{\text{owe}}$, if $\lambda_{\text{owe}}(\delta) = U$, then there is a focused legal-move function $\lambda_f$ such that $U \subseteq \lambda_f(\delta)$.
Definition 31. Given a flat legal-move function $\lambda_{fl}$, a related legal-move function $\lambda_{rt}$, a patient legal-move function $\lambda_p$, a focused legal-move function $\lambda_f$, and a filtered legal-move $\lambda_{fi}$,

- a (one-way) admissible legal-move function is defined as:
  \[ \lambda = \lambda_{fl} \cdot \lambda_{rt} \cdot \lambda_p \cdot \lambda_{owe} \cdot \lambda_{fi}; \]
- a (one-way) grounded legal-move function is defined as:
  \[ \lambda = \lambda_{fl} \cdot \lambda_{rt} \cdot \lambda_p \cdot \lambda_{owe}. \]

We use $\lambda_{ADM}$ and $\lambda_{GND}$ to denote generic (one-way) admissible and grounded legal-move functions, respectively.

Since one-way expansion functions are special cases of focused legal-move functions, given Lemma 6, we obtain results for one-way expansion dialogues similar to Theorems 1 and 2 as follows.

**Corollary 1.** Given a dialogue $\mathcal{D}_{a_j}(\chi) = \delta \in \mathcal{D}$ compatible with an (one-way) admissible legal-move function $\lambda_{ADM}$, if $\omega_{scc}(\delta, \lambda_{ADM}) = true$, then $\delta$ is a-successful and the $\chi$ is supported by $\mathcal{D}\mathcal{E}\mathcal{F}(\mathcal{T}(\delta))$.

**Corollary 2.** Given a dialogue $\mathcal{D}_{a_j}(\chi) = \delta \in \mathcal{D}$ compatible with an (one-way) grounded legal-move function $\lambda_{GND}$, if $\omega_{scc}(\delta, \lambda_{GND}) = true$, then $\delta$ is g-successful and the $\chi$ is supported by $\mathcal{D}\mathcal{E}\mathcal{F}(\mathcal{T}(\delta))$.

4.4 Dialogues and Debate Forests

In this section we lift dialogue constraints and consider dialogues that may not be compatible with a focused legal-move function. Hence, we consider dialogues that allow multiple ways of (directly or indirectly) supporting the claim. In this unconstrained case, we again show that Theorems 1 and 2 hold for such dialogues.

We first define two more legal-move functions.

**Definition 32.** Given a flat legal-move function $\lambda_{fl}$, a related legal-move function $\lambda_{rt}$, a patient legal-move function $\lambda_p$, and a filtered legal-move $\lambda_{fi}$.
• an (unconstrained) admissible legal-move function is defined as:

\[ \lambda = \lambda_{fl} \cdot \lambda_{rt} \cdot \lambda_{p} \cdot \lambda_{fl}; \]

• an (unconstrained) grounded legal-move function is defined as:

\[ \lambda = \lambda_{fl} \cdot \lambda_{rt} \cdot \lambda_{p}. \]

We use \( \lambda^*_{ADM} \) and \( \lambda^*_{GND} \) to denote generic (unconstrained) admissible and grounded legal-move functions, respectively.

Unconstrained admissible and grounded legal-move functions are used to construct non-focused dialogues that are admissible and grounded, respectively. We show results of using these legal-move functions in later part of this section.

We first show an example of a dialogue compatible with some grounded legal-move function \( \lambda^*_{GND} \) (and some admissible legal-move function \( \lambda^*_{ADM} \) as well).

Example 31. A sample dialogue compatible with \( \lambda^*_{GND} \) (but not \( \lambda_{GND} \)) is:

\[
\begin{array}{c|c}
\text{a}_1 & \text{a}_2 \\
\hline
\langle \text{a}_1, \text{a}_2, 0, \text{claim}(s), 1 \rangle & \langle \text{a}_2, \text{a}_1, 1, \text{rl}(s \leftarrow p), 2 \rangle \\
\langle \text{a}_1, \text{a}_2, 1, \text{rl}(s \leftarrow q), 3 \rangle & \langle \text{a}_2, \text{a}_1, 3, \text{rl}(q \leftarrow a), 4 \rangle \\
\langle \text{a}_1, \text{a}_2, 2, \text{rl}(p \leftarrow b, c), 5 \rangle & \langle \text{a}_2, \text{a}_1, 4, \text{asm}(a), 6 \rangle \\
\langle \text{a}_1, \text{a}_2, 6, \text{ctr}(a, r), 7 \rangle & \langle \text{a}_2, \text{a}_1, 7, \text{rl}(r \leftarrow), 8 \rangle \\
\langle \text{a}_1, \text{a}_2, 5, \text{asm}(b), 9 \rangle & \langle \text{a}_2, \text{a}_1, 5, \text{rl}(c \leftarrow), 10 \rangle \\
\langle \text{a}_1, \text{a}_2, 9, \text{ctr}(b, k), 11 \rangle & \langle \text{a}_2, \text{a}_1, 0, \pi, 12 \rangle \\
\langle \text{a}_1, \text{a}_2, 0, \pi, 13 \rangle & \\
\end{array}
\]

This dialogue is not compatible with a focused legal-move function (hence not compatible with \( \lambda_{GND} \)) as both utterance \( \langle \text{a}_2, \text{a}_1, 1, \text{rl}(s \leftarrow p), 2 \rangle \) and \( \langle \text{a}_1, \text{a}_2, 1, \text{rl}(s \leftarrow q), 3 \rangle \) expand \( s \). The debate tree drawn from this dialogue is shown in Figure 4.13. We can see that this tree is not focused.

We first introduce the notion of arguments (in the form of nodes in a debate tree) being attacked as follows.
Definition 33. Given a debate tree $T$ that contains an argument $A$, which is a sub-tree of $T$, we say that $A$ is attacked in $T$ if and only if there is a node $n = (\_, ma : \_[])$ in $A$ such that $n$ has a child node $m$ in $T$.

We say that the sub-tree rooted at $m$ in $T$ is an attacker of $A$ in $T$.

For instance, the argument $\{s\} \vdash t$ in Figure 4.9 is attacked; and the attacker of $\{s\} \vdash t$ is the tree shown in Figure 4.14 (right). The notion of attacked differs from the one given in Definition 23 in two ways:

- Definition 33 defines the attacked notion with respect to an argument, or a sub-tree of a debate tree, whereas Definition 23 defines the properly attacked with respect to a single (assumption) node in a debate tree;

- Definition 33 states an argument is attacked as long as there is some node in the debate tree such that the node is a child of the argument, whereas Definition 23 defines an assumption node being properly attacked if and only if there is an actual argument that hangs from that assumption node.

Debate trees are insufficient to represent information disclosed in non-focused dialogues. To compute the acceptability of the claim of non-focused dialogues, we introduce the notion of debate forest, composed of (debate) trees.
Definition 34. Given a dialogue \( D_{a_j}^{u_i}(\chi) = \delta = (u_1, \ldots, u_m) \), the debate forest \( F(\delta) \) drawn from \( \delta \) is a set of trees. \( F(\delta) \) is \( F^m(\delta) \) in the sequence \( F^0(\delta), F^1(\delta), \ldots, F^m(\delta) \) constructed inductively from the \( \pi \)-pruned sequence \( \delta' = (u'_1, \ldots, u'_m) \) obtained from \( \delta \), as follows (below, \( L, L' \in \{P,O\}, L \neq L' \)):

1. \( F^0(\delta) \) is empty.

2. \( F^1(\delta) \) contains a single debate tree \( T_1^1(\delta) \), which contains a single node \((s, um : P[1])\).

3. Let \( F^i(\delta) \) be the \( i \)-th forest containing trees \( T_1^i(\delta), \ldots, T_{l(i)}^i(\delta) \), let \( u_{i+1}' = \langle \omega, - t, C, id \rangle \), and let \( u_i' = \langle \omega, - \omega C, t \rangle \) be the target utterance of \( u_{i+1}' \); then \( F^{i+1}(\delta) \) is obtained as follows.
   a) if \( C = r l(\beta_0 \leftarrow \beta_1, \ldots \beta_l) \) then \( F^{i+1}(\delta) \) is obtained in one of the following two cases:
      i. if there is no debate tree in \( F^i(\delta) \) that contains \((\beta_0, mr : P[t])\), then \( F^{i+1}(\delta) \) is \( F^i(\delta) \) updated as follows: for all \( T_j^i(\delta), 0 < j \leq l \), in \( F^i(\delta) \) such that \( T_j^i(\delta) \) contains \((\beta_0, - \omega : L[t])\), then, for each \( T_j^i(\delta) \),
         \[ (\beta_1, um : L[i + 1]), \ldots, (\beta_l, um : L[i + 1]) \]
      are added to \( T_j^i(\delta) \) as children of \((\beta_0, - \omega : L[t])\); and if \((\beta_0, - \omega : L[t])\) is \((\beta_0, um : L[t])\), then it is replaced by \((\beta_0, mr : L[t])\) in each \( T_j^i(\delta) \) that contains \((\beta_0, um : L[t])\);
      ii. otherwise, \( F^{i+1}(\delta) \) is \( F^i(\delta) \) with \( k \) additional debate trees, where \( k \) is the number of debate trees in \( F^i(\delta) \) that contains \((\beta_0, mr : P[t])\). For each \( T_j^i(\delta) \) that contains \((\beta_0, mr : P[t])\), a new debate tree is created by copying \( T_j^i(\delta) \) and replacing all children and sub-trees rooted at these children of \((\beta_0, mr : P[t])\) with new children
         \[ (\beta_1, um : P[i + 1]), \ldots, (\beta_l, um : P[i + 1]) \];
   b) if \( C = asm(\alpha) \) then \( F^{i+1}(\delta) \) is \( F^i(\delta) \) with all \( T_j^i(\delta) \) that contain
      \((\alpha, um : L[t])\) with \((\alpha, um : L[t])\) replaced by \((\alpha, ma : L[i + 1])\).
c) if \( C = ctr(\alpha, \beta) \) then \( F^{i+1}(\delta) \) is obtained in one of the following two cases:

i. if there exists no \( T_{ij}(\delta) \) in \( F^i(\delta) \) such that \( T_{ij}(\delta) \) contain a node
\( \bar{n} = (\alpha, ma : O[t]) \) where \( \bar{n} \) is in an argument \( A \) such that \( A \) is attacked, then \( F^{i+1}(\delta) \) is \( F^i(\delta) \) with all \( T_{ij}(\delta) \), \( 0 < j \leq l \), that contain \( (\alpha, ma : O[t]) \) each having a new node \( (\beta, um : L'[i+1]) \) as a child of \( (\alpha, ma : L[t]) \);

ii. otherwise, \( F^{i+1}(\delta) \) is \( F^i(\delta) \) with additional \( k \) debate trees, where \( k \) is the number of debate trees in \( F^i(\delta) \) that contain \( (\alpha, ma : O[t]) \). For each \( T^j_i(\delta) \) that contains \( (\alpha, ma : O[t]) \), a new debate tree is created by (1) copying \( T^j_i(\delta) \), (2) removing the attacker of the argument that contains \( (\alpha, ma : O[t]) \), and (3) adding \( (\beta, um : P[i+1]) \) as a child of \( (\alpha, ma : O[t]) \).

The construction of the debate forest drawn from the dialogue in Example 31 is in Figures 4.15 and 4.16. Conceptually, we construct debate forests using the same procedure as constructing debate trees, i.e., inserting new nodes and locating parents, etc. However, if we see a new utterance \( u \) such that adding \( u \) will yield a non-focused dialogue, we will (1) duplicate all existing trees in the forest that contain the target of \( u \) and (2) removing all sub-trees that are connected to the target of \( u \) from the duplications, and (3) add \( u \) to the duplicated trees. It is possible that more trees are duplicated in this process than needed but the forest is a set, so only a single version of the identical trees is kept. Hence, at the end, we obtain a set of unique debate trees in a debate forest.

To study properties of debate forests, we introduce the concept of sub-dialogues.

**Definition 35.** Given a dialogue \( \mathcal{D}^\chi_u(\delta) = \delta, \mathcal{D}^\chi_u(\delta)' = \delta' \) is a sub-dialogue of \( \delta \) if and only if for all utterances \( u \) in \( \delta' \), \( u \) is in \( \delta \). We say that \( \delta \) is the full-dialogue of \( \delta' \).

Conceptually, given a dialogue \( \delta \), a sub-dialogue of \( \delta \) contains a subset of utterances that are in \( \delta \). As shown later, given a dialogue \( \delta \), we want to identify some sub-dialogues \( \delta' \)'s of \( \delta \), such that each \( \delta' \) is focused and share the same claim as \( \delta \). Hence, the computation on \( \delta \) can be divided to separate computation on each \( \delta' \).

**Example 32.** Two sub-dialogues of the dialogue in Example 31 are in tables 4.3 and 4.4.
Figure 4.15: The construction of the debate forest in Example 31 (Part 1).
Figure 4.16: The construction of the debate forest in Example 31 (Part 2).
Table 4.3: A sub-dialogue of the dialogue in Example 31 (part 1).

<table>
<thead>
<tr>
<th>$a_1$</th>
<th>$a_2$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$(a_1,a_2,0,claim(s),1)$</td>
<td>$(a_2,a_1,3,rl(q ← a),4)$</td>
</tr>
<tr>
<td>$(a_1,a_2,1,rl(s ← q),3)$</td>
<td>$(a_2,a_1,4,asm(a),6)$</td>
</tr>
<tr>
<td>$(a_1,a_2,6,ctr(a,r),7)$</td>
<td>$(a_2,a_1,7,rl(r ←),8)$</td>
</tr>
</tbody>
</table>

Table 4.4: A sub-dialogue of the dialogue in Example 31 (part 2).

<table>
<thead>
<tr>
<th>$a_1$</th>
<th>$a_2$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$(a_1,a_2,0,claim(s),1)$</td>
<td>$(a_2,a_1,1,rl(s ← p),2)$</td>
</tr>
<tr>
<td>$(a_1,a_2,2,rl(p ← b,c),5)$</td>
<td>$(a_2,a_1,5,rl(c ←),10)$</td>
</tr>
<tr>
<td>$(a_1,a_2,5,asm(b),9)$</td>
<td></td>
</tr>
<tr>
<td>$(a_1,a_2,9,ctr(b,k),11)$</td>
<td></td>
</tr>
</tbody>
</table>

The first sub-dialogue (in Table 4.3) is neither a-successful nor g-successful as the proponent fails to defend $a$, an assumption in the argument $\{a\} ⊢ s$.

The second sub-dialogue (in Table 4.4) is both a-successful and g-successful as the proponent is able to construct the argument $\{b\} ⊢ s$ and defend it.

Note that Definition 35 gives a very liberal definition of sub-dialogues. We will show how such definition can be used to prove our results. Conceptually, what happens there is that we show each non-focused dialogue can be understood as a “union” of several independent focused sub-dialogues. Hence, each sub-dialogue draws a tree in the debate forest drawn from the full non-focused dialogue. Thus, as soon as one debate tree is found to be “good”, then the particular sub-dialogue draws the debate tree becomes “successful”, then the full non-focused dialogue is “successful”. Essentially, we reduce the computation on a non-focused dialogue to many separate independent computation on sub-dialogues.

The following results (Lemmas 7, 8 and 9) connect debate forest with debate trees.

**Lemma 7.** Given a debate forest $F(δ)$ drawn from $δ$ compatible with $λ$ is some grounded legal-move function $λ^*_GND$, or some admissible legal-move function $λ^*_ADM$, each tree in $F(δ)$ is a debate tree.

**Proof.** We show that given a dialogue $δ$, the inductive process in Definition 34
constructs a set of debate trees.

For each tree $T^i_j(\delta) \in F^i(\delta)$, condition 1(a) and 1(b) in Definition 15 are trivially true as all nodes in each tree are inserted in accordance with utterances in $\delta$ hence if a sentence $\beta$ is in a node, $\beta$ must be in some utterance in the dialogue.

Condition 2(a)-2(f) in Definition 15 are met by condition 3(a)-3(c) in Definition 34.

Hence, each tree in a debate forest is a debate tree per Definition 15.

We redefine patient and filtered legal-move functions with respect to debate forests, as follows.

**Definition 36.** A debate forest $F(\delta)$ is patient if and only if all debate trees in $F(\delta)$ are patient debate trees.

**Definition 37.** A debate forest $F(\delta)$ is filtered if and only if all debate trees in $F(\delta)$ are filtered debate trees.

**Definition 38.** A legal-move function $\lambda_p \in \Lambda$ is patient if and only if for every $\delta \in D$ such that $F(\delta)$ is patient, for every $u \in \lambda(\delta)$, $F(\delta \circ u)$ is still patient.

**Definition 39.** A legal-move function $\lambda_{fi} \in \Lambda$ is filtered if and only if for every $\delta \in D$ such that $F(\delta)$ is filtered for every $u \in \lambda(\delta)$, $F(\delta \circ u)$ is still filtered.

The latter two definitions are generalised versions of Definition 18 and 20 respectively as a debate tree is a special case of a debate forest that contains a single tree.

**Lemma 8.** Given a debate forest $F(\delta)$ drawn from a dialogue $\delta$, if $\delta$ is compatible with some grounded legal-move function $\lambda^*_{GND}$ or some admissible legal-move function $\lambda^*_{ADM}$, then each debate tree in $F(\delta)$ is patient and focused.

**Proof.** Since dialogues constructed with $\lambda^*_{GND}$ or $\lambda^*_{ADM}$ are patient, and by Definition 38, we know that each debate tree in a debate forest is patient, we only need to show that each debate tree is also focused.

Definition 24 defines two conditions for debate trees to be focused. Condition (a) in Definition 24 is met by 3(c) in Definition 34 as which ensures that in a single tree, a proponent argument is only attacked by a single opponent argument. Condition (b) in Definition 24 is met by 3(a) in Definition 34 as which ensures that when a proponent node is expanded such that it has a set of proponent children, then these children must be from a single utterance.
Lemma 9. Given a debate forest $\mathcal{F}(\delta)$ drawn from $\delta$, if $\delta$ is compatible with some grounded legal-move function $\lambda^*_GND$ or some admissible legal-move function $\lambda^*_ADM$, then each tree in $\mathcal{F}(\delta)$ is a debate tree drawn from a sub-dialogue of $\delta$, such that each of these sub-dialogue is compatible with $\lambda^*_GND$ or $\lambda^*_ADM$, respectively.

Proof. We prove this lemma by constructing sub-dialogues and showing each of the tree in a debate forest is a debate tree drawn from a sub-dialogue compatible with $\lambda^*_GND$ or $\lambda^*_ADM$.

Given $\mathcal{F}(\delta)$ contains $l$ debate trees $T_1(\delta), \ldots, T_l(\delta)$, the sub-dialogue $\delta_i, 0 < i \leq l$ that draws the debate tree $T_i(\delta)$ is constructed as follows.

1. $\delta_i$ is initialised to empty.
2. For each node $(\beta, F : _{[\text{id}]}) = n$ in $T(\delta_i)$,
   - if $u_{\text{id}} = (\_, \_, t, \_, \text{id})$ is in $\delta$ but not in $\delta_i$, then add $u_{\text{id}}$ to $\delta_i$;
   - let $u_t$ be the utterance in $\delta$ such that $u_{\text{id}}$ is related to $u_t$, if $u_t$ is not in $\delta_i$, then add $u_t$ to $\delta_i$.
3. Sort $\delta_i$ in the order of utterance IDs.

It is easy to see $\delta_i$ constructed above are sub-dialogues of $\delta$ and $T_i(\delta)$ is drawn from $\delta_i$. It is also trivially true that $\delta_i$ is flat, given $\delta$ is. We now show $\delta_i$ is related.

By Definition 15, we know in a debate tree, there are 6 cases that two nodes $n$ and $n'$ can be connected ($n$ is the parent of $n'$):

1. $n = (\_, mr : L[\text{id}])$ and $n' = (\_, mr : L'[\text{id}'])$, or
2. $n = (\_, mr : L[\text{id}])$ and $n' = (\_, ma : L[\text{id}'])$, or
3. $n = (\_, mr : L[\text{id}])$ and $n' = (\_, um : L[\text{id}'])$, or
4. $n = (\_, ma : L[\text{id}])$ and $n' = (\_, mr : L'[\text{id}'])$, or
5. $n = (\_, ma : L[\text{id}])$ and $n' = (\_, ma : L'[\text{id}'])$, or
6. $n = (\_, ma : L[\text{id}])$ and $n' = (\_, um : L'[\text{id}'])$,

in which $L, L' \in \{P, O\}, L \neq L', \text{id}, \text{id}' \in ID \setminus \{ID_0\}$. By Definition 11, the utterance $u_{\text{id}'} = (\_, \_, \_, \_, t, \_i)$ is related to the utterance $u_{\text{id}} = (\_, \_, \_, \_, \text{id})$ in cases 1, 3, 4 and 6. In cases 2 and 5, there is an utterance $u_t = (\_, \_, \text{id}, asm(\_, t), \_)$,
such that \( u_t \) is related to \( u_{id} \) and \( u_{id'} \) is related to \( u_t \). It can be seen that the previous construction of \( \delta_i \) includes all utterances \( u_{id}, u_{id'} \) and \( u_t \) but no other utterance. Hence \( \delta_i \) is related.

