A Generative Framework for Argumentation-Based Inquiry Dialogues

Elizabeth Black

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
of the
University of London.

Advanced Computation Laboratory
Cancer Research UK
and
Department of Computer Science
University College London

2006

I, Elizabeth Black, confirm that the work presented in this thesis is my own. Where information has been derived from other sources, I confirm that this has been indicated in the thesis.

Abstract

My PhD focusses on argumentation-based communication between agents. I take, as a starting point, the argumentation system proposed by García and Simari [22], which allows a single agent to reason about its beliefs. I define a novel dialogue system that allows two agents to use García and Simari's system to carry out *inter*-agent argumentation. I define two specific protocols for two different types of inquiry dialogue that I define: argument inquiry and warrant inquiry. Argument inquiry dialogues are often embedded within warrant inquiry dialogues.

Other existing inquiry dialogue systems only model dialogues, meaning that they describe what a legal inquiry dialogue is, but they do not provide the means to actually generate such a dialogue. Such systems provide a protocol, which dictates what the possible legal next moves are at each point in a dialogue *but not which of these moves to make*. I present a system that not only includes two dialogue-game style protocols, one for the argument inquiry dialogue and one for the warrant inquiry dialogue, but also includes an intelligent strategy, for an agent to use with these protocols, that selects exactly one of the legal moves to make.

As my system is generative, it allows me to investigate the precise behaviour of the dialogues it produces. I propose a benchmark against which I compare my dialogues, and use this to define soundness and completeness properties for argument inquiry and warrant inquiry dialogues. I show that these properties hold for all dialogues produced by my system. Finally, I go on to define another intelligent strategy for use with warrant inquiry dialogues. I show that this also leads to sound and complete dialogues but, in many situations, reduces the redundancy seen in the dialectical tree produced during the dialogue.

Acknowledgements

I'd like to thank both my supervisors Tony Hunter and John Fox for their continued support and encouragement throughout the past four years. Thanks to Tony for his invaluable help and advice regarding both the technical aspects of this thesis and academic life in general. Thanks to John for providing such a rich and interesting domain in which to base my work and for the stimulating discussions that have helped drive this thesis forward. Thanks to everyone in my lab for helping to create such a constructive and enjoyable working environment—I'd particularly like to thank Sanjay Modgil and Matt Williams for their helpful feedback and comments on my work, and David Glasspool for his support throughout my MSci project and at the early stages of my PhD. Thanks to Alyssa Alabassi for inviting me to the breast clinic with her and for sharing her knowledge of the breast cancer domain with me. Thanks to Peter McBurney for his reassurance and feedback on my ideas at the early stages of my PhD. Thanks to John Dowell and Robin Hirsch for providing me with the opportunity to partake in a mock viva and for taking the time to read my thesis in progress. Thanks to Paul Bates for his input as part of my thesis committee. Thanks to Ulises Cortés, Javier Vázquez-Salceda and the other members of the Knowledge Engineering and Machine Learning Group at UPC for looking after me so well when I visited Barcelona, and thanks to the AgentCities project for partly funding these visits. Thanks to Michael Wooldridge and Francesca Toni for taking the time to read this thesis and for examining me. Thanks to my family and friends for their support and their pride in me. Finally, thanks to Cancer Research UK and all of its supporters for funding this PhD.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

In this chapter I start by giving an overview of the work presented here and describe the structure of this thesis. I then provide the context for this work, summarise the research questions I address, and end by highlighting the contributions made by this work.

1.1 Overview

The work described here is intended to inform the CREDO project[1]—an ongoing project being undertaken at the Advanced Computation Laboratory[2] at Cancer Research UK. The aim of CREDO is to develop and test a general approach to building and verifying clinical systems for supporting multidisciplinary patient care. In such systems, different parts of the care process are the responsibility of different individuals (doctors, nurses, etc.), who possess different skills and responsibilities, and who are frequently in different places, but who must nevertheless work together as a team. These properties make it natural to model such an organisation as a multi-agent system, where the different agents involved are working in parallel to meet their own objectives, but are also working towards the same overriding goal of maximising the quality of patient care [63].

The goal of the project is to develop a software agent architecture in which agents can support the individuals in their work, and support communication and coordination between them in a way that produces measurable benefits in speed and effectiveness of care, and in improvements to patient safety. It is expected that such a system would bring serious improvements to patient care [31, 27], the standard of which often differs immensely depending on which medical centre is visited. Indeed, various studies have suggested that a patient with breast cancer who visits a specialist centre may be up to seven times more likely to receive successful treatment than a patient who visits a general medical facility [20, page 5]. The CREDO system is intended to ensure that all medical professionals are providing the best treatment, thus standardising patient care.

The dialogue system I propose in this thesis focusses on the communication between exactly two agents in the CREDO system. Agent communication is a key issue in multi-agent systems, as it allows agents to coordinate their actions and share information in order to jointly achieve their goals. The work reported here is theoretical in nature and is concerned with inquiry dialogues in particular. Walton

[1] http://www.acl.icnet.uk/lab/credo.html
[2] http://www.acl.icnet.uk/lab/index.html

and Krabbe [65, page 66] define an inquiry dialogue as arising from an initial situation of "general ignorance", and as having the main goal to achieve the "growth of knowledge and agreement". They say that each individual participating in an inquiry dialogue has the goal to "find a 'proof' or destroy one". No formal definition of inquiry dialogues is given, leaving this classification somewhat open to interpretation. To address this, I have defined two different types of inquiry dialogue, each of which I believe fits this general definition: *argument inquiry* and *warrant inquiry*. In a warrant inquiry dialogue, the 'proof' takes the form of a dialectical tree (essentially a tree with an argument at each node, whose arcs represent the counter-argument relation, and that has at its root an argument whose conclusion is the topic of the dialogue). In an argument inquiry dialogue, the 'proof' takes the form of an argument for the topic of the dialogue. Argument inquiry dialogues are commonly embedded in warrant inquiry dialogues.

A key contribution of this work is that I not only provide a protocol for modelling inquiry dialogues, but I also provide two specific strategies to be followed, making this system sufficient to also *generate* inquiry dialogues. Other works have also addressed the automation of dialogues (e.g. [1, 5, 29, 51]), however, none have provided a specific mechanism that, at each point in time in an inquiry dialogue, selects exactly one move to make. This makes it hard to analyse the precise behaviour of inquiry dialogues, and it misses the opportunity to make intelligent selection of the next move. As far as I am aware, mine is the only example of a system that incorporates an intelligent strategy capable of generating inquiry dialogues.

My dialogues are defined in terms of a set of moves that can be made within the dialogue, a protocol that returns a set of moves that may legally be made at any point in the dialogue, and a strategy function that returns exactly one of the legal moves at any point in the dialogue, which is the move that is subsequently made. I propose a benchmark against which to compare my dialogues and show that the first strategy I define leads to sound and complete inquiry dialogues in relation to this benchmark. I then consider types of redundancy that appear in the dialectical tree constructed during a warrant inquiry dialogue, and define another strategy that I show reduces such redundancy and yet still leads to sound and complete warrant inquiry dialogues.

This thesis is structured as follows.

- **Chapter 1**—this chapter. Provides the context for this work and explains the research questions addressed.

- **Chapter 2**—is a review of the relevant literature.

- **Chapter 3**—describes the argumentation system used in this framework, based on that of García and Simari [22].

- **Chapter 4**—describes the general dialogue system, gives the protocols for the warrant inquiry and argument inquiry dialogues, and gives the first strategy for investigation, called the exhaustive strategy.

- **Chapter 5**—is an analysis of the general framework and of dialogues produced by the exhaustive strategy.
- **Chapter 6**—gives details of the strategy intended to lead to reduced redundancy in the dialectical tree produced by the warrant inquiry dialogue, called the pruned tree strategy.
- **Chapter 7**—is an analysis of dialogues produced by the pruned tree strategy.
- **Chapter 8**—summarises the contributions made in this thesis, discusses its shortcomings and considers future work.

1.2 Context

In this section I discuss the context of this thesis. This is split into two sections. Firstly, Section 1.2.1 discusses a problem that has arisen in the medical domain, called the medical knowledge crisis [20, pages 3–12]. A promising approach to solving this problem is to provide multi-agent technology that will support doctors and other health care professionals, but to do so the agents in the system must be able to communicate effectively and reliably, which is where the focus of my thesis lies. In Section 1.2.2 I discuss the issues concerning agent communication.

1.2.1 Medical knowledge crisis and a possible solution

The medical profession is facing a knowledge crisis of increasing severity [20, pages 3–12]. Medical knowledge is expanding at an unprecedented rate, while the resources available to apply it remain fixed. Similarly, medical technologies and theories have progressed substantially over the last few decades, whilst the practices and skills within the medical profession have remained largely unchanged. The disparity between human capabilities and the results that it should, given our knowledge, be possible to achieve is exacerbated by the lack of financial resources available to the medical profession. This has led to the undesirable situation in which patients receive varying levels of care, with the likelihood of recovery dependent on which medical centre the patient visits. For example, breast cancer is one of the more common forms of cancer, with about one in eleven western women contracting it at some point in their lives. As such, there is an international consensus on the best way to go about treating the disease. Despite this, there is much variation in the likelihood of curing the disease between different treatment centres. According to various studies, a patient may be up to seven times more likely to receive a cure at one of the best specialist centres than at a general medical facility [20, page 5].

Why is it the case then that there are variations in levels of care of breast cancer when there is an established ‘best practice’ for treating the disease? There may be differences in resources available in treatment centres, and it is also likely that there will be different levels of experience and expertise in different treatment centres. A doctor faced with a form of cancer that they have not seen before will not have as much relevant knowledge to draw on as a doctor who specialises in that form of cancer. It must also be remembered that doctors are only human and may make the wrong decision even if they have all the relevant information available to them.

Currently in the United Kingdom, a general practitioner (GP) is responsible for providing primary care for a couple of thousand people. This means they need good knowledge of hundreds of diseases and their symptoms, and also of a drug database of thousands of items. As well as this, they are also required to act as gatekeepers to hundreds of specialist services, which requires them to have enough knowledge to know when to refer a patient and who to refer the patient to. They must keep abreast of medical developments in order to continue to provide a high standard of care, and any mistakes that they make could have disastrous consequences for their patients.

Doctors are expected by many to be infallible and are held responsible for any mistakes that they may make. However, levels of care provided by doctors are not going to improve simply by demanding the impossible from them. Doctors cannot be expected to hold such a large body of knowledge in their head, nor can they be expected to always make the correct decisions. It is widely believed that the only way to improve this situation is to introduce new technologies that will aid clinicians in the organisation of their knowledge, and in their decision-making and the management of their work [27]. In the UK, a national secure network, called NHS net, is being developed to link every organisation and healthcare professional in the NHS. This should cause a dramatic improvement to communications within the health service. It will allow new knowledge to be effectively distributed throughout the medical community. With the coming use of electronic medical records, the NHS net will also allow much better access to patient data. While this sort of development will help to ensure that doctors have all the relevant knowledge available to them, it does not guarantee that they will make the best decision based on this knowledge.

It is clear that patients are not uniformly receiving the best level of care that medical knowledge makes available. This has led to the interest in “evidence-based medicine” [45], in which any decisions made are based on all the available scientific information, and not just on a clinician’s individual experience and opinion. In order to make this possible, ways must be found of aiding the clinicians in their knowledge management, so that they are able to recall all the relevant knowledge when required and make the best possible decision based on all this information.

One set of tools currently widely available to aid doctors are clinical guidelines. These usually come in the form of official statements from health organisations and agencies on how best to care for medical conditions or to perform clinical procedures [16] and are intended to guide the doctor in his treatment of the patient. Clinical guidelines exist for many common diseases. For example, some of the clinical guidelines produced by the National Institute for Health and Clinical Excellence (NICE[3]) are for antenatal care, depression, epilepsy, and hypertension.

As well as aiding doctors in their decision-making, such as what course of treatment to embark on, guidelines also give advice on task management. They do not, however, commonly aid with disease diagnosis. There is a clear need for such guidelines, as a recent estimate from the US Institute of Medicine suggests that there may be as many as 98,000 unnecessary deaths each year that are the result of avoidable clinical error [31]. However, these paper-based guidelines are often not used in practice.

[3] http://www.nice.org.uk

There are several reasons for this: they can be difficult to follow and may consist of many pages of text and diagrams; they are not able to give patient specific advice; there are so many of them that doctors often have problems retrieving the relevant guideline at the appropriate time; they are often viewed as intrusive if used during a doctor-patient consultation.

An important aim of these guidelines is to try and achieve a consistent level of care across the country. However, to have any chance of being effective, a way of implementing these guidelines must be found that ensures doctors will follow them. Passive dissemination of guidelines, by simply publishing them in journals and distributing them to the medical community, is not enough. A review of nineteen studies of passive dissemination of guidelines concluded that it is unlikely that this sort of dissemination alone will lead to any change in the behaviour of healthcare professionals [37]. Whilst considerable effort has gone into developing clinical guidelines, there has been less of an attempt made to ensure that these guidelines are used effectively on a day-to-day basis.