With the above results and Lemma 8, this lemma holds.

We prove results for dialogues compatible with some \( \lambda_{GND}^* \) and \( \lambda_{ADM}^* \), which generalise results using \( \lambda_{GND}^- \) and \( \lambda_{ADM}^- \) given earlier.

**Theorem 3.** Given a dialogue \( D_{a_i a_j}(\beta) = \delta \) compatible with some \( \lambda_{GND}^* \), \( \delta \) is g-successful if there is a debate tree \( T(\delta_i) \) drawn from a sub-dialogue \( \delta_i \) of \( \delta \) such that \( \delta \) is compatible with \( \lambda_{GND}^* \) and \( \omega_{scc}(\delta_i, \lambda_{GND}) = true \).

**Proof.** Let \( F(\delta) \) be the debate forest drawn from \( \delta \) and \( F(\delta) \) contains debate trees \( T(\delta_1), \ldots, T(\delta_n) \). Given Lemma 9, we know all trees in \( F(\delta) \) are debate tree drawn from some sub-dialogue of \( \delta \) and each sub-dialogue is compatible with \( \lambda_{GND}^* \).

Suppose \( \delta_i \) is compatible with \( \lambda_{GND}^* \) and \( \omega_{scc}(\delta_i, \lambda_{GND}) = true \), then \( \delta_i \) is g-successful and \( \beta \) is grounded in \( AF_i \), the ABA framework drawn from \( \delta_i \). We need to show \( \beta \) is also grounded in \( A\mathcal{F} \), the ABA framework drawn from \( \delta \).

Since \( \delta_i \) is compatible with \( \lambda_{GND}^* \) and \( \omega_{scc}(\delta_i, \lambda_{GND}) = true \), then it is not the case that there is an argument that attacks \( DEF(T(\delta_i)) \) that has not been countered attacked in the ABA framework drawn from \( \delta \). By Definition 34 each debate tree in \( F(\delta) \) represents a set of arguments that support the claim of \( \delta \). Hence each tree contains its own set of defence set, i.e., \( DEF(T(\delta_i)) \) is different from \( DEF(T(\delta_j)) \), where \( 1 \leq i, j \leq n, i \neq j \). Therefore, if a defence set of a tree is grounded in the ABA framework drawn from the sub-dialogue, it is also grounded in the ABA framework drawn from the full-dialogue. Hence \( \delta \) is g-successful if \( \delta_i \) is g-successful.

Similarly, the result holds for dialogues compatible with some \( \lambda_{ADM}^* \), as shown in the following theorem.

**Theorem 4.** Given a dialogue \( D_{a_i a_j}(\chi) = \delta \) compatible with some \( \lambda_{ADM}^* \), \( \delta \) is a-successful if there is a debate tree \( T(\delta_i) \) drawn from a sub-dialogue \( \delta_i \) of \( \delta \) such that \( \delta \) is compatible with \( \lambda_{ADM}^* \) and \( \omega_{scc}(\delta_i, \lambda_{ADM}) = true \).

Theorems 3 and 4 do not refer to debate forests drawn from dialogues directly, rather, they specify conditions of sub-dialogues of the dialogue in consideration. This is partly because the successful outcome function \( \omega_{scc} \) is defined for some
Figure 4.17: The debate tree drawn from the dialogue in Example 10.

admissible or grounded legal-move function $\lambda_{ADM}$ or $\lambda_{GND}$, but not for $\lambda^*_{ADM}$ or $\lambda^*_GND$ (see Definition 29 on Page 77). Though we could overwrite Definition 29 and redefine it for $\lambda^*_{ADM}$ and $\lambda^*_GND$, we choose not to as the current version of Theorems 3 and 4 clearly indicates that testing the acceptability of the claim of a non-focused dialogue can be reduced to testing the acceptability of the claim of sub-dialogues of the original dialogue, whereas these sub-dialogues draw debate trees and show properties we presented in the earlier section.

4.5 Illustration

We revisit Example 10 here. As we show in Chapter 3, the dialogue $\delta = D_{a1}^{a1} (\text{boy.innocent})$ is (top-down) related and flat.

The debate tree drawn from this dialogue, $T(\delta)$, is shown in Figure 4.17. $T(\delta)$ is patient, focused and filtered. Hence, $\delta$ is compatible with a flat, related, patient, focused and filtered legal-move function, $\lambda$. Hence $\delta$ is also flat, related, patient, focused and filtered.

Since $\omega_{ex}(\delta, \lambda) = \omega_{sec}(\delta, \lambda) = true$, $\delta$ is exhaustive and positive as well. Then, by Theorems 1, 2 and Proposition 1, $\delta$ is a-/g-/i-successful.

4.6 Related Work

In the previous chapter, we have mentioned that Parsons et al. [PMSW07] study utterance relevance in dialogues. We have stated that some legal-move functions
presented in this chapter bring similar effect to our dialogue model. Indeed, we can see that our patient legal-move function \((\lambda_p)\) and focused legal-move function \((\lambda_f)\) jointly resemble their R1 related-ness (every new utterance has a direct impact on the claim). Our related legal-move function \((\lambda_{rt})\) has some of the features of their R2 related-ness (every new utterance directly or indirectly impacts the claim) in that all related utterances have impact on the claim.

We have also mentioned Prakken’s work on persuasion dialogue [Pra05] in earlier chapters. In that work, Prakken maps his dialogue framework to the grounded semantics. Roughly speaking, in terms of obtaining the soundness result, i.e., a “successful” dialogue means that the claim of the dialogue is “acceptable”, there are two main differences between his approach and ours. Firstly, we rely on mapping our dialogues to abstract dispute trees whereas he uses a form of labelling. Hence, in our case, “successful” dialogues can be mapped to “good” abstract dispute trees; and in his case, “successful” dialogues have winning arguments that are labelled \(\text{in}\). Secondly, to support non-focused dialogues, we use the debate forest, and show individual trees in a forest can be mapped to abstract dispute trees, whereas he put all arguments into a single tree and then identifies a “winning strategy” that represents a sub part of the single tree.

Compared with [FT11a], the work presented in this chapter is a substantial extension. Firstly, this chapter shows soundness results with respect to grounded semantics, rather than just admissibility. Secondly, [FT11a] uses “dialectical trees”, which are mapped to \text{concrete dispute trees} [DKT06] whereas this work uses debate trees (Definition 15), which are mapped to \text{abstract dispute trees} [DKT06]. Moreover, [FT11a] define dialectical trees constructively, and this work defines debate trees (Definition 15) declaratively. The declarative definition makes some of the later results, e.g., that debate forests are composed of debate trees (Lemma 7), possible, as it would be difficult to claim two constructive definitions being equivalent. Thirdly, in this chapter, we have studied non-focused dialogues and introduced the notion of debate forest, and no such results were presented in [FT11a].

4.7 Conclusion

In this chapter, we have completed the presentation of our dialogue model by defining several legal-move functions and outcome functions. Table 4.5 summarises various types of dialogues we have defined with these legal-move and outcome functions. The system is generic and can be adapted in various applications. If we...
Table 4.5: Summary of types of dialogues defined with legal-move and outcome functions from Chapters 3 and 4.

<table>
<thead>
<tr>
<th>Dialogue</th>
<th>$\Lambda$ / $\Omega$</th>
<th>Brief Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Flat</td>
<td>$\lambda_{fl}$</td>
<td>Assumptions cannot be rule heads.</td>
</tr>
<tr>
<td>ABA</td>
<td>$\lambda_{fl}, \omega_{ABA}$</td>
<td>The framework drawn is a flat ABA framework.</td>
</tr>
<tr>
<td>Related</td>
<td>$\lambda_{rt}$</td>
<td>Utterances in a dialogue are (top-down) related.</td>
</tr>
<tr>
<td>Coherent</td>
<td>$\lambda_{fl}, \lambda_{rt}$</td>
<td>Dialogues that are flat and related.</td>
</tr>
<tr>
<td>Patient</td>
<td>$\lambda_{p}$</td>
<td>Only actual arguments are attacked.</td>
</tr>
<tr>
<td>Filtered</td>
<td>$\lambda_{fi}$</td>
<td>No duplicated attacks.</td>
</tr>
<tr>
<td>Focused</td>
<td>$\lambda_f$</td>
<td>Only a single way to support the claim.</td>
</tr>
<tr>
<td>Exhaustive</td>
<td>$\omega_{ex}$</td>
<td>All utterances allowed by a legal-move are made.</td>
</tr>
<tr>
<td>Positive</td>
<td>$\omega_{lw}$</td>
<td>Proponent answers all attacks.</td>
</tr>
<tr>
<td>One-way</td>
<td>$\lambda_{owe}$</td>
<td>Only a single way to support / attack the claim.</td>
</tr>
</tbody>
</table>

Table 4.6: Summary of our dialogue framework.

<table>
<thead>
<tr>
<th>Dialogue Goal:</th>
<th>Generic</th>
</tr>
</thead>
<tbody>
<tr>
<td>Topic Language:</td>
<td>Assumption-based Argumentation</td>
</tr>
<tr>
<td>Comm. Language:</td>
<td>Utterances (Definition 1)</td>
</tr>
<tr>
<td>Context:</td>
<td>Generic</td>
</tr>
<tr>
<td>Protocol:</td>
<td>Legal-move functions ($\lambda$)</td>
</tr>
<tr>
<td>Effect Rules:</td>
<td>Not directly applicable</td>
</tr>
<tr>
<td>Outcome Rules:</td>
<td>Outcome functions ($\omega$)</td>
</tr>
</tbody>
</table>

revisit Prakken’s list on dialogue elements in Chapter 2 (Page 34), we can see our dialogue framework can be summarised as in Table 4.6. We prove soundness of our model by connecting it with an argumentation framework. Our results amount to show a correspondence between justifying the claim of a dialogue by proving its acceptability with respect to a given semantics in the argumentation framework drawn from the dialogue.

Note that our soundness result is solely with respect to the shared information that has been put forward in a dialogue, i.e., the acceptability of the dialogue claim with respect to a certain argumentation semantics is tested in the ABA framework drawn from a dialogue; it has little implication on the acceptability of the dialogue in the “union” of the two agents. Hence, it is possible that a sentence, $\beta$, is acceptable with respect to a certain argumentation semantics, $S$, in the union of the two agents’ ABA frameworks, but a dialogue with $\beta$ as its claim is not successful with
respect to $S$. (The inverse is also true: it is possible for the claim of a successful dialogue being not acceptable in the union of the two agents’ knowledge-bases.) Hence, there is no completeness result for our dialogue model as it is. However, this issue will be addressed in the next chapter, when certain agent behaviours can be enforced through strategy-move functions.

In particular, in the next chapter, we present how agents can develop strategies which are compliant to the protocol given by legal-move functions yet fulfil agents’ aims in various dialogue types using strategy-move functions.
5 Dialogue Strategies

5.1 Introduction

In the previous chapter, we have focused on studying various properties of a class of dialogue protocols and showing some soundness results by proving that if a dialogue is compatible with certain legal-move functions, and certain outcome functions return true for such dialogue, then the claim of the dialogue is acceptable with respect to some argumentation semantics in the ABA framework drawn from the dialogue.

In this chapter, we shift our focus to agents’ behaviours in dialogues. Since agents participating in dialogues may have different aims and dialogues of different types have different goals, agents need to determine the appropriate information to disclose. We give strategies to help agents identify “suitable” utterances (and their content) in order to advance dialogues towards their goal while achieving their individual objectives.

In Chapter 3 we have introduced strategy-move functions to characterise agents’ behaviours in dialogues. Unlike legal-move functions that define public dialogue protocols, strategy-move functions decide, amongst all utterances allowed by the legal-move function, which of these are appropriate for the agent. In this chapter, we define a set of strategy-move functions suitable to characterise agents’ behaviours.

This chapter is organised as follows. Section 5.2 gives preliminary definitions used throughout this chapter. Section 5.3 introduces some specific classes of strategies-move functions. Section 5.4 shows formal results for dialogues constructed using these strategy-move functions. Section 5.5 revisits the Twelve Angry Men example and illustrates notions presented in this chapter. Section 5.6 discusses a few relate works. Section 5.7 concludes.
5.2 Preliminaries

As in Chapter 3 and 4, we consider two agents, \(a_1\) and \(a_2\) sharing a language \(\mathcal{L}\). In addition, in this chapter, we think of these agents as being equipped with ABA frameworks\(^1\). We will often use the ABA framework an agent is equipped with to denote the agent itself, hence \(a_1 = \langle \mathcal{L}, R_1, A_1, C_1 \rangle\) and \(a_2 = \langle \mathcal{L}, R_2, A_2, C_2 \rangle\).

We will often need to refer to the “union” of the (ABA frameworks of the) two agents, which amounts to the joint beliefs of the agents. This union represents all beliefs that the two agents have collectively, formalised by the following notion:

**Definition 40.** Given frameworks \(F_1 = \langle \mathcal{L}, R_1, A_1, C_1 \rangle\) and \(F_2 = \langle \mathcal{L}, R_2, A_2, C_2 \rangle\), the *joint framework* (of \(F_1\) and \(F_2\)) is \(F_J = F_1 \uplus F_2 = \langle \mathcal{L}, R_1 \cup R_2, A_1 \cup A_2, C_J \rangle\), where \(C_J(\alpha) = C_1(\alpha) \cup C_2(\alpha)\), for all \(\alpha\) in \(A_1 \cup A_2\).\(^2\)

Given frameworks \(F_J\) and \(F_1\), \(F_1\) is a *sub-framework* of \(F_J\), written \(F_1 \sqsubseteq F_J\), if and only if there exists \(F_2\) such that \(F_1 \uplus F_2 = F_J\). We use \(F_J\) to denote \(a_1 \uplus a_2\).

Note that the joint framework of any two ABA frameworks trivially is an ABA framework.

**Example 33.** We again revisit Example 10. Table 5.1 gives in one go the ABA frameworks of \(a_1\) and \(a_2\) (as indicated in the rightmost column) as well as their joint framework \(F_J\) (given by the entire table). Note that there are more rules in \(F_J\) than in \(F_5\), e.g., \(w_2\_is\_blonde \leftarrow \) and \(w_1\_is\_poor \leftarrow \) are in \(F_J\) but not \(F_5\), see Table 5.1 and Example 10 (Page 43).

Throughout this chapter, we will assume that \(a_1, a_2\) and \(F_J\) are flat. This is the case in Example 33. Also, we will use the notation \(A \vdash_R \beta\) for an argument where \(\beta\) is the claim, \(A\) is the supporting set of assumptions, and \(R\) is the set of rules used to construct this argument, e.g., in Example 33, given:

\[
\rho = \text{boy\_innocent} \leftarrow \text{boy\_not\_proven\_guilty}
\]
	hen

\[
\{\text{boy\_not\_proven\_guilty}\} \vdash_{\{\rho\}} \text{boy\_innocent}
\]
is an argument.

---

\(^1\)Note that we do not force agents to use ABA as their internal knowledge representation. The idea is that both agents are capable of re-representing their internal beliefs in ABA and communicate with ABA.

\(^2\)We assume that \(C(\alpha) = \{\}\) if \(\alpha \not\in A\).
Table 5.1: ABA frameworks for Example 33.

| Rules: \((\mathcal{R}, f)\) |  
|---|---|---|
| boy\_innocent \(\leftarrow\) boy\_not\_proven\_guilty | \(a_1, a_2\) |
| boy\_proven\_guilty \(\leftarrow\) w1\_is\_believable | \(a_1, a_2\) |
| w1\_not\_believable \(\leftarrow\) w1\_contradicted\_by\_w2 | \(a_1, a_2\) |
| w1\_contradicted\_by\_w2 | \(a_1\) |
| w2\_not\_believable \(\leftarrow\) w2\_has\_poor\_eyesight | \(a_1\) |
| w2\_has\_poor\_eyesight | \(a_2\) |
| w2\_is\_blonde | \(a_2\) |
| w1\_is\_poor | \(a_2\) |

| Assumptions: \((\mathcal{A}, f)\) |  
|---|---|
| boy\_not\_proven\_guilty | \(a_1, a_2\) |
| w1\_is\_believable | \(a_1, a_2\) |
| w2\_is\_believable | \(a_1, a_2\) |

| Contraries: \((\mathcal{C}, f)\) |  
|---|---|
| \(C(\text{boy\_not\_proven\_guilty}) = \{\text{boy\_proven\_guilty}\}\) | \(a_1, a_2\) |
| \(C(\text{w1\_is\_believable}) = \{\text{w1\_is\_not\_believable}\}\) | \(a_1, a_2\) |
| \(C(\text{w2\_is\_believable}) = \{\text{w2\_is\_not\_believable}\}\) | \(a_1, a_2\) |

When studying dialogue strategies, we will restrict attention to the agents’ beliefs that are (directly and indirectly) rule-related and related (respectively) to that topic, as defined below.

**Definition 41.** \(Y\) is directly rule-related to \(X\) with respect to a framework \(\langle\mathcal{L}, \mathcal{R}, \mathcal{A}, \mathcal{C}\rangle\) if and only if:

- \(X\) is an assumption \(\alpha \in \mathcal{A}\) and \(Y\) is \(\alpha\);
- \(X\) is a sentence \(\beta \in \mathcal{L} \setminus \mathcal{A}\) and \(Y\) is a rule \(\beta \leftarrow \_ \in \mathcal{R}\);
- \(X\) is a rule \(\beta_0 \leftarrow \beta_1, \ldots, \beta_n \in \mathcal{R}\) with \(n \geq 1\) and \(Y\) is
  - either a rule \(\beta_i \leftarrow \_ \in \mathcal{R}\), if \(\beta_i \notin \mathcal{A}\),
  - or an assumption \(\beta_i \in \mathcal{A}\).

Let \(O_{rr}\) be (the monotonic operator) defined, for any \(W \subseteq \mathcal{L} \cup \mathcal{R}\), as

\[
O_{rr}(W) = \{Y | Y \text{ is directly rule-related to } X \in W\}.
\]

Then, \(Y\) is (directly or indirectly) rule-related to \(X\) with respect to a framework \(\langle\mathcal{L}, \mathcal{R}, \mathcal{A}, \mathcal{C}\rangle\) if and only if \(Y\) belongs to the least fix-point of \(O_{rr}(\{X\})\).
Intuitively, rules and assumptions used to construct an argument are rule-related to the argument’s claim.

Definition 42. $Y$ is directly related to $X$ with respect to a framework $\langle L, R, A, C \rangle$ if and only if:

- $X$ is an assumption $\alpha \in A$ and $Y$ is $C(\alpha) = \perp$;
- $X$ is a sentence $\beta \in L \setminus A$ and $Y$ is a rule $\beta \leftarrow \perp \in R$;
- $X$ is a rule, and $Y$ is directly rule-related to $X$;
- $X$ is $C(\perp) = B$ and $Y$ is
  - a rule $\beta \leftarrow \perp \in R$, for some $\beta \in B$,
  - or an assumption $\alpha \in B \cap A$.

Let $O_r$ be (the monotonic operator) defined, for any $W \subseteq L \cup R \cup \{C(\alpha) = B | \alpha \in A, B \subseteq L\}$, as

$$O_r(W) = \{Y | Y \text{ is directly related to } X \in W\}.$$

Then, $Y$ is (directly or indirectly) related to $X$ with respect to a framework $\langle L, R, A, C \rangle$ if and only if $Y$ belongs to the least fix-point of $O_r(\{X\})$.

The following lemma is trivially true by Definitions 41 and 42.

Lemma 10. For all arguments $A \vdash_R \chi$, if $X \in A \cup R$, then $X$ is related to $\chi$.

The notions of rule-related and related can be used to identify suitable sub-frameworks of frameworks, as follows:

Definition 43. Given a framework $F = \langle L, R, A, C \rangle$ and a sentence $\chi \in L$, let $Y = \{X | X \text{ is rule-related to } \chi \text{ with respect to } \langle L, R, A, C \rangle\}$. The $\chi$-rule-related framework of $F$ is $F^{\chi r} = \langle L, R^{\chi r}, A^{\chi r}, C^{\chi r} \rangle$, with

- $R^{\chi r} = Y \cap R$,
- $A^{\chi r} = Y \cap A$,
- $C^{\chi r}(\alpha) = \{\}$ for each $\alpha \in A^{\chi r}$.