According to [19], there are three factors that play an important part in shaping peoples behaviour.

- Perceived benefits weighed against perceived costs; for example, improved patient outcome versus costs associated with the change.
- Perceptions about the attitudes of "respected others" to the behaviour.
- Self-efficacy, or the belief in one's ability to perform a particular behaviour.

So, to ensure that doctors modify their behaviour to meet a clinical guideline, it is important to minimise the cost to the doctor that this change in behaviour entails. It is also vital to maximise doctors' confidence in their ability to follow the guideline, and to make the guideline as easy to follow as possible. An alternative to publishing paper guidelines is to represent the medical knowledge contained in these guidelines formally, in such a way that the knowledge can be applied by a computer to support clinicians in their work [8]. Experience suggests that embedding clinical guidelines in a computer decision support system, rather than presenting them as static, paper-based guidelines, improves the clinician acceptance of the guideline, notably changes behaviour and practice, and significantly improves the quality of patient care [53, 10].

An executable computer language for representing clinical guidelines has been developed at the Advanced Computation Laboratory. This language is called PRO*forma* [21, 20, 8]. PRO*forma* makes it possible to capture a range of clinical tasks, such as reminders for the collection of patient data, decision-making and the scheduling of actions. Individual tasks may be grouped together to form plans that can then represent complex clinical procedures. Once such a procedure has been represented in PRO*forma* it can be used in the following ways [21].

- It can be reviewed by specialist clinicians and scientists to ensure that it captures best clinical practice.
- It can be electronically disseminated as a source of up-to-date reference material.
- It can be enacted by a computer in order to assist medical staff to follow the recommended process.

- It can also be more easily retrieved at the appropriate time, as the standardised form makes it easier to index the guidelines.

The intention is that this new technology will allow clinicians to use the established clinical guidelines much more effectively.

CREDO is an ongoing project at the Advanced Computation Laboratory that uses PRO*forma* to represent the knowledge needed to support the care of patients in the breast cancer domain. This domain is typical of the more general medical domain in that it consists of many different services, located in many different locations, that all have to interact in many different ways in order to ensure the best possible care for the patients. For example, there is the screening service—often provided by a mobile unit that is in a different location each day, the radiotherapy service—located in a department at a hospital, and the genetics and risk assessment service—often a unit in a different hospital. When the specialists are deciding on a diagnosis or the best treatment plan for a patient they may need information and expertise from these, and other, services.

The need for distributed decision-making means that multi-agent systems are a reasonable candidate for modelling medical organisations such as the breast cancer domain. The AgentCities Working Group on Health Care have identified some other features that are common in the health care domain and suggest the application of multi-agent technology [44]: data is distributed; the software solution is complex; there is a lack of centralised control; there is a need to maintain independence between the health care entities; communication and coordination are essential; information and advice need to be obtained proactively.

Medical services in general represent an interesting and challenging domain in which to investigate multi-agent functionality, because of the requirements for coordinating heterogeneous networks of services, which may be time-critical and may involve significant levels of uncertainty about the situations that can arise, the consequences of actions etc. In such systems different parts of the care process are the responsibility of different individuals (doctors, nurses etc.), who possess different skills and responsibilities, and who are frequently in different places but who must nevertheless work together as a team. The CREDO goal is to develop a multi-agent system in which agents can support the individuals in their work, and support communication and coordination between them in a way that produces measurable benefits in speed, effectiveness and safety of patient care processes.

It is the communication between the different CREDO agents that I am particularly interested in. As an example of the kind of communication I wish to support, consider the following scenario—called the *referral agent scenario*. When a patient displaying breast symptoms visits a GP, the GP must decide whether the symptoms warrant a normal referral to the breast cancer clinic, an urgent (within two weeks) referral to the breast cancer clinic, or if the patient can be managed by the GP. There is an official clinical guideline that states how this decision should be made and which doctors should adhere to, but, in reality, these are very seldom referred to by GPs. As a result, mistakes in referral are often made, with the most common result being that people are referred for urgent appointments when actually they should have been normally referred or treated by the GP. This puts an unnecessary strain on the breast cancer clinics, and means that people who really do need urgent referrals may have to wait longer.

Suppose that each GP has their own agent that keeps track of all the GP's patients' information. There is also a central referral agent, which is programmed with information from the guidelines as to when patients should be referred and what level of referral they demand. A patient comes to see the GP. He types her symptom information into his computer. The GP's agent immediately sees that these are breast symptoms and knows that a referral decision must be made. The GP agent enters into a communication with the referral agent to see what the official guideline recommendation for this specific patient is. The two agents pool the relevant parts of their patient specific and general guideline knowledge in order to come up with this recommendation. The GP agent immediately reports the recommendation and the reasons for it to the GP, who is still with the patient and can now act on this recommendation.

1.2.2 Agent communication

Wooldridge and Jennings describe an agent as a computer system that displays the following properties (these bullet points are quoted verbatim) [66, pages 4–5].

- 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.

A multi-agent system, then, is a network of two or more of these agents interacting in some way to achieve both their own internal goals and the shared, overarching goals of the network. Multi-agent technology has earned much interest in recent years and is currently being applied in many different domains. They are commonly applied to problems that are solved more efficiently by division of the problem into smaller parts and distribution of these smaller problems among the agents. These problems are often distributed in the real world, such as with the problem of caring for a breast cancer patient, making it natural to apply a multi-agent system.

Agent communication is a key issue in multi-agent systems. It allows agents to coordinate their actions and share information in order to cooperate to jointly achieve their goals. Labrou and Finin go so far as to say that [35]

> Agent-to-agent communication is key to realizing the potential of the agent paradigm, just as the development of human language was key to the development of human intelligence and societies.

In order to communicate, agents firstly need a shared, unambiguously specified language. It is computationally infeasible to provide agents with a language as expressive as natural language. Instead, the convention in agent communication languages (ACLs) is to provide a small set of language primitives, which classify the intention of the message, called *performatives*. The content of the message is then treated separately, and this is the proposition to which the performative applies.

This approach stems from speech act theory, which was developed by Austin [4] and Searle [62]. They focussed on the intention and effect of language, rather than what is actually said. For example, if someone said to you "Can you pass the salt?", it is likely that they do not want to know whether you are physically able to pass the salt, but rather that they desire you to pass it to them. Messages sent by agents are considered as intentional actions that may have consequences on the environment [38].