Namely, the $\chi$–rule-related framework is a sub-framework with all rules and assumptions used in arguments for $\chi$, as illustrated next:
Example 34. (Continuation of Example 33) Let $\chi = \text{boy \_ proven \_ guilty}$, then $F_{\chi r}$ is $\langle L, R_{\chi r}, A_{\chi r}, C_{\chi r} \rangle$ with $C_{\chi r}(\alpha) = \{\}$ for each $\alpha$ and

- $R_{\chi r} = \{\text{boy \_ proven \_ guilty} \leftarrow \text{w1 \_ is \_ believable}; \text{boy \_ proven \_ guilty} \leftarrow \text{w2 \_ is \_ believable}\}$
- $A_{\chi r} = \{\text{w1 \_ is \_ believable, w2 \_ is \_ believable}\}$

Note that $\chi$-rule-related frameworks are not ABA frameworks since they define the contrary of every assumption as empty.

Similarly, we can define the topic-related framework as follows.

Definition 44. Given a framework $F = \langle L, R, A, C \rangle$ and a sentence $\chi \in L$, let $Y = \{X|X$ is related to $\chi$ with respect to $\langle L, R, A, C \rangle\}$. Then, the $\chi$-related framework of $F$ is $F_{\chi} = \langle L, R_{\chi}, A_{\chi}, C_{\chi} \rangle$ with

- $R_{\chi} = Y \cap R$;
- $A_{\chi} = Y \cap A$;
- $C_{\chi}$ is such that, for every $\alpha \in A_{\chi}$, $C_{\chi}(\alpha) = B$ if and only if $(C(\alpha) = B) \in Y$.

Namely, the $\chi$-related framework is a sub-framework that contains all information (directly or indirectly) related to $\chi$, as illustrated next:

Example 35. (Continuation of Example 33) Let $\chi = \text{boy \_ innocent}$. Then

- $w2 \_ is \_ blonde \leftarrow$
- $w1 \_ is \_ poor \leftarrow$

are not related to $\chi$. Therefore, the $\chi$-related framework, $F_{\chi}$, of $F_{J}$ is $F_{J}$ with these rules omitted.

Unlike $\chi$-rule-related frameworks, assumptions in $\chi$-related frameworks admit non-empty contraries. Thus, for any ABA framework $F$, $F_{\chi}$ is an ABA framework.

Notation 1. We will say that an argument $A \vdash_{R} \beta$ is in $\langle L, R, A, C \rangle$ if and only if $\beta \in L, A \subseteq A$, and $R \subseteq R$. Moreover, given a dialogue $\delta \in D$, we will say that an argument $A$ is in $\delta$ and $\delta$ contains $A$ if and only if $A$ is in $F_{\delta}$.

98
The following two lemmas trivially hold by Definitions 43 and 44.

**Lemma 11.** Given an ABA framework \( \mathcal{F} \), an argument \( A = A \vdash_R \chi \) is in \( \mathcal{F} \) if and only if \( \chi \) is in \( \mathcal{F}^{\chi_r} \).

**Lemma 12.** Given an ABA framework \( \mathcal{F} = \langle \mathcal{L}, \mathcal{R}, \mathcal{A}, \mathcal{C} \rangle \), \( \chi \in \mathcal{L} \), and the \( \chi \)-related framework \( \mathcal{F}^\chi \), then \( \chi \) is \( S \)-acceptable in \( \mathcal{F} \) if and only if \( \chi \) is \( S \)-acceptable in \( \mathcal{F}^\chi \), \( S \in \{ \text{admissible, grounded, ideal} \} \).

### 5.3 Classes of Strategy-move Functions

We have introduced the concept of strategy-move function in Chapter 3. In the remainder of this section we define a number of classes of strategy-move functions, each describing a certain behaviour. The first strategy-move function characterises the “truthfulness” of agents. If a dialogue is constructed with a \textit{truthful strategy-move function} with respect to \( a_k \), then \( a_k \) only utter rules, assumptions, and contraries it believes (namely, from its ABA framework).

**Definition 45.** A \textit{truthful strategy-move function} \( \phi \in \Phi \) for agent \( a_k \) (\( k \in \{1, 2\} \)) is such that, given a dialogue \( \delta \in \mathcal{D} \) and a legal-move function \( \lambda \in \Lambda \), for all \( u \in \phi(\delta, \lambda) \) made by \( a_k \), the content \( C \) of \( u \) is such that:

1. if \( C = rl(\rho) \), then \( \rho \in \mathcal{R}_k \),
2. if \( C = asm(\alpha) \), then \( \alpha \in \mathcal{A}_k \),
3. if \( C = ctr(\beta, \beta') \), then \( \beta' \in \mathcal{C}_k(\beta) \).

We refer to a generic truthful strategy-move function as \( \phi_t \).

The second strategy-move function we define characterises the “completeness” of an agent’s utterances: the \textit{thorough strategy-move function} specifies that agents must not utter \( \pi \) if there is any other “truthful” utterance allowed by the given legal-move function.

**Definition 46.** A \textit{thorough strategy-move function} \( \phi \in \Phi \) for agent \( a_k \) (\( k \in \{1, 2\} \)) is such that, given \( \delta \in \mathcal{D} \) such that \( \delta \) is constructed with a truthful strategy-move function with respect to \( a_k \), given \( \lambda \in \Lambda \), then for all \( u \in \phi(\delta, \lambda) \) made by \( a_k \), if \( u \) is a pass-utterance then there exists no regular utterance \( u' \in \lambda(\delta) \cap U^k \) such that \( \delta \circ u \) is constructed with a truthful strategy-move function.

We refer to a generic thorough strategy-move function as \( \phi_h \).
We further define the notion of non-attack strategy-move function, specifying that agents do not utter contraries. Hence, agents that use the non-attack strategy can only construct arguments.

**Definition 47.** A non-attack strategy-move function $\phi \in \Phi$ for agent $a_k$ ($k \in \{1, 2\}$) is such that given $\delta = \langle u_1, \ldots, u_n \rangle$ constructed with $a_k$ using $\phi$, then there is no utterance $u_i$, $1 \leq i \leq n$ of the form $\langle a_k, \omega, ctr(\omega), \omega \rangle$ in $\delta$.

We refer to a generic non-attack strategy-move function as $\phi_{na}$.

For convenience, we also define the class of non-attack-thorough strategy-move functions, as follows.

**Definition 48.** A non-attack-thorough strategy-move function $\phi \in \Phi$ for agent $a_k$ ($k \in \{1, 2\}$) is such that, given $\delta \in D$ such that $\delta$ is constructed with a truthful strategy-move function with respect to $a_k$, given $\lambda \in \Lambda$, then for all $u \in \phi(\delta, \lambda)$ made by $a_k$, $u$ is not of the form $\langle \omega, \omega, ctr(\omega), \omega \rangle$ and if $u$ is a pass-utterance then there exists no utterance of the form $\langle \omega, \omega, rl(\omega), \omega \rangle$ or $\langle \omega, \omega, asm(\omega), \omega \rangle$, $u \in \lambda(\delta) \cap U_k$ such that $\delta \circ u$ is constructed with a truthful strategy-move function.

We use $\phi_{nh}$ to denote a generic non-attack-thorough strategy-move function.

Intuitively, an agent that uses $\phi_{nh}$ in a dialogue attempts to utter all rules and assumptions from its ABA framework.

Agents that use the pass strategy-move may initiate dialogues but do not utter any other information throughout the dialogue.

**Definition 49.** A pass strategy-move function $\phi \in \Phi$ for agent $a_k$ ($k \in \{1, 2\}$) is such that given $D_{a_i} \chi = \delta$ and $\lambda \in \Lambda$, if $\gamma(\delta) = a_k$, then for $l \neq k$, $l = i, j$, $\phi(\delta, \lambda) = \begin{cases} \{(a_k, a_i, 0, claim(\chi), 1)\} & \text{if } \delta = \langle \rangle \text{ and } k = i; \\ \{(a_k, a_i, 0, \pi, ID) | ID \in ID\} & \text{otherwise.} \end{cases}$

We refer to a generic pass strategy-move function as $\phi_{ps}$.

### 5.4 Results

We give a number of properties for strategy-move functions. The non-attack and thorough strategy-move functions jointly give an agent disclosing all rules and assumptions in arguments for the topic.

**Proposition 2.** Given a coherent dialogue $D_{a_i} \chi = \delta$ for $a_i, a_j \in \{a_1, a_2\}$ constructed using a non-attack thorough strategy-move function $\phi_{nh}$ with respect to $a_k, k \in \{i, j\}$, then for all arguments $A = A \vdash_R \chi$ in $a_k$, $\lambda$ is in $F_\delta$. 

100
Proof. It follows from Definition 48. For any argument \( A \vdash_R \chi \), for all \( \alpha \in A \) and \( \rho \in R \), \( \alpha \) and \( \rho \) are related to \( \chi \) (Lemma 10). Given that \( \delta \) is coherent and constructed with \( \phi_{nh} \) with respect to \( a_k \), all such \( \alpha \) and \( \rho \) in \( a_k \) must be disclosed in \( \delta \), i.e., there are utterances of the forms \( \langle \ldots, rl(\rho), \ldots \rangle \) or \( \langle \ldots, asm(\alpha), \ldots \rangle \) in \( \delta \). (It is uncertain which agent makes these utterances though, as both agents can make them. However, it is certain that if the other agent does not make such utterances, \( a_k \) will.) Therefore if \( A \vdash_R \chi \) is in \( a_k \), it is also in \( F_\delta \).

Note that Proposition 2 does not specify a strategy-move function for the other agent \( a'_k \in \{a_i, a_j\}, a'_k \neq a_k \). Hence Proposition 2 describes a situation where, regardless of the strategy \( a'_k \) uses, as long as \( a_k \) uses \( \phi_{nh} \), then all arguments for the claim of the dialogue in \( a_k \) are disclosed in \( \delta \).

**Proposition 3.** Given a coherent dialogue \( D_{a_i a_j}(\chi) = \delta \), constructed with a thorough strategy-move function \( \phi_{tf} \), for \( a_i, a_j \in \{a_1, a_2\} \), it holds that \( F_\delta \subseteq F_J \).

**Proof.** From Definition 45 as if both agents only utter information from their ABA frameworks, then obviously \( F_\delta \) is a sub-framework of the joint ABA framework of the two agents.

**Proposition 4.** If \( F_J \) is flat and \( D_{a_i a_j}(\chi) = \delta \) is a coherent dialogue constructed with \( \phi_{h} \), then \( F_\delta \) is flat.

**Proof.** This proposition is true as (1) \( F_\delta \) must be an ABA framework as both \( a_i \) and \( a_j \) are and (2) every rule, assumption and contrary in \( F_\delta \) is also in \( F_J \) (Proposition 3), so the flatness of \( F_J \) implies the flatness of \( F_\delta \) (since trivially any sub-framework of a flat ABA framework is flat).

Since a thorough strategy-move function \( \phi_h \) represents truthfulness and thoroughness, coherent dialogues constructed with \( \phi_h \) contain all information about the dialogue topic from the two agents. In this case, the ABA framework drawn from a dialogue and the topic-related framework obtained from the joint framework of the two agents are the same. Formally:

**Lemma 13.** Given a coherent dialogue \( \delta = D_{a_i a_j}(\chi) \) constructed with a thorough strategy-move function \( \phi_h \), if the \( \chi \)-related framework of \( F_J \) is \( F_\chi \), then \( F_\chi = F_\delta \).

**Proof.** It follows from Definitions 45 and 46 and the condition that \( \delta \) is coherent. In order to prove it, we need to show that \( \exists W \), such that \( W \) is either a rule, assumption, or contrary in \( F_J \) and \( W \) is (directly and indirectly) related to \( \chi \) but \( W \) is not
the content of any utterance in \( \delta \). Such a \( W \) cannot exist because \( \delta \) is constructed with \( \phi_h \), hence both agents are bound to utter all rules, assumptions, and contraries that are (directly and indirectly) related to \( \chi \) from their ABA frameworks.

We link the acceptability result between a dialogue constructed with a thorough strategy-move function \( \phi_h \) with the union of the two agents’ ABA frameworks.

**Theorem 5.** Given a coherent dialogue \( D_{a_i}^{a_j}(\chi) = \delta \), for \( a_i, a_j \in \{a_1, a_2\} \), constructed with a thorough strategy-move function \( \phi_h \), \( \chi \) is \( S \)-acceptable in \( F_J \), for \( S \in \{ \text{admissible, grounded, ideal} \} \), if and only if \( \chi \) is \( S \)-acceptable in \( F_\delta \).

**Proof.** By Lemma 12, the acceptability of \( \chi \) is the same in \( F_J \) and \( F_\chi \), the \( \chi \)-related framework of \( F_J \). By Lemma 13, given a coherent \( \delta \in D \) constructed with \( \phi_h \), we have \( F_\delta = F_\chi \). Hence this theorem holds.

However, the acceptability of the topic in the joint framework can sometimes be assessed with a sub-framework of \( F_J \), as shown in the following theorem.

**Theorem 6.** Given a focused dialogue \( D_{a_i}^{a_j}(\chi) = \delta \) constructed with a thorough strategy-move function \( \phi_h \), for \( a_i, a_j \in \{a_1, a_2\} \), if \( \delta \) is a-/g-successful, then \( \chi \) is \( S \)-acceptable in \( F_J \), for \( S \) being admissible, and grounded respectively.

**Proof.** Let \( F_\delta \) be \( \langle L, R_\delta, A_\delta, C_\delta \rangle \) and let \( F_J \) be \( \langle L, R_J, A_J, C_J \rangle \).

(1) Since \( \delta \) is focused and a-/g-successful, \( \chi \) is \( S \)-acceptable in \( F_\delta \). Then there is \( A \vdash \chi \) and a set of \( S \)-acceptable assumptions \( A \subseteq A_\delta \) (in \( F_\delta \)), such that \( A \subseteq A \) (Theorem 1 and Theorem 2). Hence there does not exist \( A' \subseteq A_\delta \) such that \( A' \) attacks \( A \) and \( A' \) is not attacked by \( A \).

(2) Since \( \delta \) is focused and constructed with \( \phi_h \), then for all assumptions \( A^* \subseteq A_J \), if \( A^* \) attacks \( A \) in \( F_J \), then \( A^* \subseteq A_\delta \).

By (1) and (2), there is no \( A^* \subseteq A_J \), such that \( A^* \) attacks \( A \) and \( A^* \) is not attacked by \( A \). Hence \( A \) and \( \chi \) are \( S \)-acceptable in \( F_J \).

Theorem 6 is useful as if the agents want to justify the acceptability (under admissible and grounded semantics) of a topic in the joint framework, then it is sufficient to justify the topic using a focused dialogue, thus requiring less disclosure of agents’ beliefs. Note that Theorem 6 does not hold for the ideal semantics as focused dialogues do not compute it. Also, the converse of Theorem 6 is not true. Indeed, given that \( \delta \) is a focused dialogue constructed with \( \phi_h \), if \( \delta \) is not a-/g-successful, and hence \( \chi \) is not \( S \)-acceptable with respect to \( F_\delta \), then \( \chi \) may or may not be \( S \)-acceptable in \( F_J \), as illustrated next in Example 36:
Table 5.2: A dialogue that is constructed with a truthful strategy-move function in Example 33.

<table>
<thead>
<tr>
<th>Agent 1</th>
<th>Agent 2</th>
<th>Turn</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>a₁</td>
<td>a₂</td>
<td>0</td>
<td>claim(boy_innocent), 1</td>
</tr>
<tr>
<td>a₂</td>
<td>a₁</td>
<td>1</td>
<td>rl(boy_innocent ← boy_not_proven_guilty), 2</td>
</tr>
<tr>
<td>a₁</td>
<td>a₂</td>
<td>0</td>
<td>π, 3</td>
</tr>
<tr>
<td>a₂</td>
<td>a₁</td>
<td>0</td>
<td>π, 4</td>
</tr>
</tbody>
</table>

Table 5.3: A dialogue that is constructed with a non-attack thorough strategy-move function in Example 33.

<table>
<thead>
<tr>
<th>Agent 1</th>
<th>Agent 2</th>
<th>Turn</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>a₁</td>
<td>a₂</td>
<td>0</td>
<td>claim(boy_innocent), 1</td>
</tr>
<tr>
<td>a₁</td>
<td>a₂</td>
<td>1</td>
<td>rl(boy_innocent ← boy_not_proven_guilty), 2</td>
</tr>
<tr>
<td>a₁</td>
<td>a₂</td>
<td>2</td>
<td>asm(boy_not_proven_guilty), 3</td>
</tr>
<tr>
<td>a₂</td>
<td>a₁</td>
<td>0</td>
<td>π, 4</td>
</tr>
<tr>
<td>a₂</td>
<td>a₁</td>
<td>0</td>
<td>π, 5</td>
</tr>
</tbody>
</table>

Example 36. Let $F_{J} = (L, R, A, C)$, with $R = \{ \chi \leftarrow a; \chi \leftarrow b; b \leftarrow \}$ and $A = \{ \}$. Then the dialogue shown in the following table is focused but not (a-/g-)successful, even if $\chi$ is $S$-acceptable in $F_{J}$, for $S \in \{ \text{grounded, admissible} \}$.

<table>
<thead>
<tr>
<th>Agent 1 (a₁)</th>
<th>Agent 2 (a₂)</th>
</tr>
</thead>
<tbody>
<tr>
<td>\langle a₁, a₂, 0, claim(\chi), 1 \rangle</td>
<td>\langle a₂, a₁, 1, rl(\chi \leftarrow a), 2 \rangle</td>
</tr>
<tr>
<td>\langle a₁, a₂, 0, π, 3 \rangle</td>
<td>\langle a₂, a₁, 0, π, 4 \rangle</td>
</tr>
</tbody>
</table>

5.5 Illustration

In this section, we reuse the *Twelve Angry Men* setting given in Examples 10 and 33. We first show a dialogue that is constructed with a truthful strategy-move function in Table 5.2. Given the two agents use a truthful strategy-move function, this dialogue could terminate at any point. The two agents only utter rules, assumptions, and contraries from their knowledge base in this dialogue. We then show a dialogue that is constructed with a non-attack thorough strategy-move function in Table 5.3. This dialogue “constructs” an (actual) argument for the claim boy_innocent and then terminates. There is no information (rules, assumptions, or contraries) that attacks the claim is uttered. We also point out that the dialogue shown in Table 3.1 is constructed with a thorough strategy-move function, as all information for and against the claim has been uttered in this dialogue.
5.6 Related Work

As introduced in Chapter 2, Black and Hunter [BH09] present a formal system for inquiry dialogues based on DeLP [GS04] as the underlying argumentation framework. In that work, they also studied agent strategies for argument inquiry and warrant inquiry (see Chapter 2). Though some of our strategy-move functions (e.g., thorough and non-attack-thorough strategy-move functions) share similar spirits with the ones given in [BH09], we have taken the approach of defining agent strategies as compositions of strategy-move functions, rather than give a specific strategy-move function per dialogue type. As we demonstrate in Chapter 6 and 7, our approach brings us flexibility in the types of dialogue that we can model.

Boella et al [BGH07] use the MacKenzie dialogue system introduced in [Mac79] to map some dialogue protocols into strategies. Our work is orthogonal as we fix the dialogue framework and study strategies that apply in it.

This Chapter is based on [FT12a], we have included more examples to illustrate our results.

5.7 Conclusion

In this chapter, we have studied some dialogue strategies agents can use in various dialogues. In particular, we have defined a set of classes of strategy-move functions, summarised in Table 5.4. With these strategy-move functions, we have shown that:

1. if an agent uses a non-attack-thorough strategy-move function to construct a dialogue, then the resulting dialogue contains all arguments for the dialogue claim that exist in this agent’s knowledge base (Proposition 2);

2. if both agents use truthful strategy-move functions to construct a dialogue, then information contained in the resulting dialogue is a subset of the union of the two agents knowledge-base (Proposition 3);

3. if the joint framework of the two agents’ ABA framework is flat, and the two agents use thorough strategy-move functions to construct dialogues, then the ABA framework drawn from the resulting dialogue is flat (Proposition 4);

4. if both agents use thorough strategy-move functions to construct a dialogue, then ABA framework drawn from the resulting dialogue contains all infor-
<table>
<thead>
<tr>
<th>Strategy-move Functions</th>
<th>Brief Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Truthful ($\phi_{tf}$)</td>
<td>An agent only utters truthful information.</td>
</tr>
<tr>
<td>Thorough ($\phi_{th}$)</td>
<td>An agent utters $\pi$ when there is nothing else.</td>
</tr>
<tr>
<td>Non-attack ($\phi_{na}$)</td>
<td>An agent doesn’t attacks the claim.</td>
</tr>
<tr>
<td>Non-attack thorough ($\phi_{nah}$)</td>
<td>An agent utters all supports to the claim.</td>
</tr>
<tr>
<td>Pass ($\phi_{ps}$)</td>
<td>An agent makes the claim and pass.</td>
</tr>
</tbody>
</table>

...
6 Dialogue Instantiations (I)

6.1 Introduction

In previous chapters, we have introduced our argumentation based dialogue framework. In this chapter and the next, we show how the framework can be instantiated to several types of dialogues. Each of the presented instantiations serves as an example in applying our dialogue model. In this chapter, we focus on information-seeking, inquiry and persuasion dialogues.