There have been two main attempts at developing ACLs: KQML and FIPA-ACL. The first ACL to be developed was KQML [17]. This came about as a result of the Knowledge Sharing Effort—a initiative of the Defense Advanced Research Projects Agency of the US Department of Defense. A KQML message can be conceptualised into three distinct layers [36]. The *communication* layer specifies the sender, the receiver, and the unique identifier of the message. The *message* layer specifies the intentional force (or performative) of the message, as well as the knowledge representation language being used to specify the content, and any ontologies being used. Finally, the *content* layer specifies the content of the message, in whatever knowledge representation language has been specified in the message layer.

I will now give an example of a KQML message where AGENT1 tells AGENT2 that it is forecast to snow tomorrow. The language used to represent the content of the message is Prolog and a weather ontology is being used.

```
(tell
    :sender      AGENT1
    :receiver    AGENT2
    :language    Prolog
    :ontology    Weather
    :content     forecast(snow, tomorrow)
```

The syntax to KQML is relatively simple but it was originally defined without any precise formal semantics, attracting some criticism [11]. Finin and Labrou went on to provide the semantics of KQML in terms of preconditions, postconditions and completion conditions of each performative [32, 33, 34].

A more recent attempt to provide a standard ACL is FIPA-ACL. This is a result of the Foundation for Intelligent Physical Agents (FIPA) initiative[4]. FIPA-ACL has a precise semantics based on a formal language called SL. The semantics of each performative is given in terms of feasibility conditions and rational effects [38]. Feasibility conditions are the necessary conditions that the sender of the message must achieve. Rational effects describe the effects that an agent can expect to occur as a result of the message. For example, if AGENT1 wishes to **INFORM** AGENT2 of proposition P, then it must satisfy the feasibility conditions that AGENT1 believes the proposition P, AGENT1 does not believe that

[4]http://www.fipa.org/

AGENT2 already has some belief about the proposition P and AGENT1 intends that AGENT2 comes to believe that the proposition P is true. The rational effect of sending this inform message is that AGENT2 comes to believe proposition P [18].

Agents use communication to further their intentions. Due to the complexity of the agents and the situations they find themselves in, sequences of single, unrelated messages are normally not sufficient for this task; instead the agents must engage in coherent dialogues. At any point in the dialogue, an agent must select the appropriate message that has the greatest chance of affecting the world in such a way as to help them achieve their goals. One approach to allowing agents to have extended dialogues is to simply provide the agents with an ACL and allow the dialogues to emerge from the ACL semantics. In order to make the best attempt at choosing the optimal message to send, an agent must try to infer what the other agents' intentions are. However, an agent cannot reliably infer another agent's intentions based simply on the messages that have been produced, as these messages could have been intended to achieve any one of several different goals. Greaves *et al.* [23, page 119] call this the *Basic Problem*:

> Modern ACLs, especially those based on logic, are frequently powerful enough to encompass several different semantically coherent ways to achieve the same communicative goal, and inversely, also powerful enough to achieve several different communicative goals with the same ACL message.

If nothing constrains the use of the language apart from its semantics then it is very hard for agents to compute what message they should send in order to optimise their position, as they cannot be sure what the other agents' mental states are.

Conversation policies have been used to reduce the complexity of deciding what message to send. They limit the amount of messages that an agent has to consider by constraining the sequences of semantically coherent messages that lead to a goal [23]. As Greaves *et al.* say [23, page 23]

> [...] conversation policies limit the possible ACL productions that an agent can employ in response to another agent, and they limit the possible goals that an agent might have when using a particular ACL expression.

So, conversation policies make the decision process used by the agents to select a message tractable. However, they have been criticised as being often only semi-formally stated [9], and very inflexible [9, 12].

An alternative approach is the use of dialogue games to structure the dialogue. This gives us something between conversation policies—which appear to be too restrictive, and free use of the agent communication language—which is computationally intractable, and it is this approach which I intend to use to structure the dialogues within my system.

Dialogue games stem from argumentation theory, and this is a particular draw for the medical domain. Argumentation-based communication languages allow a rich flow of information between agents. They allow agents to give reasons for their position and to alter their position in light of new information. In the medical domain, it is vital that agents are able to back up their claims, giving the reasons why they have arrived a a particular diagnosis or made a particular decision. It is also true that information is changing all the time, whether this is information about the state of the environment such as a patient's blood pressure, or information resulting from research into the general behaviour of a disease. It is important that agents in the medical domain can respond to new information, altering their position accordingly.

Dialogue games are played between two or more players, although for the sake of simplicity this work will deal only with games between exactly two players, called the *participants*. For a given dialogue game there are generally

- a set of legal moves that the players can make;
- one commitment store for each of the participants in the dialogue, which maintains the set of propositions that the participant is currently committed to;
- a set of rules governing the use of these moves;
- a set of rules defining the effect of a move on the commitment stores.

Dialogue games typically formalise one or more of the Walton and Krabbe typology of human dialogues [65]: information-seeking, inquiry, persuasion, negotiation, deliberation and eristic. These dialogue types are classified according to three characteristics

- the initial situation—particularly in terms of what conflicts of knowledge exist;
- the main goal of the dialogue—to which all participating agents subscribe;
- the participants' individual aims.

Multi-agent research has so far chosen to ignore the eristic type of dialogue, as this is intended to "serve primarily as a substitute for [physical] fighting" [65, page 76] and so is not expected to be a useful type of dialogue for agents to take part in. The classifications for the other five dialogue types are given in Table 1.1. One attraction of dialogue games is that it is possible to embed games within games, allowing complex conversations made up of nested dialogues of more than one type (e.g. [59, 41]).

Dialogue game protocols have been defined for all the five main dialogue types, for example: information-seeking [48, 26]; inquiry [40, 39]; persuasion [3, 12]; negotiation [2, 26, 42, 61]; deliberation [25]. However, almost all of these only provide the protocols intended to model the dialogues, they are not generative systems. This brings us to an area of the literature which appears to be particularly lacking—how an agent should navigate through the legal dialogue structure. The agent must make these choices in such a way that it maximises its chance of achieving its goals. This is a concern that is usually left up to the agent developers, with no theoretical guidance.

Although most of the existing work on dialogue games is concerned with dialogue modelling and not dialogue generation, there are a few exceptions to this. Rahwan, McBurney and Sonenberg [58] give an account of the different factors which must be considered when designing a dialogue strategy. Work done by Parsons, Wooldridge and Amgoud [46, 47] explores the effect of different agent attitudes, which

Type	*Initial Situation*	*Main Goal*	*Participant's Aims*
Persuasion	Conflicting points of view	Resolution of such conflicts	Persuade the other
Negotiation	Conflict of interests and need for cooperation	Making a deal	Get the best out of it for oneself
Inquiry	General ignorance	Growth of knowledge and agreement	Find a 'proof' or destroy one
Deliberation	Need for action	Reach a decision	Influence outcome
Information-seeking	Personal ignorance	Spreading knowledge and revealing positions	Gain, pass on, show, or hide personal knowledge

Table 1.1: Types of dialogue [65, page 66].

reduce the set of legal moves from which an agent must choose a move, but do not select exactly one of the legal moves to make. Pasquier's cognitive coherence theory [50] attempts to address the pragmatic issue of dialogue generation, and I will discuss this further in the next chapter.

In recent work, Amgoud and Hameurlain [1] propose a decision model that selects the best move to make at a point in a dialogue and a formalism for representing the arguments on which to base this decision, but they do not provide a specific strategy for use with inquiry dialogues. Another group that have proposed a framework for defining strategies is Kakas *et al.* [28, 29]. However, they also do not provide any specific strategies. As far as I am aware, the work presented here is the only work that proposes a specific strategy that allows the generation of inquiry dialogues.

As agent communication is such a young field, and as a consequence of the fact that there are very few proposals of specific strategies, we still don't know much about the formal properties of the dialogues produced by the various systems proposed. There are some results on termination. Sadri *et al.* [61] show that a dialogue under their protocol always terminates in a finite number of steps, and Parsons *et al.* [46, 47] consider the termination properties of the protocols given in [3, 2]. There are also some complexity results: Parsons *et al.* [46, 47], and Dunne *et al.* [15, 14] consider questions such as "How many algorithm steps are required, for the most efficient algorithm, for a participant to decide what to utter in a dialogue under a given protocol?" and "How many dialogue utterances are required for normal termination of a dialogue under the protocol?".

If we are to use dialogue game protocols in the safety-critical medical domain, then we must certainly understand the behaviour of the dialogues that they produce. I am particularly interested in the outcome of my dialogues, and propose a benchmark which I use to define soundness and completeness properties. As far as I am aware, the only other similar work that considers soundness and completeness properties is that of Sadri *at al.* [60], who define different agent programs for negotiation. If such an agent program is both exhaustive and deterministic, then exactly one move is suggested by the program at a timepoint, making such a program generative and allowing consideration of soundness and completeness properties. They discuss the soundness and completeness of some proposed agent programs. Other work misses the chance to better understand the dialogue behaviour by considering such properties, as they do not provide specific strategies.

1.3 Research questions

I will now summarise the three main research questions that I address in this thesis, which were derived from the requirements of the medical domain and the CREDO project in particular. In the literature review in Chapter 2, I consider existing dialogue systems and show that none of them address each of my research questions.

1. *Can I define a system that allows automatic generation of inquiry dialogues between two agents?* I want my system to be of practical use within the medical domain, and this means that agents must be able to actually generate dialogues (i.e. the system must provide more than just a protocol for modelling legal inquiry dialogues, it must also provide a specific strategy for selecting exactly one of the legal moves). I am focussing on inquiry dialogues as these are particularly useful in a cooperative medical domain, where different agents may often need to share knowledge in order to come up with new information (for example, the referral agent scenario described in Section 1.2.1, page 15).

2. *Can I propose a benchmark system against which to compare my system, and then show that the dialogues produced by my system are sound and complete in relation to the conclusions drawn by the benchmark system?* As my dialogue system is intended for use in the safety-critical medical domain, it is essential that dialogues it produces arrive at the appropriate outcome. This guarantee of a certain outcome given a certain situation is lacking from most other comparable proposals.

3. *Can I define a second specific strategy that generates dialogues that produce a smaller output than those generated by the first strategy, and yet are still sound and complete?* This research question is not driven particularly from the medical domain, but from a general desire to improve the system.

1.4 Novel contributions made by this thesis

The contributions made by this thesis address the three main research questions summarised in Section 1.3.

1. *Can I define a system that includes a specific, intelligent strategy that allows automatic generation of inquiry dialogues between two agents?*
   In Chapter 4, I present a dialogue system along with a protocol for the argument inquiry dialogue and a protocol for the warrant inquiry dialogue. These protocols (Definitions 4.3.2 and 4.4.3) return the set of legal moves at a point in a dialogue. I go on to provide a specific strategy for use

by an agent with either of these protocols (Definition 4.5.4) that returns exactly one of the legal moves at a point in a dialogue, which is the move that the agent makes (and so allows automatic dialogue generation). The strategy I provide is intelligent as I will show that it leads to sound and complete dialogues.

2. *Can I propose a benchmark against which to compare my system, and then show that the dialogues produced by my system are sound and complete in relation to the conclusions drawn by the benchmark system?*

   In Chapter 5, I propose a benchmark and use this to define what it means for an argument inquiry dialogue to be sound and complete (Definitions 5.5.1 and 5.5.2) and what it means for a warrant inquiry dialogue to be sound and complete (Definitions 5.7.1 and 5.7.2). I go on to show that dialogues generated by my system are both sound and complete (Theorems 5.5.1, 5.5.2, 5.7.1 and 5.7.2).

3. *Can I define another specific strategy that is, in some sense, more efficient than the first but still leads to sound and complete dialogues?*

   In Chapter 6, I consider types of redundancy that appear in dialectical trees. I then define another specific strategy (Definition 6.0.1) that reduces the amount of occurrences of these types of redundancy that appear in the dialectical tree constructed during a warrant inquiry dialogue. In Chapter 7, I show that dialogues generated by this strategy are both sound and complete (Theorems 7.6.1 and 7.6.2). I then show that the dialectical tree produced by this second strategy is never larger, and is sometimes smaller, than that produced by agents following the first strategy that I defined in Chapter 4 (Theorems 7.8.1 and 7.8.2).

1.5 Summary

In this chapter, I have introduced my work and the context for it. I have given an overview of the structure of this thesis, and summarised the three main research questions that it addresses. Finally, I have highlighted the contributions made here. In the next chapter I will give a review of the existing systems that are relevant to mine.

Chapter 2

State of the art

In this chapter I review the main proposals that are comparable to mine. I start by identifying four desirable properties that hold for my system, and go on to show that no existing system has all four of these properties.

2.1 Desired features for my dialogue system

When I first started to investigate this area of research what struck me most was that none of the existing theories provided everything necessary for an agent to automatically generate dialogues. If such a theory is to be useful in a project such as CREDO, then it needs to be of practical use, and this motivated my work. I have decided on four features that I believe are necessary for a dialogue system to be useful as part of a multi-agent system in the medical domain.