This chapter is organised as follows. Section 6.2 presents our formalisations of information-seeking and inquiry dialogues, and proves soundness and completeness of strategies resulting from strategy-move functions with respect to these formalisations, using the results from Chapter 5. Section 6.3 presents our formalisation of persuasion dialogues. Therein, we connect our results with mechanism design. Section 6.4 concludes.

6.2 Information-seeking and Inquiry

We start our discussion with information-seeking and inquiry dialogues. As seen in Chapter 2, the main characteristics of information-seeking and inquiry dialogues are summarised in Table 6.1.

Thus, in both information-seeking and inquiry, agents need to determine the appropriate information to disclose. We use strategy-move functions defined in Chapter 5 to help agents identify “suitable” utterances that advance information-seeking and inquiry dialogues towards their goals while fulfilling the participants’ aims.

We prove that dialogues where agents adopt specific classes of strategy-move functions are (i) sound and (ii) complete for information-seeking and inquiry, in that (i) the dialogues constructed with these strategy-move functions achieve the main goals of these dialogue types, starting from the initial situation, and (ii) the existence of (information-seeking and inquiry) dialogues achieving the goals guar-
Table 6.1: Information-seeking and inquiry dialogues (from [WK95]).

<table>
<thead>
<tr>
<th><strong>Information-seeking Dialogue</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>Initial Situation - Personal ignorance;</td>
</tr>
<tr>
<td>Main Goal - Spreading knowledge &amp; revealing positions;</td>
</tr>
<tr>
<td>Participant’s Aims - Gain, pass on, show or hide personal knowledge.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>Inquiry Dialogue</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>Initial Situation - General ignorance;</td>
</tr>
<tr>
<td>Main Goal - Growth of knowledge &amp; agreement;</td>
</tr>
<tr>
<td>Participant’s Aims - Find a “proof” or destroy one.</td>
</tr>
</tbody>
</table>

Table 6.2: Two formulations of information-seeking dialogues.

<table>
<thead>
<tr>
<th><strong>Information-seeking Dialogue:</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>IS-Type I:</strong> Initial Situation: some $A \vdash \chi$ in $a_2$ which is not in $a_1$.</td>
</tr>
<tr>
<td>Main Goal: find $\delta$ such that all $A \vdash \chi$ in $a_2$ are in $\mathcal{F}_\delta$.</td>
</tr>
</tbody>
</table>

| **IS-Type II:** Initial Situation: some $A \vdash \chi$ in $a_2$ but none in $a_1$. |
| Main Goal: find $\delta$ such that one $A \vdash \chi$ in $a_2$ is in $\mathcal{F}_\delta$. |

antee the existence of dialogues, constructed with these strategy-move functions, also achieving these goals. We prove these results for two novel formulations of each of information-seeking and inquiry dialogues, formalising the definitions of Table 6.1, and given in Tables 6.2 and 6.4, discussed later.

### 6.2.1 Information-seeking Dialogues

In Table 6.2, we model information-seeking dialogues as involving a *questioner* agent $a_1$ posing a topic, $\chi$, and an *answerer* agent $a_2$ uttering information of relevance to $\chi$. The purpose is to spread knowledge about arguments for $\chi$. We assume that the questioner contributes no information, apart from initiating the dialogue; and the answerer is interested in conveying information for $\chi$, but not against it. Thus, there is an asymmetric distribution of information, in line with the original description.

The two formulations differ in that IS-Type I dialogues convey all arguments whereas IS-Type II dialogues convey only one argument. Both types can be seen as concrete formalisations of the informal notion in Table 6.1.

The following result sanctions the soundness, for IS-Type I information-seeking,
of the questioner using a pass strategy-move function $\phi_{ps}$ and the answerer using a non-attack strategy-move function $\phi_{nh}$ to construct coherent dialogues.

**Proposition 5.** Let $A \vdash \chi$ be in $a_2$ but not in $a_1$. Then, if a coherent dialogue $\delta = D_{a_2}^{a_1}(\chi)$ is constructed by $a_1$ using some $\phi_{ps}$ and $a_2$ using some $\phi_{nh}$, then $F_\delta$ contains all $A \vdash \chi$ in $a_2$.

**Proof.** Directly from Proposition 2, where $a_1$ uses $\phi_{ps}$, rather than an unspecified strategy-move function. $\Box$

The next result sanctions the completeness, for IS-Type I information-seeking, of coherent dialogues with the answerer using some $\phi_{nh}$. Informally, the following proposition states that if there is a dialogue that captures all arguments for a topic in the answerer agent, then there is a coherent dialogue constructed by the questioner agent using a pass strategy and the answerer agent using a non-attack-thorough strategy which captures all arguments for the topic in the answer agent as well.

**Proposition 6.** Let $A \vdash \chi$ be in $a_2$ but not in $a_1$. Then, if there is $\delta = D_{a_2}^{a_1}(\chi)$ such that $F_\delta$ contains all $A \vdash \chi$ that are in $a_2$, then there exists a coherent dialogue $\delta'$ constructed by $a_1$ using some $\phi_{ps}$ and $a_2$ using some $\phi_{nh}$ such that $F_{\delta'}$ contains all $A \vdash \chi$ that are in $a_2$.

**Proof.** $\delta'$ can be constructed as follows. Given that $a_1$ uses $\phi_{ps}$, $a_1$ will utter the claim, but not contribute any rule, assumption, or contrary to $\delta'$. Given that there exists $A_1 \vdash R_1 \chi, \ldots, A_n \vdash R_n \chi$ in $a_2$, $\delta'$ can be constructed in a way such that for every $\alpha \in A_1 \cup \ldots \cup A_n$ and $\rho \in R_1 \cup \ldots \cup R_n$, there is an utterance of the form $\langle a_2, a_1, \_ , asm(\alpha), \_ \rangle$ or $\langle a_2, a_1, \_ , rl(\rho), \_ \rangle$, respectively, in $\delta'$; and there is no other regular utterance made by $a_2$. The resulting $\delta'$ is coherent, as rules and assumptions in an argument are related. Moreover, $a_2$ uses $\phi_{nh}$, as $a_2$ utters no contraries. $\Box$

The following result sanctions the completeness, for IS-Type II information-seeking, of the questioner using some $\phi_{ps}$ and the answerer using some $\phi_{nh}$ to construct focused dialogues.

**Proposition 7.** Let $A \vdash \chi$ be in $a_2$, and $a_1$ be such that there is no argument $A' \vdash \chi$ in $a_1$. Then, if there is a dialogue $D_{a_2}^{a_1}(\chi) = \delta$ such that $A = A \vdash \chi$ is in $F_\delta$, then there exists a focused dialogue $\delta'$ constructed by $a_1$ using some $\phi_{ps}$ and $a_2$ using some $\phi_{nh}$ such that $A$ is in $F_{\delta'}$.  

108
Table 6.3: Information-seeking dialogue in Example 33.

\[
\begin{align*}
\langle a_2, a_1, 0, \text{claim}(w_1\text{ not believable}), 1 \rangle \\
\langle a_1, a_2, 1, \text{rl}(w_1\text{ not believable} \leftarrow w_1\text{ contradicted by } w_2), 2 \rangle \\
\langle a_1, a_2, 2, \text{rl}(w_1\text{ contradicted by } w_2 \leftarrow), 3 \rangle \\
\langle a_1, a_2, 0, \pi, 4 \rangle \\
\langle a_2, a_1, 0, \pi, 5 \rangle
\end{align*}
\]

The proof of this proposition is similar to the one in Proposition 6, except that there is only one argument constructed in the dialogue.

**Proof.** Let \( a_1 \) use \( \phi_{ps} \). Given that \( A \vdash R \chi \) is in \( a_2 \), \( \delta' \) can be constructed such that for every \( \rho \in R \) and \( \alpha \in A \), there is an utterance of the form \( \langle a_2, a_1, \_ \_ \_ \text{rl}(\rho), \_ \_ \_ \rangle \) or \( \langle a_2, a_1, \_ \_ \_ \text{asm}(\alpha), \_ \_ \_ \rangle \), respectively, in \( \delta' \); and there is no other regular utterance made by \( a_2 \). Such \( \delta' \) is focused, as it contains a single argument. Moreover, \( a_2 \) uses \( \phi_{nh} \), as \( a_2 \) utters no contraries but rules and assumptions from its ABA framework.

Note that for the reasons illustrated in Example 36, given that there is an \( A = A \vdash \chi \) in \( a_2 \), it is not the case that all focused dialogues \( \mathcal{D}_{a_2}^{01}(\chi) \) constructed with \( a_1 \) using \( \phi_{ps} \) and \( a_2 \) using \( \phi_{nh} \) contain \( A \).

An information-seeking dialogue constructed with \( \phi_{nh} \) and \( \phi_{ps} \) for the two agents in Example 33 is shown next.

**Example 37.** Table 6.3 shows an information-seeking dialogue, in which the questioner queries about arguments for the claim \( w_1\text{ not believable} \) to the answerer. Note that all rules used in this example are known by the answerer only. Since there is a single argument for the topic, the dialogue is focused. It is easy to see that this dialogue is also coherent. There is no difference between IS-Type I and IS-Type II in this example. Note that, unlike everywhere else in this section, in this example \( a_2 \) is the questioner and \( a_1 \) is the answerer.

### 6.2.2 Inquiry Dialogues

We formulate inquiry dialogue in two ways, shown in Table 6.4, where \( S \in \{ \text{admissible, grounded, ideal} \} \). For I-Type I, dialogues start with a topic, contain topic-related rules, assumptions, and contraries uttered by both agents, and eventually reach the “proof” or “disproof” of the topic. For I-Type II, dialogues start
Table 6.4: Two formulations of inquiry dialogues.

<table>
<thead>
<tr>
<th>Inquiry Dialogue</th>
<th>I-Type I: Initial Situation: it is uncertain if $\chi$ is $\mathcal{S}$-acceptable in $\mathcal{F}_J$</th>
<th>Main Goal: testing the $\mathcal{S}$-acceptability of $\chi$ in $\mathcal{F}_J$</th>
</tr>
</thead>
<tbody>
<tr>
<td>I-Type II:</td>
<td>There is no argument $A \vdash \chi$ in either $a_1$ or $a_2$.</td>
<td>Main Goal: testing whether there exists $A \vdash \chi$ is in $\mathcal{F}_J$</td>
</tr>
</tbody>
</table>

with a topic, contain rules and assumptions in arguments for this topic. We give no specific name to agents in inquiry dialogues. Note that here agents contribute to dialogues symmetrically.

The next proposition sanctions soundness and completeness, for I-Type I inquiry, of using thorough strategy-move functions $\phi_h$ to construct coherent dialogues.

**Proposition 8.** To test the $\mathcal{S}$-acceptability of $\chi$ in $\mathcal{F}_J$ is to test the $\mathcal{S}$-acceptability of $\chi$ in $\mathcal{F}_\delta$ for a coherent $\delta \in \mathcal{D}$ constructed using a thorough strategy-move function $\phi_h$.

**Proof.** Directly from Theorem 5.

The following result sanctions the soundness and completeness for I-Type II inquiry of using a non-attack thorough strategy-move function $\phi_{nh}$ to construct coherent dialogues.

**Proposition 9.** If there is no $A \vdash \chi$ in either $a_1$ or $a_2$, to test whether $A = A \vdash \chi$ is in $\mathcal{F}_J$ is to test whether $A$ is in a coherent dialogue constructed using a non-attack thorough strategy-move function $\phi_{nh}$.

**Proof.** We show that $A = A \vdash R \chi$ is in $\mathcal{F}_J$ if and only if $A$ is in $\mathcal{F}_\delta$, where $\delta$ is coherent and constructed using $\phi_{nh}$.

1. Since $\delta$ is constructed using $\phi_{nh}$, $a_1$ and $a_2$ are truthful, therefore all arguments $A_1 \vdash \chi, \ldots, A_n \vdash \chi$ in $\mathcal{F}_\delta$ are in $\mathcal{F}_J$.

2. We show that all arguments $A_1 \vdash \chi, \ldots, A_n \vdash \chi$ in $\mathcal{F}_J$ are in $\mathcal{F}_\delta$. Suppose $A' = A' \vdash R' \chi$ is in $\mathcal{F}_J$ but not in $\mathcal{F}_\delta$, then $\exists X \in A' \cup R'$, such that $X$ is not the content of any regular utterance in $\delta$. But this cannot be, by Lemma 10, as $X$ is related to $\chi$; and, for all such $X$, there is an utterance $u = (\_ \, \_ , \_ , CT(X), \_ )$ in $\delta$, where $CT(\_ )$ is $rt(\_ )$ or $asm(\_ )$, as $\delta$ is coherent and constructed with $\phi_{nh}$. Hence we have a contradiction. 

110
An I-Type I inquiry, coherent dialogue constructed with a thorough strategy-move function $\phi_h$ is shown below.

**Example 38.** Table 3.1 in Chapter 3 shows a I-Type I inquiry dialogue for the two agents in Example 33. This dialogue is coherent. From this dialogue, we can draw an ABA framework $\mathcal{F}_\delta$, that is composed of all rules, assumptions, and contraries shown in the dialogue. Clearly, $\mathcal{F}_\delta$ is topic-related framework of $\mathcal{F}_J$, and the $S$-acceptability of the topic can be examined in $\mathcal{F}_\delta$.

### 6.2.3 Related Work

As we discuss in Chapter 2, Black and Hunter [BH09] present a formal system for inquiry dialogues. Our work differs in several ways. Firstly, we give formal definitions of the goals of information-seeking and inquiry dialogues in terms of argumentation semantics. Secondly, we have studied information-seeking dialogues whereas they focused solely on inquiry. Thirdly, the underlying dialogue framework we use is generic rather than tailored to inquiry.

Parsons et al [PWA03] present a study on information-seeking, inquiry and persuasion dialogues, focusing on complexity results. They use classical logic as the base for argumentation and specify dialogue protocols for each dialogue type, in an algorithmic manner. Finally, they do not compare dialogue outcomes with the joint knowledge held by the two agents.

### 6.2.4 Summary

In this section, we have shown how to use some dialogue strategies we defined earlier in information-seeking and inquiry dialogues, some formal interpretations of these dialogues. We have shown that the specified dialogue strategies are suitable for these interpretations.

We have shown that in information-seeking dialogues, the answerer should be truthful and disclose directly related information about the topic; whereas in inquiry dialogues, both agents should be truthful and disclose directly or indirectly related information about the topic.

### 6.3 Persuasion Dialogues with Mechanism Design

After studying information-seeking and inquiry dialogues, we shift our focus to persuasion dialogues in this section. We have briefly introduced the concept of
persuasion dialogue in Chapter 2, here, we approach dialogues from a mechanism design [Jac03] perspective to develop strategies for agents and study properties thereof. Similarly to the previous section, we build upon our dialogue model presented in previous chapters.

We model persuasion as dialogues starting with the *persuader* agent posing a topic, and then subsequently the persuader and the *persuadee* agent putting forward information *for* and *against* the topic (respectively). A persuasion is successful if the topic is “proved” through this dialogue. One difficult part in persuasion is to prevent the persuader putting forward misleading information that does not hold in its knowledge base. We specify conditions under which the persuader will not utter such information.

In mechanism design terms, we consider dialogues as strategies and information disclosed in dialogues as actions. Using the notion of strategy-move function introduced in Chapter 5, we define two additional strategy-move functions to characterise information uttered by agents in persuasion dialogues.

### 6.3.1 Mechanism Design Background

Mechanism design (e.g. see [Jac03]) provides an abstraction of distributed problem solving amongst interacting, self-interested agents. In the language of mechanism design, agents are characterised by *types*, which are abstractions of their internal, private beliefs. Given \( I \geq 2 \) agents, the space of possible types for agent \( i \) (\( 1 \leq i \leq I \)) is denoted by \( \Theta_i \) and its type is \( \theta_i \in \Theta_i \). Moreover, \( \Theta = \Theta_1 \times \ldots \times \Theta_I \).

Inter-agent interactions have a number of potential *outcomes* \( O \). A given *social choice function/rule* \( f : \Theta \rightarrow O \) characterises what can be deemed to be an optimal outcome of the interaction for every vector of agent types.

Agents’ self-interest is dictated by their preferences over the outcomes, given their type, expressed in terms of (private) *utility functions* \( u_i : O \times \Theta_i \rightarrow \mathbb{R} \). The public face of agents is given by their actions, where \( \Sigma_i \) is the set of possible actions of agent \( i \) and \( \Sigma = \Sigma_1 \times \ldots \times \Sigma_I \). The decision for agent \( i \) of which action to perform is given by a *strategy*. Let \( S_i \) denote the space of possible strategies for agent \( i \), \( S = S_1 \times \ldots \times S_I \) and \( S_{-i} \) denote \( S_1 \times \ldots \times S_{i-1} \times S_{i+1} \times \ldots \times S_I \). Then a strategy \( s_i \in S_i \) is a function \( s_i : \Theta_i \times \Sigma \rightarrow \Sigma_i \). A strategy \( s = (s_1, \ldots, s_i, \ldots, s_I) \) is often represented as \((s_i, s_{-i})\) where \( s_{-i} = (s_1, \ldots, s_{i-1}, s_{i+1}, \ldots, s_I) \).

Finally, a *mechanism* \( M = (\Sigma, g) \) consists of the action space \( \Sigma \) and an *outcome function/rule* \( g : S \rightarrow O \), where \( g(s) \) is the outcome implemented by \( M \) for
strategy \( s \).

Since \( g(s) \in \mathcal{O} \), utility functions can be equivalently thought of as \( u_i : S \times \Theta_i \to \mathbb{R} \) (where \( u_i(s, \theta_i) \) stands for \( u_i(g(s), \theta_i) \)). Also, as strategies determine actions, the outcome function can be equivalently thought of as \( g : \Sigma \to \mathcal{O} \), where \( g(\sigma) \) is the outcome implemented by the mechanism for action \( \sigma \).

A social choice function specifies the desired goal of an interaction amongst agents, whereas a mechanism is a means of characterising the agents’ behaviour in the interaction. Several characterisations of strategies have been provided as ways to predict how (rational) agents will behave in a mechanism. In particular, a strategy \( s_i \) is dominant (for agent \( i \)) if it maximises the agent’s utility irrespectively of the other agents’ strategies:

\[
\forall s_{-i} \in S_{-i}, \forall s'_i \in S_i \left[ u_i((s_i, s_{-i}), \theta_i) \geq u_i((s'_i, s_{-i}), \theta_i) \right].
\]

For a mechanism \( M = (\Sigma, g) \) and a social choice function \( f \), \( M \) implements \( f \) if and only if \( g(s) = f(\theta) \), where \( s \) is a dominant strategy.

### 6.3.2 Preliminaries

In this section, upon computing the \( S \)-acceptability of a sentence \( s \) (for \( S \in \{ \text{admissible, grounded, ideal} \} \)) in a framework \( F = \langle \mathcal{L}, \mathcal{R}, \mathcal{A}, \mathcal{C} \rangle \), if \( \exists \alpha \in \mathcal{A}, \mathcal{C}(\alpha) = \{\} \) (hence \( F \) is not an ABA framework), then for all such \( \alpha \), we let \( \mathcal{C}(\alpha) = \{\text{new}\} \), where \( \text{new} \notin \mathcal{L} \) and then replace \( \mathcal{L} \) with \( \mathcal{L} \cup \{\text{new}\} \). Conceptually, this treatment states that an assumption with no known contrary is considered as an assumption with an unsupported contrary.

We also implement persuasion dialogues as a specialisation of the dialogue model described in Chapter 3 by specialising the \( ID \) field of utterances in a dialogue. Given a dialogue \( \langle u_1, \ldots, u_n \rangle = \delta \), for each \( u_i = \langle \ldots, \ldots, id_i \rangle \) in \( \delta \), we let \( id_i \in \mathbb{N} \) such that \( id_i \) (the claim) is odd; and the parity of \( id_i \), for \( u_i = \langle \ldots, t, C, id \rangle \) is the same as the parity of \( t \), if \( C \) is \( \text{rl}(\ldots) \) or \( \text{asm}(\ldots) \); or the opposite of the parity of \( t \), if \( C \) is \( \text{ctr}(\ldots) \).