- **Provides inquiry protocol.** I chose to focus my attentions on the inquiry dialogue as it is a cooperative dialogue that embodies one of the more general goals of the medical domain—making a justified claim, such as providing reasons for why a patient should be urgently referred to a specialist. It is also one of the dialogue types to receive the least attention in the literature so far.

- **Generative.** I am interested in defining a practical system that will allow two agents to automatically generate a dialogue. For a dialogue system to be generative it must specify exactly one move to be made at any point in the dialogue.

- **Formally specified.** I want my system to be of use in the real world. Specifying such a system formally should remove any ambiguity about how the protocol should be followed and will facilitate the investigation of the properties of the system.

- **Provides results about dialogue outcome.** As I am concerned with designing a theory that may be used in the medical domain, it is important that the behaviour of the system is well-understood and suitable to the domain. This means that it needs to be certain that the system is going to behave in the intended manner. In particular, I am interested in results about the outcome of the dialogue and need to know that a dialogue system is always going to produce the desired outcome in any given situation.

	Inquiry protocol	Generative	Formal	Outcome results
The dialogue system proposed in this thesis	YES	YES	YES	YES
Amgoud *et al.* [3, 2, 46, 47, 48]	YES	NO	YES	YES
McBurney and Parsons [39, 40]	YES	NO	NO	NO
Bench-Capon [5]	NO	YES	NO	NO
Sadri *et al.* [60, 61]	NO	NO	YES	YES
Kakas *et al.* [28, 29, 30]	NO	NO	YES	NO
Amgoud and Hameurlain [1]	NO	NO	YES	NO
Pasquier and Chaib-draa [50, 49, 51, 52]	NO	YES	NO	NO
Prakken [54, 55, 56]	NO	NO	YES	YES

Table 2.1: Comparison of different features of dialogue systems.

As I am concerned with the medical domain, it is desirable that my dialogues are predetermined. That is to say, I do not wish it to be possible for an agent to use an intelligent strategy to influence the outcome of the dialogue, rather, I wish the dialogues to always lead to the 'ideal' outcome. I wish my dialogues to be sound and complete, in relation to some standard benchmark. I compare the outcome of my dialogues with the outcome that would be arrived at by a single agent that has as its beliefs the union of both the agents participating in the dialogue's beliefs. This is, in some sense, the 'ideal' situation, where there are no constraints on the sharing of beliefs. I discuss this further in Chapter 5.

In the rest of this chapter I am going to discuss other similar approaches that go someway to providing the features discussed above. First, I present a table, Table 2.1, comparing these works with mine on the four features listed above. As this table shows, none of the existing dialogue systems discussed in this chapter provide all four features. The system that I propose in this thesis does provide all four features.

I will now give an overview of the works other than mine that are mentioned in Table 2.1.

2.2 Amgoud *et al.*

My framework provides two specific protocols for two types of inquiry dialogue. Along with deliberation, this is a dialogue type that has been somewhat overlooked in the field, and there are only two other groups with examples of inquiry protocols. One of these groups is the Liverpool-Toulouse group (Amgoud *et al.*). When the Liverpool-Toulouse group carried out their work, there were very few existing results concerning dialogue protocols, and their work addresses this. They formally defined some protocols for several different dialogue types [3, 2], including inquiry, and intentionally kept these simple, allowing them to explore the protocol properties in some depth [46, 47, 48].

The simplicity of the protocols put forward, whilst being a good place to start, restricts the behaviour of the agents so much that they would not always be usable, particularly in a domain as complicated as the medical domain. For example, let us consider a version of the Liverpool-Toulouse inquiry protocol [48, page 360]. The goal of this protocol is to find, from the union of the agents' beliefs, a set of propositions that will act as the support of an argument for a specific conclusion. This is precisely the same as the goal of my argument inquiry dialogue. In this thesis, I will go on to show that if such a support exists in the union of the agents' beliefs, then the dialogue generated by my system will find it. However, this is not the case of the Liverpool-Toulouse protocol, as you will see from the following example.

Let us assume that we have two agents, I and R, and that these agents are trying to find a support for a. Let an agent X's beliefs be represented by Σ^X.

$$\Sigma^I = \{b \rightarrow a\}$$

$$\Sigma^R = \{c \rightarrow a, c\}$$

We can construct an argument for a from $\Sigma^I \cup \Sigma^R$: $\langle\{c, c \rightarrow a\}, a\rangle$. However, if we follow the Liverpool-Toulouse protocol [48, page 360], then we get the following dialogue (presuming that $b \rightarrow a$ is acceptable to agent R).

$$I : assert : b \rightarrow a$$

$$R : accept : b \rightarrow a$$

$$R : pass$$

$$I : pass$$

The dialogue terminates now, as two $pass$ moves have been made in sequence. The argument for a does not appear in the union of the agents' commitment stores and so the dialogue is unsuccessful, despite the fact that there is potentially an argument to be found. This is not intended to be a criticism of the Liverpool-Toulouse work; the simplicity of the protocols was deliberately enforced by the group in order to allow them to start investigating protocol properties.

Since the theory provided by the Liverpool-Toulouse group is intended for modelling dialogues, it is not generative. That is to say, there is no strategy given that precisely informs an agent which of the available legal moves it should make at any point in a dialogue. They do, however, assign agent attitudes [48, pages 353–355], and these act as partial strategies, restricting the set of legal moves further, but not usually to a unique move. For example, if an agent is *thoughtful*, then it can only assert a proposition p if it can construct an acceptable argument for p, but the Liverpool-Toulouse theory does not describe how an agent must choose between several possible assert moves.

In the Liverpool-Toulouse system, agents are thought of as having certain characteristics, for example, an agent can have a *confident*, *careful* or *thoughtful* assertion attitude, depending on which arguments it is prepared to assert. My strategies were designed with a specific goal in mind regarding the outcome of the dialogue that both participating agents share. It is not clear how the different attitudes of the Liverpool-Toulouse agents would affect the dialogue outcome.

It is not yet clear whether one of these approaches is better than the other, but it seems likely that this will depend on the application for which the framework is intended. The approach that I adopted was to define different strategies to be used with a specific protocol. If an agent in my system enters a dialogue then the outcome of the dialogue is predetermined by the fixed protocol, the strategy being followed, and the belief bases of the agents involved. As the Liverpool-Toulouse group point out [47], this could be viewed as a positive or negative feature. In a safety-critical domain, such as that of breast cancer care, this is a positive property. We want dialogues to always lead us to a certain outcome given the current situation. However, you might imagine that in a more competitive environment we would not want the outcome of the dialogue to be predetermined. We might want it to be possible for an agent to behave intelligently in order to alter the outcome of the dialogue in its favour.