Loosely speaking, this specialisation enforces that if an utterance \( u \) supports or defends the claim, then its \( ID \) is odd (we say that \( u \) is odd); otherwise, its \( ID \) is even (and we say that \( u \) is even). For any \( u \) and \( u' \), if \( u \) is odd and \( u' \) is even, we say that \( u \) and \( u' \) are of the opposite type. We assume all dialogues in the rest of this section are persuasion dialogues and follow this specialisation.

We will adopt the following notation: given \( \delta \in \mathcal{D} \), we use \( U_i^\delta \) to denote the set of regular utterances made by \( a_i \) in \( \delta \). With an abuse of notation, we say that a content \( C \) (a rule, an assumption, or a contrary) is in \( U_i^\delta \) if and only if \( C \) is the
content of some \( u \in U_i \).

Given \( F = \langle \mathcal{L}, \mathcal{R}, \mathcal{A}, \mathcal{C} \rangle \), and \( C \) either a rule, an assumption, or a contrary, we say that \( C \) is in \( F \) if and only if \( C \in \mathcal{R} \) (if \( C \) is a rule), or \( C \in \mathcal{A} \) (if \( C \) is an assumption), or \( C(\alpha) \in C(\alpha) \) (if \( C \) is a contrary).

### 6.3.3 Strategy-move Functions used in Persuasion

We define more strategy-move functions that represent the “proponent” and “opponent” roles in a persuasion dialogue. The **proponent strategy-move** function defines agents that only utter utterances that support or defend the claim (odd utterances); whereas the **opponent strategy-move** function defines agents that only utter utterances that attack the claim or some of its defences (even utterances). Pass utterances are allowed in both strategy move functions.

**Definition 50.** A **proponent strategy-move function** \( \phi \in \Phi \) for agent \( a_k \) \((k \in \{1, 2\})\) is such that, given \( \delta \in D \) such that \( \delta \) is constructed with a thorough strategy-move function with respect to \( a_k \), given \( \lambda \in \Lambda \),

- if \( \phi_h(\delta, \lambda) \subseteq PASS \), then \( \phi(\delta, \lambda) \subseteq PASS \);

- otherwise, let \( S = \{X | X \in \phi_h(\delta, \lambda), X \text{ is odd}\} \),

  if \( S = \emptyset \), then \( \phi(\delta, \lambda) \subseteq PASS \), otherwise, \( \phi(\delta, \lambda) = S \).

We refer to a generic proponent strategy-move function as \( \phi_p \).

**Definition 51.** An **opponent strategy-move function** \( \phi \in \Phi \) for agent \( a_k \) \((k \in \{1, 2\})\) is such that, given \( \delta \in D \) such that \( \delta \) is constructed with a thorough strategy-move function with respect to \( a_k \), given \( \lambda \in \Lambda \),

- if \( \phi_h(\delta, \lambda) \subseteq PASS \), then \( \phi(\delta, \lambda) \subseteq PASS \);

- otherwise, let \( S = \{X | X \in \phi_h(\delta, \lambda), X \text{ is even}\} \),

  if \( S = \emptyset \), then \( \phi(\delta, \lambda) \subseteq PASS \), otherwise, \( \phi(\delta, \lambda) = S \).

We refer to a generic opponent strategy-move function as \( \phi_o \).

Trivially, the proponent (opponent) strategy-move functions describe utterances that are truthful, thorough, and support (attack) the claim of the dialogue, respectively, as we shown in the next example.

**Example 39.** Let \( a_1 = a_2 = \langle \mathcal{L}, \mathcal{R}, \mathcal{A}, \mathcal{C} \rangle \), where:
Table 6.5: An example dialogue constructed with a $\phi_p$ and a $\phi_o$.

<table>
<thead>
<tr>
<th>$a_1$</th>
<th>$a_2$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\langle a_1,a_2,0,claim(s),1 \rangle$</td>
<td>$\langle a_2,a_1,0,\pi,2 \rangle$</td>
</tr>
<tr>
<td>$\langle a_1,a_2,1,asm(s),3 \rangle$</td>
<td>$\langle a_2,a_1,3,ctr(s,a),4 \rangle$</td>
</tr>
<tr>
<td>$\langle a_1,a_2,0,\pi,5 \rangle$</td>
<td>$\langle a_2,a_1,4,rl(a \leftarrow),6 \rangle$</td>
</tr>
<tr>
<td>$\langle a_1,a_2,0,\pi,7 \rangle$</td>
<td>$\langle a_2,a_1,0,\pi,8 \rangle$</td>
</tr>
</tbody>
</table>

- $R = \{ a \leftarrow \}$,
- $A = \{ s \}$,
- $C$ is such that $C(s) = \{ a \}$.

We let $a_1$ uses a proponent strategy-move function $\phi_p$ and $a_2$ uses an opponent strategy-move function $\phi_o$, a possible dialogue $D_{a_1}^{a_2}(s)$ is shown in Table 6.5. As we can see in this example that $a_1$ only makes utterances that support $s$ whereas $a_2$ only makes utterances that attack $s$.

### 6.3.4 Persuasion Dialogue Results

We model persuasion as follows: the persuader/proponent ($a_1$) utters arguments that support the topic ($\chi$) whereas the persuadee/opponent ($a_2$) attacks the persuader by uttering counter arguments. We equate persuasion to $\mathcal{S}$-acceptability of the topic in the ABA framework drawn from a dialogue as follows:

**Definition 52.** Given a dialogue $\delta = D_{a_1}^{a_2}(\chi)$, $a_2$ is persuaded (by $a_1$) if and only if $\chi$ is $\mathcal{S}$-acceptable in $F_\delta$, for $\mathcal{S} \in \{\text{admissible, grounded, ideal}\}$. We use PERSUADED and NOT_PERSUADED to denote that $a_2$ is persuaded or not, respectively, given a dialogue.

We link persuasion and mechanism design as follows. We define the types of agents as their ABA frameworks:

**Definition 53.** The types for agents $a_1, a_2$ are $\theta_1 = a_1$ and $\theta_2 = a_2$.

In ABA-dialogues, agents interact by putting forward rules, assumptions and contraries about a topic, conveyed as contents of utterances. We hence view frameworks as actions and dialogues as strategies, in mechanism design terms, as follows:
Definition 54. The action spaces for agents $a_1$, $a_2$ are $\Sigma_1 = \Sigma_2 = AF(L)$ (respectively).

Definition 55. Given a dialogue $D^{a_i}_{a_j}(\chi) = \delta$, the dialogue strategy $s^\delta_k$ for $a_k$ ($i, j, k = 1, 2, i \neq j$) with respect to $\delta$ is such that, given the framework $F_\delta$ drawn from $\delta$, $s^\delta_k(\theta_k, F_\delta) = \sigma_k$ where $\sigma_k = (L, R_{\sigma_k}, A_{\sigma_k}, C_{\sigma_k})$ and:

- $R_{\sigma_k} = \{ \rho | \langle a_k, \ldots, rl(\rho), \_ \rangle \text{ is in } \delta \}$,
- $A_{\sigma_k} = \{ \alpha | \langle a_k, \ldots, asm(\alpha), \_ \rangle \text{ is in } \delta \}$,
- $C_{\sigma_k}$ is such that, for every $\alpha \in A_{\sigma_k}, C_{\sigma_k}(\alpha) = \{ \beta | \langle a_k, \ldots, ctr(\alpha, \beta), \_ \rangle \text{ is in } \delta \}$.

We say that $\delta$ is the dialogue of $s^\delta = (s^\delta_1, s^\delta_2)$.

Note that the framework drawn from a dialogue is equal to the joint framework of $\sigma_1, \sigma_2$ given by the strategies of the two agents, i.e. $F_\delta = \sigma_1 \cup \sigma_2$.

Since a framework can be drawn from a dialogue, we define the outcomes of a persuasion dialogue as the ABA framework drawn from the dialogue, as follows:

Definition 56. The outcomes are $O = \{ F | F \in AF(L) \text{ and } F = F_\delta \text{ for some } \delta \in D \}$.

To define the utility functions, we first define the payment of utterances and dialogues. We then carry the payment of a dialogue to the ABA framework drawn from the dialogue. Hence the payment of agent actions are linked to the outcome.

We consider the payment of an utterance as the “cost” to the agent that makes the utterance. The payment is 0 if the utterance is considered honest by the other agent; and positive if it is considered as a lie. We treat the payment probabilistically such that the payment is computed as the product of the probability of the other agent believing the utterance is a lie and the “damage” of such a lie.

Definition 57. Let $i, j = 1, 2, j \neq i$.

Given an utterance $u \in U^i$, the payment of $u$ with respect to $a_i$ is $T_i^u = d_i^u \cdot p_i^u$ where $d_i^u \leq 0$ is the damage to agent $a_i$ if $u$ is considered a lie by $a_j$; and $0 \leq p_i^u \leq 1$ is the probability of $a_j$ considering $u$ a lie. Given an utterance $u \in U^j$, the payment of $u$ with respect to $a_i$ is $T_i^u = 0$.

Given a dialogue $\delta = \{u_1, \ldots, u_n\}$ between $a_i$ and $a_j$, let $T = \{ T_i^{u_k} | u_k \text{ is in } \delta \text{ and } T_i^{u_k} > 0 \}$. Then the payment of $\delta$ for $a_i$ is: $T_i = \max(T)$ if $T \neq \{ \}$, and $T_i = 0$ otherwise.1 Moreover, the payment of $F_\delta$ for $a_i$ is $T_i$.

1For $S = \{x_1, \ldots, x_m\} \subseteq R$, if $l = \max(S)$, then $l \in S$ and $\exists k \neq l, k \in S$ such that $k > l$.  

116
We do not address here the problem of determining the damage and probability values and assume that these are given. As it will become clear below, we use \( \max(T) = T_i \) to ensure that the payment is higher than the reward, should an agent lie (even the slightest lie would result in a negative utility).

With payments defined, we define the utility functions for agents in persuasion:

**Definition 58.** Let \( W_i^X \geq 0 \) be the *reward of topic* \( \chi \) for agent \( a_i \), for \( i = 1, 2 \). Let \( F_\delta \) be the ABA framework drawn from a dialogue \( \delta \). Then, the utility functions \( u_1, u_2 \) for \( a_1, a_2 \) respectively are defined as:

- if PERSUADED: \[ *u_1(F_\delta, \theta_1) = W_i^X + T_i, \quad *u_2(F_\delta, \theta_2) = 0, \]
- if NOT_PERSUADED: \[ *u_1(F_\delta, \theta_1) = 0, \quad *u_2(F_\delta, \theta_2) = W_i^X + T_i. \]

Since we always associate topics with dialogues, we will also say that the reward of dialogue \( \delta \) for \( a_i \) is \( W_i^X \) if \( W_i^X \) is the reward for the topic of \( \delta \).

The *outcome function* maps agent actions to outcomes as follows.

**Definition 59.** The outcome function for \( \sigma_1 \in \Sigma_1, \sigma_2 \in \Sigma_2 \) is: \[ g_p(\sigma_1, \sigma_2) = \sigma_1 \sqcup \sigma_2. \]

We show that the dialogue strategy \( s_\delta \) is a dominant strategy for agents in persuasion under the following two conditions: (1) introducing rules, assumptions, and contraries that are not in an agent’s ABA framework in the dialogue makes the payment higher than the reward of the dialogue; and uttering rules, assumptions and contraries in an agent’s ABA framework ensures that the payment is lower than the reward of the dialogue; (2) there is no overlap in content between utterances from \( a_i \) and \( a_j \).

**Theorem 7.** Given \( D_{a_1}^{\phi_p}(\chi) = \delta \), if \( \delta \) is constructed with \( \phi_p \) for \( a_1 \) and with \( \phi_o \) for \( a_2 \), then the dialogue strategy \( s_\delta \) is dominant under the conditions that:

1. \[ \bullet \text{ for all } C \text{ in } a_i \ (i = 1, 2), \text{ if } u = \langle a_i, a_j, \_, C, \_ \rangle \text{ is in } \delta \text{ then } T_i^u + W_i^X > 0; \]

2. \[ \bullet \text{ for all } C \text{ not in } a_i \ (i = 1, 2), \text{ if } u = \langle a_i, a_j, \_, C, \_ \rangle \text{ is in } \delta \text{ then } T_i^u + W_i^X < 0; \]

3. \[ \bullet \text{ for all } C_1 \text{ in } U_1^{a_i} \text{ and } C_2 \text{ in } U_2^{a_i}, \ C_1 \neq C_2. \]
Proof. Let \(\sigma_1, \sigma_2\) be actions given by \(s^\delta\). To show that \(s^\delta\) is dominant is to show that there is no other strategy which gives actions \(\sigma'_1\) and \(\sigma'_2\) such that 
\[ u_i(g(\sigma'_1, \sigma'_2), \theta_i) > u_i(g(\sigma_1, \sigma_2), \theta_i), \; i \in \{1, 2\} \]. We show this by examining properties of \(\phi_p\) and \(\phi_o\).

From the definition of \(\phi_p\), we know that an agent who uses this strategy (the proponent):

- makes utterances from its knowledge base;
- only makes utterances that support the claim; and
- hides no utterance that supports the claim.

Similarly, from the definition of \(\phi_o\), we know that an agent who uses this strategy (the opponent):

- makes utterances from its knowledge base;
- only makes utterances that attack the claim; and
- hides no utterance that attacks the claim.

In cases, we show strategies that define any other behaviours give no higher utilities for the proponent and the opponent using \(\phi_p\) and \(\phi_o\), respectively.

Firstly, we show that truthfulness yields better utility than introducing lies in a dialogue. Since \(a_1\) uses \(\phi_p\) and \(a_2\) uses \(\phi_o\), \(\sigma_i \subseteq a_i\). Hence, by condition (1), \(u_i \geq 0\). Also by condition (1), for any \(\sigma'_i\) with content not in \(a_i\), 
\[ u_i(g_p(\sigma'_1, \sigma'_2), \theta_i) \leq 0. \] Hence, any strategy that gives \(\sigma_i\) with contents not in \(a_i\) is no better than \(s^\delta\).

Secondly, we show that disclosing information that is in \(a_i\) but not in \(\sigma_i\) produces no higher utility, as \(a_1\) utters odd utterances and \(a_2\) utters even utterances that form \(\sigma_1\) and \(\sigma_2\), respectively. Since the contents of odd utterances are in arguments that support/defend the claim, whereas the content of even utterances are in arguments that attack the claim/its defences, neither agent gains higher utility by uttering utterances of the opposite type.

Thirdly, we show that disclosing less information allowed by \(\phi_p\) and \(\phi_o\) produces no higher utility for \(a_1\) and \(a_2\), respectively. By condition (2), there is no utterance by \(a_i\) with content that can be used in arguments uttered by \(a_j\). Hence, disclosing less information than allowed by \(\phi_p\) and \(\phi_o\) yields no higher utility. \(\Box\)

We define the social choice function for persuasion as follows:
Definition 60. Given a topic $\chi$, the persuasion social choice function is: $f_p(\theta_1, \theta_2) = \mathcal{F}$ defined inductively by:

- $\mathcal{F}_0$ is the framework in $\mathcal{A} \mathcal{F}(\mathcal{L})$ with empty sets of rules and assumptions;

- $\mathcal{F}_1 = \mathcal{F}_1^P \sqcup \mathcal{F}_1^O$, where $\mathcal{F}_1^P = \langle \mathcal{L}, \mathcal{R}_1^P, \mathcal{A}_1^P, \mathcal{C}_1^P \rangle$ (the proponent sub-framework of $\mathcal{F}$) and $\mathcal{F}_1^O = \langle \mathcal{L}, \mathcal{R}_1^O, \mathcal{A}_1^O, \mathcal{C}_1^O \rangle$ (the opponent sub-framework of $\mathcal{F}$), such that:
  - $\mathcal{R}_1^P = \{ X | X \in \mathcal{R}_1, X \text{ is directly related to } \chi \}$,
  - $\mathcal{A}_1^P = \{ X | X \in \mathcal{A}_1, X \text{ is directly related to } \chi \}$,
  - $\mathcal{C}_1^P$ is such that, for any $\alpha \in \mathcal{A}_1^P$, $\mathcal{C}_1^P(\alpha) = \{ \}$;
  - $\mathcal{R}_1^O = \{ \}, \mathcal{A}_1^O = \{ \}, \mathcal{C}_1^O$ is such that, for any $\alpha \in \mathcal{A}_1^O$, $\mathcal{C}_1^O(\alpha) = \{ \}$;

- Given $\mathcal{F}_i = \mathcal{F}_i^P \sqcup \mathcal{F}_i^O$, $\mathcal{F}_{i+1} = \mathcal{F}_{i+1}^P \sqcup \mathcal{F}_{i+1}^O$, where $\mathcal{F}_{i+1}^P = \langle \mathcal{L}, \mathcal{R}_{i+1}^P, \mathcal{A}_{i+1}^P, \mathcal{C}_{i+1}^P \rangle$ and $\mathcal{F}_{i+1}^O = \langle \mathcal{L}, \mathcal{R}_{i+1}^O, \mathcal{A}_{i+1}^O, \mathcal{C}_{i+1}^O \rangle$, such that:
  - $\mathcal{R}_{i+1}^P = \{ X | X \in \mathcal{R}_i, X \text{ is directly related to } \mathcal{R}_i^P \cup \mathcal{C}_i^P \}$,
  - $\mathcal{A}_{i+1}^P = \{ X | X \in \mathcal{A}_i, X \text{ is directly related to } \mathcal{R}_i^P \cup \mathcal{C}_i^P \}$,
  - $\mathcal{C}_{i+1}^P$ is such that, for any $\alpha \in \mathcal{A}_{i+1}^P$, $\mathcal{C}_{i+1}^P(\alpha) = \{ \beta | \alpha \in \mathcal{A}_i^O \text{ and } \beta \in \mathcal{C}_i(\alpha) \}$,
  - $\mathcal{R}_{i+1}^O = \{ X | X \in \mathcal{R}_i, X \text{ is directly related to } \mathcal{R}_i^O \cup \mathcal{C}_i^O \}$,
  - $\mathcal{A}_{i+1}^O = \{ X | X \in \mathcal{A}_i, X \text{ is directly related to } \mathcal{R}_i^O \cup \mathcal{C}_i^O \}$,
  - $\mathcal{C}_{i+1}^O$ is such that, for any $\alpha \in \mathcal{A}_{i+1}^O$, $\mathcal{C}_{i+1}^O(\alpha) = \{ \beta | \alpha \in \mathcal{A}_i^P \text{ and } \beta \in \mathcal{C}_i(\alpha) \}$.

The persuasion social choice function $f_p$ constructs $\mathcal{F}$. Roughly speaking, $\mathcal{F}$ is defined so that it contains rules, assumptions, and contraries that support the claim in the persuader and rules, assumptions, and contraries that attack the claim in the persuadee. Given the construction of such $\mathcal{F}$, the following theorem holds.

Theorem 8. Under conditions (1) and (2) specified in theorem 7, the mechanism $\mathcal{M} = (\Sigma, s^h)$ implements the persuasion social choice function $f_p$.

---

1 With an abuse of notation, we say that $X$ is directly related to $S$ where $X$ is a rule, an assumption, or a contrary and $S$ is a set of rules, assumptions and contraries, if and only if $X$ is directly related to some $s$ such that $s \in S$. 