The Liverpool-Toulouse group show that their protocols terminate and also put an upper bound on the number of moves to termination [48]. They provide some complexity results, but these are to do with determining whether an argument for a certain conclusion exists in a knowledge base, and not about the complexity of the dialogue as a whole. In [47], the group consider the outcome of dialogues, however, as their dialogues are not predetermined, they were not able to give results about what specific outcome to expect from a dialogue.

	Inquiry protocol	Generative	Formal	Outcome results
Amgoud *et al.* [3, 2, 46, 47, 48]	YES	NO	YES	YES

2.3 McBurney and Parsons

The other work that provides something that could be classified as an inquiry dialogue is that proposed by McBurney and Parsons. They provide two protocols that each could be classed as inquiry protocols [39, 40]. In one paper [39], they present a protocol for what they call a discovery dialogue. They state that this type of dialogue differs from inquiry dialogues as, in a discovery dialogue, agents are trying to discover a proof for something previously not known. Inquiry dialogues, the authors claim, are concerned with finding a proof for a specific fact. For example, the aim of an inquiry dialogue might be to try to find a proof for a specific risk associated with a situation, whilst the aim of a discovery dialogue would be to try and find proofs for any risks associated with the situation. I feel that McBurney and Parsons' discovery protocol has sufficiently similar goals to what I require for it to be an interesting comparison for my system. I also feel it is unproductive to overly concern oneself with classifying dialogue protocols in terms of the Walton and Krabbe terminology, as it can be somewhat subjective.

The authors present an informal model of the ten stages involved in a discovery dialogue. Examples of some of these stages are [39, page 129]

Open Dialogue: Opening of the discovery dialogue.

Discuss Purpose: Discussion of the purpose of the dialogue.

> Share Knowledge: Presentation of date items relevant to the purpose, drawing only on each individual participant's individual knowledge base.
>
> Discuss Mechanisms: Discussion of potential rules of inference, causal mechanisms, metaphorical modes of reasoning, legal theories, etc.

An informal description of some of the moves that may be made within the discovery dialogue is given, and the authors then go on to claim that each of the ten stages presented can be executed by judicious choice of the available moves. This theory is intended for modelling dialogues and is not generative.

In another paper [40], McBurney and Parsons present a protocol for scientific inquiry. This protocol is presented in a similar manner. They list Hitchcock's Principles of rational mutual inquiry [24], which they desire to hold for their protocol, and then detail the possible moves that may be made. They go a little further than in [39] in defining the behaviour of the system, but it is still not a generative theory and no strategy for choosing the correct move to make is given. Some results are given regarding the properties being upheld, but as these are informal it is more of a discussion than a set of proofs. No results regarding the outcome of dialogues is given.

Neither of these inquiry dialogue systems provides a specific strategy for use by the agents. In fact, there is a clear lack of specific strategies in the literature. One that does exist is that provided by Bench-Capon [5], for what he classifies as a persuasion dialogue but I believe is very similar to my warrant inquiry dialogue. I discuss Bench-Capon's proposal in the following section.

	Inquiry protocol	Generative	Formal	Outcome results
McBurney and Parsons [39, 40]	YES	NO	NO	NO

2.4 Bench-Capon

Bench-Capon does provide agents with the ability to automatically generate dialogues. His paper defines preconditions on moves to be made, telling us whether a move is legal at a point in the dialogue, but beyond that it also describes a strategy that is sufficient to determine which of the available legal moves an agent should make at any point in a dialogue. Interestingly, Bench-Capon states that the participating agents [5, page 6]

> [...] are not intended to "win", but rather to arrive at a position where there is a supported claim on which they agree, together with a fully formed supporting argument structure.

As in my inquiry dialogues, Bench-Capon's agents are concerned with arriving at the best answer given their joint knowledge, not with persuading the other agents to accept its beliefs. I believe that this goal means that this dialogue could actually be classified as an inquiry dialogue, and is equivalent to my warrant inquiry dialogue. Unlike my work, Bench-Capon's strategy is not given formally, and this would lead to problems if someone were to try and investigate this system further. No evaluation of this system is given, and a more formal specification would be needed before the properties of the system could be properly explored.

	Inquiry protocol	Generative	Formal	Outcome results
Bench-Capon [5]	NO	YES	NO	NO

2.5 Sadri *et al.*

Sadri *et al.* provide us with a theory for agent negotiation [60, 61]. They give an approach to defining negotiation protocols in terms of dialogue constraints which take the form of if-then rules: IF you just received performative p AND the conditions in the conjunction $c_1 \wedge \ldots \wedge c_n$ hold THEN make the move p'. This work is interesting to me as the authors consider some similar properties to those I have been considering. They discuss the fact that in order for a protocol to produce exactly one reply at any point in the dialogue then *exactly* one dialogue constraint must fire. They give some examples of protocols and discuss whether this property holds for them. If this property does hold for a protocol then it may be considered to be generative, which is the result that I am interested in.

The authors also consider properties of completeness and correctness for their system, and show that correctness and weak completeness properties hold for certain classes of agents. I feel that it is important to consider these kinds of properties when proposing new dialogue theories, as this allows us to assess the suitability of the theory for different situations. However, as the authors here deal only with the unique negotiation context, it is not possible to make further comparisons between their work and mine.

	Inquiry protocol	Generative	Formal	Outcome results
Sadri *et al.* [60, 61]	NO	NO	YES	YES

2.6 Kakas *et al.*

Kakas *et al.* [28, 29, 30] have proposed a formalism that can be used to represent both public dialogue protocols and private agent strategies. It is a three level formalism. The bottom level gives rules of the form

$$r_{j,i}(Y, S', S) : p_j(X, Y, S') \leftarrow p_i(Y, X, S), c_{ij}$$

where i and j are taken from the set of performatives in use (e.g. assert, request, accept) and c_{ij} (which may be empty) are the conditions under which it the agent X may utter $p_j(X, Y, S')$ upon receiving $p_i(Y, X, S)$ from agent Y. $r_{j,i}(Y, S', S)$ names the rule. These rules are called *dialogue steps* by Kakas *et al.* An example of such a rule [30, page 196] is

$$r_{acc,req}(Y, P, P) : accept(X, Y, P) \leftarrow request(Y, X, P), have(X, P)$$

which states that if an agent has a resource and receives a message requesting that resource then it can accept the request.