119
Mechanism Design Concepts

<table>
<thead>
<tr>
<th>Concept</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type space (of agent $i$)</td>
<td>$\Theta_i$</td>
</tr>
<tr>
<td>Outcomes</td>
<td>$\emptyset$</td>
</tr>
<tr>
<td>Utility function (for agent $i$)</td>
<td>$u_i : \mathcal{O} \times \Theta_i \rightarrow \mathbb{R}$</td>
</tr>
<tr>
<td>Action space</td>
<td>$\Sigma = \Sigma_1 \times \ldots \times \Sigma_I$</td>
</tr>
<tr>
<td>Strategy (of agent $i$)</td>
<td>$s_i : \Theta_i \times \Sigma_i \rightarrow \Sigma_i$</td>
</tr>
<tr>
<td>Social choice function</td>
<td>$f : \Theta_1 \times \ldots \times \Theta_I \rightarrow \mathcal{O}$</td>
</tr>
</tbody>
</table>

Persuasion ($I = 2$)

<table>
<thead>
<tr>
<th>Concept</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type space</td>
<td>$\Theta_i$</td>
</tr>
<tr>
<td>Outcomes</td>
<td>$\emptyset$</td>
</tr>
<tr>
<td>Utility function (for agent $i$)</td>
<td>$u_i : \mathcal{O} \times \Theta_i \rightarrow \mathbb{R}$</td>
</tr>
<tr>
<td>Action space</td>
<td>$\Sigma = \Sigma_1 \times \ldots \times \Sigma_I$</td>
</tr>
<tr>
<td>Strategy (of agent $i$)</td>
<td>$s_i : \Theta_i \times \Sigma_i \rightarrow \Sigma_i$</td>
</tr>
<tr>
<td>Social choice function</td>
<td>$f : \Theta_1 \times \ldots \times \Theta_I \rightarrow \mathcal{O}$</td>
</tr>
</tbody>
</table>

Table 6.6: Dialogue as mechanism

Proof. Since $s^\delta$ is a dominant strategy (under conditions (1) and (2)), to show that $\mathcal{M}$ implements $f_p$ is to show $g_p(\sigma_1, \sigma_2) = f_p(\theta_1, \theta_2)$, i.e., $\mathcal{F} = f_p(\theta_1, \theta_2) = \sigma_1 \sqcup \sigma_2$, namely rule, assumption and contrary in $\mathcal{F}$ is also in $\sigma_1 \sqcup \sigma_2$ and vice versa. This is the case as, by the definitions of $\phi_p$ and $\phi_o$, the set of utterances made by $a_1$ constitutes the proponent sub-framework of $\mathcal{F}$ and the set of utterances made by $a_2$ constitutes the opponent sub-framework of $\mathcal{F}$. Hence, $\mathcal{F} = \sigma_1 \sqcup \sigma_2$. $\blacksquare$

We summarise the connection between mechanism design and dialogues in table 6.6.

We illustrate our results in the following example.

Example 40. In this example, we let $a_1$ and $a_2$ be as follows:

- $\mathcal{R}_1 = \{ \chi \leftarrow q_1; \chi \leftarrow q_2; q_1 \leftarrow a_1; q_2 \leftarrow a_2; c_2 \leftarrow; c_3 \leftarrow \}$,
- $\mathcal{A}_1 = \{ a_1; a_2 \}$,
- $\mathcal{C}_1$ is: $\mathcal{C}_1(a_1) = \{ c_1 \}; \mathcal{C}_1(a_2) = \{ c_2 \}$,
- $\mathcal{R}_2 = \{ \}$,
- $\mathcal{A}_2 = \{ a_1, a_2, a_3 \}$,
- $\mathcal{C}_2$ is: $\mathcal{C}_2(a_1) = \{ a_3 \}; \mathcal{C}_2(a_2) = \{ c_2 \}; \mathcal{C}_2(a_3) = \{ c_3 \}$.

The dialogue proceeds as shown in Table 6.7. The ABA framework drawn from the dialogue, $\mathcal{F}_\delta$, is the following.

- $\mathcal{R}_\delta = \{ \chi \leftarrow q_1; q_1 \leftarrow a_1; \chi \leftarrow q_2; q_2 \leftarrow a_2 \}$
- $\mathcal{A}_\delta = \{ a_1, a_2, a_3 \}$
- $\mathcal{C}_\delta$ is: $\mathcal{C}_\delta(a_1) = \{ a_3 \}; \mathcal{C}_\delta(a_2) = \{ c_2 \}$. 

120
Table 6.7: A persuasion dialogue example.

<table>
<thead>
<tr>
<th>a1:</th>
<th>a2:</th>
</tr>
</thead>
<tbody>
<tr>
<td>⟨a1, a2, 0, claim(χ), 1⟩</td>
<td>⟨a2, a1, 7, ctr(a1, a3), 8⟩</td>
</tr>
<tr>
<td>⟨a1, a2, 1, rl(χ ← q1), 3⟩</td>
<td>⟨a2, a1, 8, asm(a3), 10⟩</td>
</tr>
<tr>
<td>⟨a1, a2, 3, rl(q1 ← a1), 5⟩</td>
<td>⟨a2, a1, 15, ctr(a2, c2), 16⟩</td>
</tr>
<tr>
<td>⟨a1, a2, 5, asm(a1), 7⟩</td>
<td>⟨a2, a1, 0, π, 18⟩</td>
</tr>
<tr>
<td>⟨a1, a2, 11, rl(q2 ← a2), 13⟩</td>
<td></td>
</tr>
<tr>
<td>⟨a1, a2, 13, asm(a2), 15⟩</td>
<td></td>
</tr>
<tr>
<td>⟨a1, a2, 0, π, 19⟩</td>
<td></td>
</tr>
</tbody>
</table>

Clearly, \( F_δ \subseteq a_1 \uplus a_2 \) so both a1 and a2 have uttered information from their ABA frameworks. Since they have also hidden certain part of their knowledge base, so \( F_δ \neq a_1 \uplus a_2 \).

In a more realistic example, we can have a dialogue as follows.

**Example 41.** In this example, we have two agents, Jenny (J) and Amy (A), planning a film night together. Jenny would like to persuade Amy to watch the movie Terminator (Ter for short). The dialogue is as follows.

\[
\begin{align*}
\langle J, A, 0, claim(watchMovie(Ter)) \rangle, 1 \\
\langle J, A, 1, rl(watchMovie(Ter) ← fun(Ter), goodScreenTime(Ter)), 2 \rangle \\
\langle J, A, 2, rl(fun(Ter) ← actionMovie(Ter)), 3 \rangle \\
\langle J, A, 3, rl(actionMovie(Ter) ←), 4 \rangle \\
\langle J, A, 2, asm(goodScreenTime(Ter)), 5 \rangle \\
\langle A, J, 5, ctr(goodScreenTime(Ter), late(Ter)), 6 \rangle \\
\langle A, J, 6, asm(late(Ter)), 7 \rangle \\
\langle J, A, 7, ctr(late(Ter), finishByTen(Ter)), 8 \rangle \\
\langle J, A, 8, rl(finishByTen(Ter)), 9 \rangle \\
\langle J, A, 0, π, 10 \rangle \\
\langle A, J, 0, π, 11 \rangle 
\end{align*}
\]

A natural language reading of this dialogue is shown below.
Jenny: Let’s watch Terminator.
Jenny: We can watch Terminator, if it is fun and has the right screening time.
Jenny: I think action movies are pretty fun.
Jenny: Moreover, I know Terminator is an action movie.
Jenny: I also think Terminator starts at the right time.
Amy: Are you sure it won’t be too late?
Amy: I am just afraid Terminator starts late.
Jenny: It won’t be too late if it finishes by 10 o’clock.
Jenny: And, indeed Terminator finishes by 10 o’clock.
Jenny: OK, we can watch Terminator.
Amy: OK.

As we can see from this example, the proponent agent, Jenny, starts the dialogue and puts forward utterances that support the claim, watchMovie(Ter), whereas the opponent agent, Amy, attacks this proposal by questioning if it has the right screening time. At the end, Amy is persuaded as the claim of the dialogue is acceptable (with respect to an admissible, grounded or ideal) in the ABA framework drawn from the dialogue.

6.3.5 Related Work

Amgoud and Maudet [AM02] present an early work on studying agent strategies in persuasion dialogues. Their approach derives dialogue strategies from pre-defined agent profiles, e.g., agreeable agent that accepts everything, argumentative agent that challenge everything, etc. They have not linked dialogue results with agents’ internal beliefs.

Kakas et al. [KMM04] present a study on strategies used in agent interactions. However, the strategies in [KMM04] are more like rules than describing agent profiles. Also, the proposal of [KMM04] is not concerned with dialogues, and hence is orthogonal to our work.

Black and Katie [BA11] present a study on dialogue systems that support deliberation dialogues. Their underlying argumentation framework is the instantiated value-based argumentation framework, hence their dialogue model and results are concerned with agents with preferences. Their system relies upon agents estimating their counterparts’ preferences and does not study strategies using mechanism design.

Introducing mechanism design approach in argumentation research is a rela-
tively new phenomenon. Rahwan and Larson [RL11] present a few examples in logical mechanism design. However, the main point of this work is to demonstrate the feasibility of introducing mechanism design as a tool in the design of logical inference procedures, whereas our paper focuses on directly applying mechanism design in a particular type of argumentation based dialogue, for persuasion.

Rahwan et al. [RLT09] and Pan et al. [PLR10] have introduced the Argumentation Mechanism Design as a paradigm for studying argumentation using game-theoretic techniques. Those two papers have shown results that: for the purpose of having more arguments accepted with respect to various semantics, agents would disclose all their arguments if and only if no argument in one agent’s argumentation framework attacks any other argument in this agent’s argumentation framework, directly or indirectly. Unlike those works, our work has focused on argumentation dialogues. Furthermore, [RLT09] and [PLR10] used the abstract argumentation (AA) framework defined in [Dun95] whereas our work is based on ABA.

Caminada [Cam09] presents a study of three different types of dishonesty: lie, BS, and deception, where lie is uttering information that is directly inconsistent with one agent’s knowledge base; BS is uttering information that are made up, i.e., not necessarily inconsistent with the agent’s knowledge base, but does not exist in the agent’s knowledge base; and deception is hiding information. Our definition of lying captures the first two types of dishonesty. It can be viewed that our truthful strategy-move function rules out lie and BS. We view that deception can be purposefully allowed in persuasion.

Kraus and Schechter [KS03] present a study on collaborative agent behaviours for resource sharing. Though this study involves game theoretic aspects, it is not linked to argumentation nor formal argumentation based dialogues. Rather, it presents a study of the resource sharing problem with a game theory based approach with specific constraints.

6.3.6 Summary

In this section, we have studied strategies that agents may use in persuasion dialogues. Building upon our dialogue and model, we define strategy-move functions that describe suitable utterances for agents in persuasion. We have brought mechanism design techniques into argumentation based dialogues by mapping dialogue components into various mechanism design results. We have proved that under specified conditions, neither the persuader agent nor the persuadee agent will lie
in dialogues. We have also defined a persuasion social choice function and proved that the dialogue strategy implements such social choice function.

As an early work in introducing mechanism design in argumentation based dialogues, the two main contribution of this work are: (1) mapping various mechanism design concepts into argumentation and argumentation based dialogues; and (2) showing that the dialogue mechanism for persuasion have desired properties under specific conditions.

6.4 Conclusion

In this chapter, we instantiated our dialogue model in previous chapters for three different dialogue types: information-seeking, inquiry, and persuasion. We have demonstrated how various strategy-move functions can be deployed to represent agent behaviours in these dialogues and how techniques in mechanism design can be used in persuasion dialogues.

In the next chapter we will further illustrate how our dialogue models can be extended and applied in two other types of dialogues.
7 Dialogue Instantiations (II)

7.1 Introduction

After information-seeking, inquiry and persuasion, we study another set of dialogues, dialogues with goals that are represented in terms of predicates. In particular, we focus on conflict resolution and discovery dialogues. We show how a modified version of our dialogue framework can be applied to support these two types of dialogues.

This chapter is organised as follows. Section 7.2 formally defines the notion of dialogue goal. Section 7.3 presents our formalisation of conflict resolution dialogues and conflict resolution dialogue sequences. Section 7.4 presents our formalisation of discovery dialogues. Section 7.5 concludes.

7.2 Preliminaries

In both conflict resolution and discovery dialogues, we define a notion of goal, that could represent a belief, desire or intention agents may want to share. Thus, we define a goal in the context of the shared language $L$.

Definition 61. A goal (with respect to $L$) is of the form $\exists X \ G$ such that

- $X$ is a tuple of variables;
- there exists $\beta = \{X/t\}$ for $t$ a tuple of terms such that $G\beta \in L$.

A (goal) realisation is $G\beta \in L$ such that $\beta = \{X/t\}$.

We will leave the existential quantifier of goals implicit.

Similarly to the rest of this thesis, here we consider two agents $a_1$ and $a_2$, both are capable of representing their knowledge base in ABA. We will use $a_1, a_2$ to

\[G\beta\] stands for $G$ with all occurrences of (elements in) $X$ replaced by (corresponding elements in) $t$. (Namely $G\beta$ is a substitution).
stand for the agents and their ABA frameworks. Moreover, given \( a_i \in \{a_1, a_2\} \), 
\( a_i = a_1 \) if \( a_i = a_2 \), and \( a_i = a_2 \) if \( a_i = a_1 \).

Note that agents will typically hold different views (and so have different rules, assumptions, and contraries in their ABA frameworks), but share the same underlying language. Until section 7.3.4, we assume that if a sentence \( \beta \) in the underlying language is deemed to be an assumption by both agents (namely \( \beta \in A_1 \cap A_2 \)), then the agents agree on what the contrary is (namely \( C_1(\beta) = C_2(\beta) \)).

To simplify our presentation, in this chapter, the ABA framework agents use is in the form presented in [BDKT97], such that given an ABA framework \( \langle L, R, A, C \rangle \), \( C \) is a total mapping from \( A \) into \( L \) (rather than a total mapping from \( A \) into \( 2^L \)).

7.3 Conflict Resolution Dialogues

In this section, we consider conflicts arising when the agents share a goal, disagree on how to realize it, and would benefit from identifying a joint realization. The disagreement may occur for two reasons. Firstly, agents have incomplete information, and may reason with different assumptions to fill gaps in the information available to them. Since some assumptions may be incorrect, agents may be misinformed and decide on incompatible realisations that lead to conflict. Secondly, agents may have different views of what makes a suitable realisation of the goal.

In this setting, to resolve conflicts between the agents is to build a shared ABA framework where both agents’ views are taken into account and misunderstandings eliminated. We show how this can be achieved by deploying, in a tailored manner, our dialogue frameworks.

We prove formally that (a specific form of) conflict resolution results from identifying, by means of argumentation dialogues, an admissible set of arguments. Thus, our dialogues provide a fully distributed solution to the conflict resolution problem while at the same time linking to a well-known argumentation semantics.

7.3.1 Motivating Example

We reuse the two agents watching moving scenario shown in Example 41 with a few changes in the setting. In this example, we have two movies: Lord of the Rings (LoR) and Terminator (Ter) that are both screening.

Jenny wants to pick a fun movie, and finds action movies fun. She does not want to watch a movie she has already seen. Jenny believes Ter is an action movie and,
since she has not seen it before, wants to watch it.

Amy also wants to watch a fun movie and does not want to watch a movie she has already seen, but she finds fantasy, rather than action, movies fun. Amy has watched the trailer of LoR and believes it is both an action and a fantasy movie. Given that she has never seen it before, Amy concludes she wants to watch LoR.

After exchanging information and finding out that LoR is an action movie, Jenny agrees to watch it. Thus, LoR is a conflict resolution.

We will model conflict resolution using a sequence of (pairs of) dialogues. We give here an informal presentation of the first dialogue in the sequence for this example. For illustration purposes, we present this dialogue in natural language.

J: Let’s see if Terminator is a good movie to watch.
J: I will watch Terminator if it is fun and there is no objection to it.
J: Terminator would be fun if it is an action movie.
A: Yes, Terminator is an action movie.
J: I propose we watch Terminator then.
A: We can watch it unless we have watched it before.
J: OK, I have not.
A: OK. I have not either.

Since Terminator satisfies Jenny, we move to the second dialogue, in which Terminator’s satisfiability for Amy is examined. Now, Amy starts the dialogue and the two agents proceed similarly as in the previous dialogue, except this time Amy believes fantasy movies are fun and Terminator is not a fantasy movie. Hence the dialogue fails. Since Terminator is rejected by Amy, Amy starts the next dialogue, proposing Lord of the Rings. Using this and another dialogue, the agents identify Lord of the Rings as mutually agreeable.

7.3.2 Conflict and Conflict Resolution Agents

We have defined goal and goal realisation in Section 7.2, in our motivating example, the goal is watchMovie(X), and goal realisations may be watchMovie(Ter) (for \{X/Ter\}) and watchMovie(LoR) (for \{X/LoR\})².

We assume that agents are rational, in the sense that they adopt a goal realisation only if they have reasons to do so, in terms of their internal views. As discussed earlier, we abstract the internal representation of agents’ views in terms of ABA

²In this section, X, Y etc. are (tuples of) variables; rule/assumptions/contrary schemata (with variables) are used to stand for the set of all their instances with respect to constants.
frameworks.

Rational (ABA) agents would adopt or propose a goal realisation if this is admissible to them:

**Definition 62.** A rational (goal) realisation for agent $a_i$ is a goal realisation that is admissible in $a_i$.

Note that our notion of rationality is rather weak, in that it solely requires for a realisation to stand against (internal) criticism, in the dialectical sense given by the admissibility semantics for argumentation. In our motivating example, the goal realisations watchMovie(Ter) and watchMovie(LoR) are rational for Jenny and Amy, respectively.

Agents conflict when they adopt different rational goal realisations:

**Definition 63.** A conflict (between $a_1$ and $a_2$) with respect to a goal $G$ is a pair $(G_{β_1}, G_{β_2})$ such that $G_{β_1} ≠ G_{β_2}$ and, for $i = 1, 2$, $G_{β_i}$ is a rational realisation with respect to $a_i$.

In order for agents to come to an agreement and resolve conflicts, we assume that they are cooperative and willing to concede on some of their views. This amounts to the agents envisaging that their views may be imperfect and incomplete. However, agents may still want to be committed to some of their views, especially if they reflect their desires. We thus assume that the rules in our agents’ ABA frameworks are of two types: concession rules and non-concession rules.

**Definition 64.** A conflict resolution agent is equipped with an ABA framework $AF = \langle L, R, A, C \rangle$ such that

- $R = R^C \cup R^{NC}$ and
- $R^C \cap R^{NC} = \{\}$.

We refer to $R^C$ as concession rules and to $R^{NC}$ as non-concession rules.

Note that concession and non-concession rules are syntactically indistinguishable, but in principle we could allow different syntactical formats for them, rather than separating them out in different subsets of the rule set.

In our example, a split of Jenny’s and Amy’s rules is in Table 7.1. This split reflects that agents are not willing to compromise in what they deem to be fun or
Non-Concession Rules:

\[
\text{watchMovie}(X) \leftarrow \text{fun}(X), \text{selectMovie}(X) \quad (\text{for J, A}) \\
\text{fun}(X) \leftarrow \text{actionMovie}(X) \quad (\text{for J}) \\
\text{fun}(X) \leftarrow \text{fantasyMovie}(X) \quad (\text{for A})
\]

Concession Rules:

\[
\text{actionMovie}(\text{Ter} \leftarrow) \quad (\text{for J, A}) \\
\text{fantasyMovie}(\text{LoR} \leftarrow) \quad (\text{for A}) \\
\text{actionMovie}(\text{LoR} \leftarrow) \quad (\text{for A})
\]

Table 7.1: Jenny’s and Amy’s Concession and Non-Concession Rules

the fact that they want to watch a fun movie, but they are willing to change their mind as to which movies are available options and what types they are.

A conflict resolution needs to satisfy all non-concession rules of agents, under the condition that both agents are aware of the other agent’s relevant concession rules. Thus, non-concession rules express more preferred views.

**Definition 65.** A conflict resolution \( G \beta \) for a conflict \((G \beta_1, G \beta_2)\) with respect to \( G \) is a rational realisation with respect to \( a'_1 = \langle L, R_1 \cup R_2, A, C \rangle \) and \( a'_2 = \langle L, R_2 \cup R_1, A, C \rangle \) where:

- \( A = A_1 \cup A_2 \);
- \( C(\alpha) = \begin{cases} 
  c & \text{if } \alpha \in A_1 \cap A_2, \text{ where } C_1(\alpha) = C_2(\alpha) = c, \\
  C_1(\alpha) & \text{if } \alpha \in A_1 \setminus A_2, \\
  C_2(\alpha) & \text{if } \alpha \in A_2 \setminus A_1.
\end{cases} \)

Note the second bullet is needed as agents have different assumptions. In our motivating example, where the conflict is:

\(\text{(watchMovie(Ter), watchMovie(LoR))}\),

the conflict resolution is: watchMovie(LoR).

Basically, a conflict resolution amounts to each agent taking into account all their individual rules as well as the concession rules of the other agent. This corresponds to agents being prepared to concede what they can, but not what they are absolutely inflexible about. As an extreme case, if both agents have an empty set of concession rules (namely agents are not prepared to compromise at all), then a conflict resolution is a realisation that is individually rational for each agent,
without taking any of the other agent’s views into account. At the other end of the spectrum, if both agents have all their rules down as concession rules (namely agents are prepared to compromise on everything), then a conflict resolution is a realisation that is rational in the union of the agents’ views.

Note that, in the definition of conflict resolution, the agents’ assumptions are joined. In general, since agents may not agree on what can be assumed and what can be argued for, this may cause that one or both of the $a_i'$ is not flat. For simplicity, until section 7.3.4, we assume that $a_i'$, for $i, j = 1, 2, j \neq i$, is flat.

Note that our notion of conflict resolution is in some sense “centralised”, as it assumes that agents are prepared to share their concession rules, assumptions, and contraries. By means of our specialised conflict resolution dialogues, we will provide an equivalent, distributed solution.

7.3.3 Conflict Resolution Dialogues and Sequences

As seen in Chapter 6, we define strategy-move functions to specify agent behaviours that advance the dialogue goal.

**Definition 66.** A conflict resolution strategy-move function, $\phi \in \Phi$ for agent $a_k$ ($k \in \{1, 2\}$) is a truthful strategy-move function such that, given a dialogue $\delta = D_{a_j}^n(\chi)$ and a legal-move function $\lambda \in \Lambda$, for all $u \in \phi(\delta, \lambda)$ made by $a_k$,

- for $a_k = a_i$, $\phi(\delta, \lambda) = \phi_h(\delta, \lambda)$ where $\phi_h$ is a thorough strategy-move function;
- for $a_k = a_j$, for all $u \in \phi(\delta, \lambda)$ made by $a_k$, if $u$ is a pass-utterance then there exists no regular utterance $u' = \langle a_j, a_k, C, \bot \rangle \in \lambda(\delta) \cap U^k$, where $C$ is either $asm(\bot)$, $rl(\rho)$ where $\rho \in R^C_j$, or $ctr(\bot)$ such that $\delta \circ u$ is constructed with a truthful strategy-move function.

We use $\phi_{cr}$ to denote a generic conflict resolution strategy-move function.

The conflict resolution strategy-move defined above is based on the thorough strategy-move function given in Definition 46. Conceptually, a $\phi_{cr}$ is a $\phi_h$ with limited utterances in rules. Hence, for the two agents use $\phi_{cr}$ in a dialogue, the agent who starts the dialogue is bound to utter all topic-relevant utterances from its knowledge base; whereas the other agent is bound to make all topic-relevant utterance from its assumptions, contraries, and concession rules.
We also specialise the notion of dialogues given in Chapter 3 to conflict resolution dialogues. We require a conflict resolution dialogue to start with a realisation of the agents’ goal. We also let both agents using $\phi_{cr}$ in a conflict resolution dialogue.

**Definition 67.** A conflict resolution dialogue is a dialogue $D_{a_i}^a(\mathcal{G}\beta) = \delta$ such that both $a_i$ and $a_j$ use $\phi_{cr}$ in $\delta$. We refer to such a dialogue as $cD_{a_i}(\mathcal{G}\beta)$.

We also refer to $a_i$ and $a_j$ in $cD_{a_i}(\mathcal{G}\beta)$ as the nominator and challenger, respectively.

Basically, in a conflict resolution dialogue, the challenger can only utter its concession rules, whereas the nominator can utter rules of either types.

In order to resolve conflicts, several conflict resolution dialogues may be needed in sequence:

**Definition 68.** A successful conflict resolution dialogue sequence (successful sequence in short) between $a_i$ and $a_j$ with respect to a goal $\mathcal{G}$ is a sequence

$$\delta_1 = cD_{a_i}^a(\mathcal{G}\beta_1), \delta_2 = cD_{a_i}^a(\mathcal{G}\beta_1),$$

$$\cdots,$$

$$\delta_{2n-1} = cD_{a_i}^a(\mathcal{G}\beta_n), \delta_{2n} = cD_{a_i}^a(\mathcal{G}\beta_n),$$

such that $n > 0$, $\omega_{scc}(\delta_{2n-1}, \lambda_{ADM}) = \omega_{scc}(\delta_{2n}, \lambda_{ADM}) = true$ and, for every $k < n$, $\omega_{scc}(\delta_{2k-1}, \lambda_{ADM}) \neq true$ or $\omega_{scc}(\delta_{2k}, \lambda_{ADM}) \neq true$.

$\mathcal{G}\beta_n$ is the result of the conflict resolution dialogue sequence.

**Theorem 9.** Given a conflict $(\mathcal{G}\beta_1, \mathcal{G}\beta_2)$ between $a_1$ and $a_2$ with respect to some goal $\mathcal{G}$, a conflict resolution $\mathcal{G}\beta$ exists if there is a successful conflict resolution dialogue sequence between $a_1$ and $a_2$ with respect to $\mathcal{G}$.

**Proof.** Let the successful conflict resolution dialogue sequence be $\delta_1, \delta_2, \ldots$, $\delta_{2n-1}, \delta_{2n}$. Given $\omega_{scc}(\delta_{2n-1}, \lambda_{ADM}) = \omega_{scc}(\delta_{2n}, \lambda_{ADM}) = true$, we know $\mathcal{G}\beta_n$ is admissible in both the ABA framework $\mathcal{F}_{\delta_{2n-1}}$ drawn from $\delta_{2n-1}$ and the ABA framework $\mathcal{F}_{\delta_{2n}}$ drawn from $\delta_{2n}$ by Theorem 1.

Given both $\delta_{2n-1}$ and $\delta_{2n}$ are constructed with $\phi_{cr}$, and $\phi_{cr}$ is a thorough strategy-move with a limited utterance in rules, $\mathcal{F}_{\delta_{2n-1}}$ is a $\mathcal{G}\beta_n$-related framework of $a_1'$ and $\mathcal{F}_{\delta_{2n}}$ is a $\mathcal{G}\beta_n$-related framework of $a_2'$. Hence, by Theorem 6, $\mathcal{G}\beta_n$ is admissible in $a_1'$ and $a_2'$. Hence this theorem holds.

The successful sequence in our motivating example is composed of four conflict resolution dialogues:
\[\langle J, A, 0, claim(\text{watchMovie(Ter)}), 1 \rangle\]
\[\langle J, A, 1, rl(\text{watchMovie(Ter)} \leftarrow \text{fun(Ter)}, \text{selectMovie(Ter)}), 2 \rangle\]
\[\langle J, A, 2, rl(\text{fun(Ter)} \leftarrow \text{actionMovie(Ter)}), 3 \rangle\]
\[\langle A, J, 3, rl(\text{actionMovie(Ter)} \leftarrow), 4 \rangle\]
\[\langle J, A, 2, asm(\text{selectMovie(Ter)}), 5 \rangle\]
\[\langle A, J, 5, ctr(\text{selectMovie(Ter)}, \text{seen(Ter)}), 6 \rangle\]
\[\langle J, A, 0, \pi, 7 \rangle\]
\[\langle A, J, 0, \pi, 8 \rangle\]

Table 7.2: A conflict resolution dialogue example (part 1).

\[\langle A, J, 0, claim(\text{watchMovie(Ter)}), 1 \rangle\]
\[\langle A, J, 1, rl(\text{watchMovie(Ter)} \leftarrow \text{fun(Ter)}, \text{selectMovie(Ter)}), 2 \rangle\]
\[\langle J, A, 2, asm(\text{selectMovie(Ter)}), 3 \rangle\]
\[\langle A, J, 2, rl(\text{fun(Ter)} \leftarrow \text{fantasyMovie(Ter)}), 4 \rangle\]
\[\langle J, A, 0, \pi, 5 \rangle\]
\[\langle A, J, 0, \pi, 6 \rangle\]

Table 7.3: A conflict resolution dialogue example (part 2).

\[\text{cD}_J^A(\text{watchMovie(Ter)}) = \delta_1, \text{cD}_J^A(\text{watchMovie(Ter)}) = \delta_2, \]
\[\text{cD}_J^A(\text{watchMovie(LoR)}) = \delta_3, \text{cD}_J^A(\text{watchMovie(LoR)}) = \delta_4.\]

Among them, \(\delta_2\) is not a successful dialogue, whereas the others are successful. \(\delta_1\) is in Table 7.2. \(\delta_2\) is similar to \(\delta_1\), except that the role of nominator and challenger are swapped. The dialogue terminates unsuccessfully after Amy states that Ter would be fun for her if it is a fantasy movie (as shown in Table 7.3). \(\delta_3\) is similar to \(\delta_1\), except that Amy states that LoR is an action move (Table 7.4). \(\delta_4\) is similar to \(\delta_3\) with the roles of nominator and challenger swapped (Table 7.5).

### 7.3.4 An Extension

We have assumed, from section 7.3.2, that agents agree on the contraries of assumptions they share. This is the case in our motivating example, where Jenny and Amy agree that the contrary of selectMovie(X) is seen(X). However, this may not always be the case and agents may have different views as to what the contrary of an assumption is. For example, some agent may believe that violent(X) is the contrary of selectMovie(X), to express that it does not want to watch violent movies. The notion of conflict resolution can be generalised to take this into account, as

132
Table 7.4: A conflict resolution dialogue example (part 3).

\[
\langle J, A, 0, \text{claim}(\text{watchMovie}(\text{LoR})), 1 \rangle
\]
\[
\langle J, A, 1, \text{rl}(\text{watchMovie}(\text{LoR}) \leftarrow \text{fun}(\text{LoR}), \text{selectMovie}(\text{LoR})), 2 \rangle
\]
\[
\langle J, A, 2, \text{rl}(\text{fun}(\text{LoR}) \leftarrow \text{actionMovie}(\text{LoR})), 3 \rangle
\]
\[
\langle J, A, 3, \text{rl}(\text{actionMovie}(\text{LoR}) \leftarrow ), 4 \rangle
\]
\[
\langle J, A, 2, \text{asm}(\text{selectMovie}(\text{LoR})), 5 \rangle
\]
\[
\langle A, J, 5, \text{ctr}(\text{selectMovie}(\text{LoR}), \text{seen}(\text{LoR})), 6 \rangle
\]
\[
\langle J, A, 0, \pi, 7 \rangle
\]
\[
\langle A, J, 0, \pi, 8 \rangle
\]

Table 7.5: A conflict resolution dialogue example (part 4).

\[
\langle A, J, 0, \text{claim}(\text{watchMovie}(\text{LoR})), 1 \rangle
\]
\[
\langle A, J, 1, \text{rl}(\text{watchMovie}(\text{LoR}) \leftarrow \text{fun}(\text{LoR}), \text{selectMovie}(\text{LoR})), 2 \rangle
\]
\[
\langle A, J, 2, \text{rl}(\text{fun}(\text{LoR}) \leftarrow \text{fantasyMovie}(\text{LoR})), 3 \rangle
\]
\[
\langle A, J, 3, \text{rl}(\text{fantasyMovie}(\text{LoR}) \leftarrow ), 4 \rangle
\]
\[
\langle J, A, 2, \text{asm}(\text{selectMovie}(\text{LoR})), 5 \rangle
\]
\[
\langle A, J, 5, \text{ctr}(\text{selectMovie}(\text{LoR}), \text{seen}(\text{LoR})), 6 \rangle
\]
\[
\langle J, A, 0, \pi, 7 \rangle
\]
\[
\langle A, J, 0, \pi, 8 \rangle
\]
Definition 69. A (generalised) conflict resolution $G\beta$ for a conflict $(G\beta_1, G\beta_2)$ with respect to $G$ is a rational realisation with respect to $a'_1 = \langle L', R_1 \cup R_2^C \cup R', A, C \rangle$ and $a'_2 = \langle L', R_2 \cup R_1^C \cup R', A, C \rangle$ where:

- $A = A_1 \cup A_2$;
- $C$ is defined as follows:
  1. let $C'$ be such that $C' (\alpha) = \{C_1 (\alpha), C_2 (\alpha)\}$ ($\alpha \in A$);
  2. $C (\alpha) = \begin{cases} 
  c & \text{if } C' (\alpha) = \{c\} \\
  c' & \text{if } C' (\alpha) = \{c_1, c_2\}
  \end{cases}$
  where $c' \not\in L$ and such that $C (\alpha) \neq C (\beta)$ if $\alpha \neq \beta$;
- $L' = L \cup \bigcup_{\alpha \in A} \{c' | C' (\alpha) = \{c_1, c_2\} \text{ and } C (\alpha) = c'\}$;
- $R' = R \cup \bigcup_{\alpha \in A} \{c' \leftarrow c_1, c' \leftarrow c_2 | C' (\alpha) = \{c_1, c_2\} \text{ and } C (\alpha) = c'\}$.

Here, the $L'$ and $R'$ components of the $a'_i$s are introduced to guarantee that assumptions in the $a'_i$s have a single contrary. Trivially, if different agents agree on the contrary of assumptions to start with, as we have assumed in section 7.3.2, $L'$ and $R'$ are empty and the notion of generalised conflict resolution amounts to the earlier notion.

Theorem 9 still holds under this generalisation.

7.3.5 Related Work

Tessier et. al [TCM01] present a collection of papers that study various aspects of conflicts between agents, such as the definition and categorisation of conflicts [TLFC01] and conflicts among collaborative agents [Cha01]. According to their classification, our work is about resolving knowledge conflicts. Our work thus differ from work, such as [RPSD07], where agents have conflicts of interest.

Several authors have demonstrated the versatility of various argumentation frameworks for conflict resolution, e.g. [AK07, BH08, RS09]. Instead of representing conflicts with a single argumentation framework, our approach resolves conflicts through a series of dialogues, where each dialogue is constructed from two argumentation frameworks. Moreover, differently from others, we show how arguments can be constructed and debated, for resolving conflicts, putting together in-
formation held by different agents, rather than simply exchanging fully constructed arguments (e.g. if using abstract argumentation).

Amgoud et al [ABP05] present a work on consensus forming. This follows the classical view of decision making under uncertainty while we see decision making as inference in which the different decisions are considered as assumptions.

In [FT12b], we have presented an earlier version of this conflict resolution dialogue work giving similar results, e.g., a version of Theorem 9 has been presented in [FT12b]. However, there, we have based our results purely on legal-move functions, e.g., we have defined truthful legal-move function and conflict resolution legal-move function in [FT12b] in place for truthful strategy-move function and conflict resolution strategy-move function given in this thesis, respectively. Clearly, the new setting presented in this thesis shows an improvement as these concepts should be modelled as agent behaviours rather than dialogue protocols.

7.3.6 Summary

We have presented a two-agent conflict resolution mechanism supported by dialogues. Agents are in conflict because they propose different realisations of the same goal. Through dialogues, information is shared during the dialogue and the resulting realisation satisfy both agents. We use our dialogue model defined in Chapter 3 but tailored to the needs of conflict resolution agents. In particular, the agents exchange, during dialogues, only information they are prepared to concede. Moreover, they need to engage in sequences of an even number of dialogues, taking turns in playing roles of nominator and challenger. Finally, agents need to be truthful, in contributing only views they actually hold, but only insofar as bringing into dialogues everything they are aware of that is relevant to the conflict.

We have defined a form of rational conflict resolution based on the admissibility semantics for argumentation, and made use of results in Chapters 4 and 5 to prove that successful sequences of dialogues resolve conflicts.

7.4 Discovery Dialogues

In addition to the six dialogue types described in Chapter 2, McBurney and Parsons [MP01] introduce discovery as an additional type of dialogue, different from other types in that:

“[discovery dialogues] discover something not previously known; the
question whose truth is to be ascertained may only emerge in the course of the dialogue.”

In this section, we show how our dialogue framework can be instantiated to support a special type of discovery dialogues.

Unlike information-seeking and inquiry dialogues, where dialogues starting from a known proposition, in discovery dialogues, there may be no such known proposition to start with. Instead, the dialogue participants face an open problem; they must decide on an abstract description of the goal of the dialogue and proceed by putting forward information that may contribute to identify the proposition and determine its acceptability.

In this section, we focus on a particular type of discovery dialogue, in which the two participating agents start from the same abstract description of the proposition. We call this abstract description the goal. None of the two agents have sufficient information to produce an acceptable concrete realisation of this goal. The agents’ task is then to discover an acceptable goal realisation.

In order to perform the joint discovery, the dialogue starts by one agent putting forward a sentence. Then the two agents can either expand this sentence in a top-down manner to explore its supports and attacks or bottom-up inference to identify any “higher level” arguments that are supported by it.

### 7.4.1 Another Movie Example

We again alter the two agents watching a movie example shown in Section 7.3.1 to illustrate our concept of discovery.

**Example 42.** Two agents, Jenny and Amy, are planning a film night. They would like to jointly decide on a movie. Jenny wants to pick a fun movie. She finds action movies fun. Jenny also worries about going home late so she prefers a movie that finishes by 10 o’clock. Amy knows that *Terminator* is screening and is an action movie. She also knows that *Terminator* finishes by 10 o’clock. Amy does not have any preference in selecting a movie. In order to reach agreement, the two agents may conduct a dialogue as follows.
Jenny: Let’s find a movie to watch.
Amy: Sure, I know Terminator is an action movie.
Jenny: That’s interesting. I think action movies are pretty fun.
Amy: We can watch Terminator, if it is fun and has the right screening time.
Jenny: Agreed. I think Terminator starts at the right time.
Amy: Are you sure it won’t be too late?
Jenny: Why?
Amy: I don’t know. I am just afraid so.
Jenny: It won’t be too late if it finishes by 10 o’clock.
Amy: I see. Indeed Terminator finishes by 10 o’clock.
Jenny: OK.
Amy: OK.

Jenny starts the dialogue by putting forward the goal of determining some movie to watch. Then Amy supplies one possibly relevant fact, that Terminator is an action movie. This is just a guess, in the sense that the agent does not know whether a goal realisation can be found by exploring information related to Terminator. From this utterance, agents reason bottom-up until Amy’s second utterance. Then they start top-down. From the initial guess, the dialogue constructs arguments both for and against watching Terminator. After examining the arguments, the agents decide that Terminator is a good movie to watch.

7.4.2 Discovery Dialogue Formulation

To implement discovery, we specialise the definition of utterance (Definition 1) so it accommodates a modified content field and a richer ID, as follows.

Definition 70. An utterance from agent $a_i$ to agent $a_j$ ($i, j \in \{1, 2\}, i \neq j$) with respect to $L$ is a tuple $\langle a_i, a_j, \text{Target}, C, ID \rangle$, where:

- $C$ (the content) is of one of the following forms:
  1. $\text{goal}(G)$ for some $G$ such that $\exists X G$ is a goal;
  2. $rl(s_0 \leftarrow s_1, \ldots, s_m)$ for some $s_0, \ldots, s_m \in L$ (a rule);
  3. $\text{asm}(a)$ for some $a \in L$ (an assumption);
  4. $\text{ctr}(a, s)$ for some $a, s \in L$ (a contrary);
  5. a pass sentence $\pi$, such that $\pi \notin L$. 

137
• $ID$ is a pair $[N, R]$, where $N \in \mathbb{N}$ and $R$ is either $td$ (top-down), $bu$ (bottom-up) or $nr$ (not-related) (the identifier).

• $Target \in \mathbb{N}$ (the target); $Target \leq ID$.

In the above definition, we overload "<" and "=" when comparing a target and an ID such that given a target $t$ and an ID $id = [n, r]$, $t < id$ if and only if $t < n$ and $t = id$ if and only if $t = n$. This definition is based on Definition 1, but (1) this definition adds the new related field ($R$) in an identifier to indicate different utterance relations ($td$, $bu$ or $nr$); (2) there is no “claim” used in this definition; and (3) it allows $goal(\_)$ as the content of an utterance.

Similarly to previous chapters, we refer to an utterance with content other than $\pi$ and $goal(\_)$ as regular-utterance.

We then redefine the notion of top-down related (Definition 11) as follows:

**Definition 71.** For two utterances $u_i \neq u_j$, $u_j$ is top-down related to $u_i$ if and only if $u_i = \langle \_ , \_ , C_i , [id , \_] \rangle$ and $u_j = \langle \_ , \_ , id , C_j , \_ , td \rangle$, and one of the following holds:

1. $C_j = rl(\rho_j)$, $Head(\rho_j) = h$ and either $C_i = rl(\rho_i)$ with $h \in Body(\rho_i)$, or $C_i = ctr(\_ , h)$;

2. $C_j = asm(a)$ and either $C_i = rl(\rho)$ with $a \in Body(\rho)$, or $C_i = ctr(\_ , a)$;

3. $C_j = ctr(a , \_)$ and $C_i = asm(a)$.

This modification does not consider claim utterances (as these are not allowed here). Intuitively, an utterance is top-down related to another if its target is the identifier of the latter and it contributes to expanding an argument (cases 1), identifies an assumption in the support of an argument (cases 2) or starts the construction of a counter-argument (case 3).

Similarly, we define the notion of bottom-up related as follows:

**Definition 72.** For two utterances $u_i \neq u_j$, $u_j$ is bottom-up related to $u_i$ if and only if $u_i = \langle \_ , \_ , C_i , [id , \_] \rangle$ and $u_j = \langle \_ , \_ , id , C_j , \_ , bu \rangle$, and one of the following holds:

1. $C_i = rl(\rho_i)$, $C_j = rl(\rho_j)$, and $Head(\rho_i) \in Body(\rho_j)$;

2. $C_i = asm(a)$, $C_j = rl(\rho)$, and $a \in Body(\rho)$.
Intuitively, an utterance is bottom-up related to another if its target is the identifier of the latter and it forms a “higher level” argument supported by its target. In this section, we say that an utterance $u_j$ is related to $u_i$ if $u_j$ is either top-down or bottom-up related to $u_i$. Also, no pass-utterance can be related to a regular-utterance and no utterance can be related to a pass-utterance.

After defining utterances, we alter Definition 2 to define discovery dialogues, as follows.

**Definition 73.** A discovery dialogue $dD^a_{a_j}(G)$ (between $a_i$ and $a_j$, for goal $G$ with respect to $L$), $i, j \in \{1, 2\}$, $i \neq j$, is a finite sequence $\langle u_1, \ldots, u_n \rangle$, $n \geq 0$, where each $u_l$, $l = 1, \ldots, n$, is an utterance from $a_i$ or $a_j$ (with respect to $L$), $u_1$ is an utterance from $a_i$, and:

1. the content of $u_l$ is $goal(G)$ if and only if $l = 1$;
2. the content of $u_2$ is either $rl(\_)$ or $asm(\_)$;
3. $u_1$ and $u_2$ are of the form $\langle \_, \_0, \_, [\_, nr] \rangle$;
4. the target of pass-utterances is 0;
5. each regular-utterance $u_l$, $l > 2$, is related to its target utterance;
6. no two consecutive utterances are pass-utterances, other than possibly the last two utterances, $u_{n-1}$ and $u_n$;
7. the identifier of $u_i$ is $[i, \_]$.

This definition requires dialogues to start with a goal (the first utterance). The second utterance must be a rule or an assumption. Agents make this utterance with a “wild guess” in the hope that a goal realisation can be found by exploring around this guess. All subsequent regular-utterance must be related to some earlier utterance in the dialogue. This definition is a variant of Definition 2 with dialogues starting with a goal rather than a claim and the second utterance being not related to the first. An example dialogue is given below.
An informal reading of this dialogue is given in Section 7.4.1. In this section, \(\mathcal{U}\) and \(\mathcal{D}\) stand for the sets, respectively, of all utterances as in Definition 70 and of all dialogues as in Definition 73.

Since discovery dialogues (Definition 73) are specialised dialogues (Definition 2), we will not redefine concepts such as the ABA framework drawn from a dialogue (Definition 5), legal-move functions (Definition 6), and outcome functions (Definition 8). Rather, we will reuse these concepts as they are defined for discovery dialogues. Moreover, for simplicity, we will also assume that all (discovery) dialogues in discussion are compatible with a one-way expansion legal-move function (Definition 30). Moreover, as we illustrated in the above example, we support one and only one “wild guess” in a discovery dialogue.

### 7.4.3 Debate Trees Revisited

To ensure our discovery dialogue computes admissible results for a given goal, we again use debate trees as the commitment store to keep track of information that has been disclosed in dialogues.

When constructing a debate tree from a discovery dialogue, we use a subset of utterances presented in the dialogue. Similar to Example 11, this extraction ignores the goal- and pass-utterances, i.e. a debate tree is extracted from the goal-\(\pi\)-pruned sequence obtained from a dialogue, consisting of all regular-utterances. Furthermore, by Definition 73, for all utterances \(u = (\cdots, \cdots, [i, R])\) in a goal-\(\pi\)-pruned sequence, if \(i > 1\), then \(R\) is either \(td\) or \(bu\).
We show the following process constructs a debate-tree from a discovery dialogue.

**Lemma 14.** Given a discovery dialogue $d D_{a_i}^D (\chi) = \delta$, the goal-$\pi$-pruned sequence $\delta' = \langle u'_1, \ldots, u'_m \rangle$ obtained from $\delta$, let $T^0(\delta), T^1(\delta), \ldots, T^m(\delta)$ constructed from $\delta'$ as defined below constructs a tree, then $T^m(\delta)$ is a debate tree $(L, L' \in \{P, O\}, L \neq L')$ if and only if $T^0(\delta), T^1(\delta), \ldots, T^m(\delta)$ are constructed as follows.

1. $T^0(\delta)$ is empty;

2. $T^1(\delta)$ is constructed as follows,
   a) if the content of $u'_1$ is $asm(\alpha)$, then $T^1(\delta)$ consists only of $(\alpha, ma : P[2]);$
   b) if the content of $u'_1$ is $rl(h \leftarrow b_1, \ldots, b_l)$, then $T^1(\delta)$ consists of $l + 1$ nodes, where $(h, mr : P[2])$ is a new node and
   
   $(b_1, um : P[2]), \ldots, (b_l, um : P[2])$
   
   are children of $(h, mr : P[2])$.

3. Let $T^i(\delta)$ be the $i$-th tree, for $0 < i < m$, let $u'_{i+1} = \langle \omega, t, C, [id, R] \rangle$, and let $u'_i = \langle \omega, C, t, [t, \_] \rangle$ be the target utterance of $u'_{i+1}$; then $T^{i+1}(\delta)$ is obtained according to one of the following cases:
   a) If $R = td$, then
      • if $C = rl(h \leftarrow b_1, \ldots, b_l)$ then $T^{i+1}(\delta)$ is $T^i(\delta)$ with additional $l$ nodes:
        
        $(b_1, um : L[id]), \ldots, (b_l, um : L[id])$
        
        as children of the node $(h, um : L[t])$, and this node is replaced by $(h, mr : L[id])$.
      • if $C = asm(\alpha)$ then $T^{i+1}(\delta)$ is $T^i(\delta)$ with the node $(\alpha, um : L[t])$ replaced by $(\alpha, ma : L[id])$.
      • if $C = ctr(\alpha, c)$ then $T^{i+1}(\delta)$ is $T^i(\delta)$ with an additional node:
        
        $(c, um : L[id])$ child of $(\alpha, ma : L'[t])$, where $L, L' \in \{P, O\}, L \neq L'$.
   b) If $R = bu$, then
\[ C = r l(h \leftarrow b_1, \ldots, b_l), \quad T^{i+1}(\delta) \text{ is } T^i(\delta) \text{ with } l \text{ additional nodes,} \]
in which there is a node \((h, mr : L[id]), \text{ parent of } (b_t, F : L[l])\),
such that

- if \(C_t = r l(h' \leftarrow b_1', \ldots, b_k'), \text{ then } b_t = h', F = mr;\)
- if \(C_t = asm(\alpha) \), then \(b_t = \alpha, F = ma;\)

and the remaining \(l - 1\) nodes are
\[
(b_1'', um : L[id]), \ldots, (b_{l-1}'', um : L[id]),
\]
children of \((h, mr : L[id]), \text{ where } \{b_1'', \ldots, b_{l-1}''\} = \{b_1, \ldots, b_l\} \setminus \{b_t\}.\]

**Proof.** Similar to the proof for Lemma 2, we show that all conditions in Definition 15 are met in the process. Condition 1(a-c) are met for the same reason that they met in Lemma 2 that all nodes in \(T^m(\delta)\) correspond to utterances in \(\delta\). Conditions 2(a) through 2(f) in Definition 15 are met by the process given in 3(a) and 3(b) in this lemma. \(\square\)

Since the above inductive process yields a debate tree under the condition that \(T^m(\delta)\) is a tree, we will refer to \(T^m(\delta)\) as a debate in general. Figure 7.1 gives the construction of the debate drawn from the dialogue in our example\(^3\). Note that this is a tree but in general it may not be.

To ensure a debate as a tree, we give the following definition.

**Definition 74.** A debate \(T(\delta)\) is **properly-structured** if and only if it is a tree.

This definition is needed as we need to rule out the case of related utterances with contents of the form \(p \leftarrow q, q \leftarrow r\) and \(p' \leftarrow q\), where the second utterance is top-down related to the first and the third is bottom-up related to the second. Clearly, such situation prevents a properly-structured debate tree being built.

We ensure debate trees drawn from our dialogues are properly-structured with a legal-move function, as follows.

**Definition 75.** A (one-way expansion) legal-move function \(\lambda : D \rightarrow U\) is a **properly-structured legal-move function** if and only if for every dialogue \(\delta \in D\) such that \(T(\delta)\) is properly-structured, then \(T(\delta \circ \lambda(\delta))\) is also properly-structured.

\(^1\)\(^\dagger\) represents expanding a rule within an argument. \(\dagger\) represents the attack relation between arguments. Here, wM, aM, gST, fBT, and T are a shorthand for watchMovie, actionMovie, goodScreenTime, finishByTen, and Terminator, respectively.
We use $\lambda_{ps}$ to denote a generic properly-structured legal-move function.

Given that Lemma 14 has shown that debate trees can be constructed from discovery dialogue that are compatible with $\lambda_{ps}$. We can obtain the following admissibility result of the goal of a discovery dialogue.

**Theorem 10.** Given a successful discovery dialogue $\delta = dD_{a_j}^a(G)$, let $(G\beta, \sigma : P[\cdot])$ be the root node of a properly structured debate tree $T(\delta)$, where $\sigma = \{X/t\}$. Then $G\beta$ is a goal realisation for $G$ with respect to to the ABA framework drawn from $\delta$.

**Proof.** If $G\beta$ is the sentence in the root node of $T(\delta)$ and $\delta$ is successful, then $G\beta$ is admissible with respect to $AF_\beta$, the ABA framework drawn from $\delta$, by Corollary 1.

---

![Figure 7.1: The construction of the debate tree drawn from the discovery dialogue in Example 42.](image-url)

<table>
<thead>
<tr>
<th>utterance1</th>
<th>utterance2</th>
<th>utterance3</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\lambda_{ps}$</td>
<td>$(wM(T), mr : P[4])$</td>
<td>$(wM(T), mr : P[4])$</td>
</tr>
<tr>
<td>$(aM(T), mr : P[2])$</td>
<td>$(fun(T), mr : P[3])$</td>
<td>$(fun(T), mr : P[3])$</td>
</tr>
<tr>
<td>$(gST(T), um : P[4])$</td>
<td>$(gST(T), um : P[4])$</td>
<td>$(gST(T), um : P[4])$</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>utterance4</th>
<th>utterance5</th>
</tr>
</thead>
<tbody>
<tr>
<td>$(aM(T), mr : P[2])$</td>
<td>$(wM(T), mr : P[4])$</td>
</tr>
<tr>
<td>$(fun(T), mr : P[3])$</td>
<td>$(fun(T), mr : P[3])$</td>
</tr>
<tr>
<td>$(gST(T), ma : P[5])$</td>
<td>$(gST(T), ma : P[5])$</td>
</tr>
<tr>
<td>$(aM(T), mr : P[2])$</td>
<td>$(aM(T), mr : P[2])$</td>
</tr>
<tr>
<td>$(late(T), am : O[6])$</td>
<td>$(late(T), am : O[6])$</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>utterance6</th>
<th>utterance7</th>
</tr>
</thead>
<tbody>
<tr>
<td>$(wM(T), mr : P[4])$</td>
<td>$(wM(T), mr : P[4])$</td>
</tr>
<tr>
<td>$(fun(T), mr : P[3])$</td>
<td>$(fun(T), mr : P[3])$</td>
</tr>
<tr>
<td>$(gST(T), ma : P[5])$</td>
<td>$(gST(T), ma : P[5])$</td>
</tr>
<tr>
<td>$(aM(T), mr : P[2])$</td>
<td>$(aM(T), mr : P[2])$</td>
</tr>
<tr>
<td>$(late(T), ma : O[8])$</td>
<td>$(late(T), ma : O[8])$</td>
</tr>
<tr>
<td>$(fBT(T), um : P[9])$</td>
<td>$(fBT(T), um : P[9])$</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>utterance8</th>
<th>utterance9</th>
</tr>
</thead>
<tbody>
<tr>
<td>$(wM(T), mr : P[4])$</td>
<td>$(wM(T), mr : P[4])$</td>
</tr>
<tr>
<td>$(fun(T), mr : P[3])$</td>
<td>$(fun(T), mr : P[3])$</td>
</tr>
<tr>
<td>$(gST(T), ma : P[5])$</td>
<td>$(gST(T), ma : P[5])$</td>
</tr>
<tr>
<td>$(aM(T), mr : P[2])$</td>
<td>$(aM(T), mr : P[2])$</td>
</tr>
<tr>
<td>$(late(T), ma : O[8])$</td>
<td>$(late(T), ma : O[8])$</td>
</tr>
<tr>
<td>$(fBT(T), um : P[9])$</td>
<td>$(fBT(T), um : P[9])$</td>
</tr>
</tbody>
</table>
Hence $G_β$ is a goal realisation with respect to $AF_δ$.

7.4.4 Related Work

McBurney and Parsons [MP01] present a modelling for chance discovery dialogue. The formal system in that work is defined with locutions and rules without linking to any argumentation framework, whereas our work is based on ABA. There is no argumentation semantics used in examining the result of their dialogues, whereas our work makes the connection to the admissibility semantics. Moreover, [MP01] focuses on chance discovery, whereas our work is applicable to any discoveries as long as the desired outcome can be qualified, essentially, by a predicate.

Rybakov [Ryb09] presents a logic modelling of chance discovery. Our work differs from that as is focuses on a dialogue system for discovery whereas his is mainly concerned with constructing a modal/temporal modelling for chance discovery.

Fisher [Fis97] presents a mechanism for concurrent theorem-proving. In his setting, the knowledge base (a set of formulae) is distributed at different objects and each object continuously broadcasts messages about its formulae. Upon receiving messages, an object makes inferences, transforms its formulae and sends out further messages. Even though similarity exists, this work is vastly different from ours as (1) it does not focus on discovering any particular information; (2) it is not concerned with either agents or dialogues; (3) it requires formulae to be represented in Horn Clauses. No formal results have been shown in [Fis97].

7.4.5 Summary

In this section, we introduce a formal modelling for a form of discovery dialogue for two agents, in which the desired outcomes are only partially known when dialogues start. In our setting, the two agents share the same discovery goal but neither of the two agents is capable of discovering a justified goal realisation that fulfils the shared goal. A discovery dialogue is successful if a goal realisation is found through the dialogue.

In our model, the dialogue effectively starts by one agent putting forward a piece of information that might be related to the goal realisation. Through dialogues, more information that is related to the initial utterance is gathered and examined. This process is defined with legal-move functions with the help of constructing a
debate tree. We examine the acceptability of the goal realisation with the admissibility semantics.

7.5 Conclusion

In this chapter, we have presented our formal modelling of conflict resolution and discovery dialogues. Our conflict resolution is realised through a sequence of dialogues. Our discovery dialogue is realised by recognising the “bottom-up” relation between utterances. They both demonstrate how our dialogue model can be applied to various settings.
8 Conclusion

This thesis is about argumentation based multi-agent dialogues. As we state in Chapter 1, this thesis is set to show that:

Argumentation dialogues are a viable means for agents to exchange information and deliberate.

We achieve this goal by firstly constructing a dialogue model (Chapter 3, 4 and 5) and then demonstrate how this model can be instantiated in various types of dialogue (Chapter 6 and 7). The model is sound in that if the two agents follow our protocol to construct a dialogue and our model states that the dialogue is successful, then the claim of the dialogue is acceptable with respect to the information disclosed in the dialogue.

We have given a fair amount of definitions in this thesis, though we have shown a relatively small amount of propositions, lemmas, and theorems. This is not unusual given the theoretical aspects of this thesis and high level of formality we achieve.

We claim that our dialogue model has two main advantages:

1. our dialogues output results that meet certain soundness criteria as our dialogue model is built upon a general-purpose, widely applicable argumentation framework, ABA, with established connection with argumentation semantics;

2. our dialogue model is generic and flexible as it is composed of many loosely coupled components (i.e., legal-move, outcome, and strategy-move functions) that enjoy the freedom in swapping between them to construct dialogues with different goals.

Over the course of the thesis, in addition to constructing our dialogue model, we have made some other contributions as well, such as:

1. formalising a few types of dialogues, namely information-seeking, inquiry, persuasion, conflict-resolution, and discovery (Chapter 6 and 7); and
2. using mechanism design techniques to analyse agent behaviours in persuasion dialogues (Chapter 6).

These results open up many possibilities for future investigation, as discussed below.

In a broader scope, this thesis has made a few contributions in argumentation, multi-agent systems, and the field of artificial intelligence as follows.

- In the field of argumentation, a dialogic model, as the one we presented in this thesis, brings dynamics into argumentation system. Argumentation is meant to be an interactive process where information is exchanged. However, much research on argumentation treats it as a static process in which the main concern is to reason “correctly” from a fixed set of postulates. A dialogic approach brings interaction and dynamics to the reasoning process and provides a more applicable modelling of argumentation.

- In the field of multi-agent system, as we stated in Chapter 1, agents need to communicate and interact with each other. This thesis contributes to formalising high level agents communication. As presented throughout this thesis and discussed earlier, our approach is sound and flexible.

- In the field of artificial intelligence in general, where, as Russell and Norvig [RN03] point out, AI is about understanding as well as building intelligent entities, our work should be understood in the context of enabling more capable software or hardware agents. Moreover, our work can be viewed as an attempt to bring formalism to some areas, e.g., dialogue taxonomy, which are predominantly presented informally. Such effort is worthwhile as it enables us in understanding these areas more concretely and systematically.

This thesis has touched upon a variety of topics. Some of them have been investigated more thoroughly than others. Nevertheless, a large amount of new work can continue the research presented in this thesis. We divide our discussion on future work into two groups: refinement and extensions.

We first talk about refinement. For conflict resolution and discovery dialogues, we have only shown admissibility results. Moreover, we have forced conflict resolution dialogues to be focused and discovery dialogues to be compatible with a one-way expansion legal-move function. Both constraints can be lifted by using results based on debate forests rather than debate trees. In this way, more generic dialogues can be modelled.
The work on persuasion dialogues with mechanism design can also be enhanced. In that work, we have defined the “damage” of a (potentially lying) utterance and the possibility of an utterance being considered as a lie to be some constants given to agents. Clearly, both values are not easily quantifiable. Hence, it would be interesting to see algorithms that estimate these values.

As for extensions, it would be interesting to see our dialogue model being used in other types of dialogues, namely, negotiation and deliberation. Both types of dialogues may require a new modelling of agent utility and/or action utility.

Secondly, it would be useful to experiment our dialogue model with other argumentation semantics, e.g.,prefered. Since we obtain soundness results by mapping debate trees to dispute trees, it is easy to extend our results to any new semantics if it is computed with some dispute trees.

Thirdly, as discussed in [WK95], many real dialogues are sequences of sub-dialogues of different types. Our current dialogue model supports single dialogues. It would be interesting to see how our model can be extended to support generic dialogue sequences.

Lastly, if we consider that the study of agent dialogues should contribute to agent understanding of human dialogues, then it would be interesting to study how human dialogues represented in natural language can be translated into some machine readable formal models. Moreover, if such translation is robust, then possibly many interesting inferencing can be performed over the constructed model and then useful feedbacks can be generated and sent to human users in natural language. Such achievement would advance many studies in dialogue systems and artificial intelligence as a whole.

To conclude, by designing a generic argumentation based dialogue model, proving its soundness, applying it to several types of dialogues, we have shown that: argumentation dialogues is a viable means for agents to exchange information and deliberate, as we originally aimed.
Bibliography


[TGK+08] Francesca Toni, Mary Grammatikou, Stella Kafetzoglou, Leonidas Lymberopoulos, Symeon Papavassileiou, Dorian Gaertner, Maxime Morge, Stefano Bromuri, Jarred McGinnis, Kostas Stathis, Vasa


Index

ABA argument, 29
attacked in debate tree, 81

Abstract Argumentation, 22
labelling, 26
semantics, 24
abstract dispute tree, 31
Assumption-based Argumentation, 28
semantics, 30, 31

conflict, 128
debate forest, 82
filtered, 87
patient, 87
debate tree, 60
argument drawn, 62
culprits, 68
defence set, 68
expanded, 75
filtered, 66
focused, 69
patient, 64
properly-structured, 142
dialogue, 42
coherent, 54
conflict resolution, 131
discovery, 139
sequence, 131
successful

a-successful, 50
g-successful, 50
i-successful, 50
taxonomy, 33
framework drawn, 46
goal, 125
realisation, 128
joint framework, 95
legal-move function, 48
admissible, 79, 80
filtered, 66, 87
flat legal-move, 48
focused, 69
grounded, 79, 80
one-way expansion, 78
patient, 65, 87
properly-structured, 142
related, 71
top-down related, 54
mechanism design, 112
action space, 116
dialogue strategy, 116
outcome, 116
outcome function, 117
payment, 116