The next level consists of rules that, given a set of particular circumstances, assign a higher priority to one dialogue step than to another. At the top level there are rules that take the same form as those at the middle level but only assign priorities to rules from the top two levels. Kakas *et al.* use this formalism to represent both the private agent strategy and the public dialogue protocol. Argumentation is then applied to these representations to determine either the utterance to be made (in the case of the strategy) or the set of legal utterances (in the case of the protocol).

When representing strategies, Kakas *et al.* assume the existence of some dialogue steps that express the general requirement that making two different moves at the same time is not allowed (i.e. at most one move to be made will be generated by the strategy). They discuss the fact that in order for a strategy to produce at least one utterance it must be exhaustive. They do not formally define this term but state "in the sense that the conditions of at least one of its [dialogue steps] are always satisfied" [30, page 200]. They then define what it means for a strategy to be hierarchical and give the theorem that if a strategy is exhaustive and hierarchical, and its priority relation does not contain any cycles of length greater than two, then it will always produce exactly one move to utter.

Kakas *et al.* are concerned with providing a flexible formalism for representing both private agent strategies and public dialogue protocols. They do not provide an inquiry protocol and they do not provide any specific strategies. Although they define the properties required for a strategy to be generative they do not give such a strategy. As this is the case, they are not able to consider the outcome of specific dialogues and do not provide any outcome results.

	Inquiry protocol	Generative	Formal	Outcome results
Kakas *et al.* [28, 29, 30]	NO	NO	YES	NO

2.7 Amgoud and Hameurlain

Like Kakas *et al.*, Amgoud and Hameurlain [1] also provide a formal framework for defining private agent strategies. They claim that deciding on a move to utter is a two stage process; first the agent must decide what type of move to make (e.g. assert, accept, offer), then the agent must decide on the most suitable content for this move. Amgoud and Hameurlain assume that an agent consists of strategic goals, strategic beliefs, functional goals and basic beliefs. Strategic goals and strategic beliefs are meta-level concepts which are generally independent of the subject of the dialogue and are used to determine what type of move should be made. The functional goals and basic beliefs are used to determine the content of the move. Functional goals represent what an agent wishes to achieve regarding the subject of the dialogue. Basic beliefs refer to the environment and to the subject of the dialogue.

Amgoud and Hameurlain define an argumentation-based decision model that is used to compute firstly the most preferred type of move to make and then the best content for this move, based on the different types of goals and beliefs as discussed in the previous paragraph. If there is more than one most preferred move type then one is selected at random, similarly for the best content, and so their system does ensure that exactly one move for utterance is returned by the strategy.

Amgoud and Hameurlain are interested in providing a formalism for representing agent strategies and a decision model which uses this representation to select a move to utter. They do not, however, provide an inquiry protocol. They give one very specific example relating to negotiation of the price of an object but do not provide a strategy for use in inquiry dialogues and so do not allow generation of inquiry dialogues. They do not provide any results about the outcome of dialogues.

	Inquiry protocol	Generative	Formal	Outcome results
Amgoud and Hameurlain [1]	NO	NO	YES	NO

2.8 Pasquier and Chaib-draa

A completely different approach to dialogue strategy is Pasquier and Chaib-draa's cognitive coherence approach [50, 49, 51, 52]. They, like me, are particularly interested in providing agents with the ability to dynamically generate dialogues. Pasquier and Chaib-draa are looking for answers to pragmatic questions such as "When should an agent enter into a dialogue?", "Who should the agent enter into a dialogue with?", "What type of dialogue should be entered into? And on what topic?", "How should an agent behave within a dialogue?", "When should a dialogue terminate?", and "How does a dialogue impact on an agent's private beliefs?".

Their theory extends and adapts a major social psychology theory called the cognitive dissonance theory. The cognitive dissonance theory appeals to the concept of homoeostasis, and tries to reduce the dissonance between cognitive elements. In Pasquier and Chaib-draa's formulation of the theory, the cognitive elements are both the agent's private cognitions (beliefs, desires and intentions) and the agent's public cognitions (social commitments). Elements can be either *accepted* (interpreted as true, activated or valid depending on the element's type), or *rejected* (interpreted as false, inactivated or not valid depending on the element's type). Depending on the pre-existing relations that hold between them, two types of non-ordered, binary constraints between elements are inferred.

- **Positive constraints** are inferred from positive relations such as: explanation relations, deduction relations, facilitation relations.
- **Negative constraints** are inferred from negative relations such as: mutual exclusion, incompatibility, inconsistency.

Weights are assigned to these constraints, although one can imagine that it may be a difficult task to come up with these numbers, as the only guidance as to how these weights are assigned given in [51] is that they should reflect "the importance and validity degree for the underlying relation". A constraint may be *satisfied* or not. A positive constraint is said to be satisfied if and only if the two elements that it binds are either both accepted or both rejected. A negative constraint is said to be satisfied if and only if one of the two elements that it binds is accepted and the other rejected. Two elements are said to be *coherent* if they are connected by a satisfied constraint, or *incoherent* if they are connected by a non-satisfied constraint.

The basic principle behind the theory is that an incoherence produces a tension in an agent which incites the agent to act somehow, in order to reduce the incoherence. This may mean altering an private element, such as a belief or an intention, or it may mean altering a public social commitment. In the case of altering a social commitment, the agent must enter into a dialogue. The agent chooses what type of dialogue to enter into, who with, and on what topic by matching the expected successful outcome of the dialogue with a new commitment that will reduce the incoherence.

The cognitive coherence approach is intended to be an all encompassing approach to agent behaviour. Not only should it allow an agent to determine when to enter into a dialogue, what type of dialogue that should be, who it should be with and what topic that dialogue should be on, it should also allow an agent to decide which of all the legal messages to send at any point in a dialogue. However, there is no formal account of how this should occur. The general principle of maximising coherence is explained but no formal details are given.

It is not very clear how the constraints between the elements are inferred, as no formal description is given. For example, imagine an agent had two beliefs, one for b and one for $\neg b$; it seems clear that these two elements would be linked by a negative relation. Now, imagine that the agent had the following beliefs: a, $a \rightarrow b$, b, $\neg b$. What constraints should be attached to these elements? Should we link a and b with a positive constraint as a is a member of the support of an argument whose conclusion is b? A more difficult question is should we have a negative constraint from $\neg b$ to either a or $a \rightarrow b$? $\neg b$ is not inconsistent with either a or $a \rightarrow b$ on their own but if you put them all together then you get an inconsistency.

The only validation of this theory that is given is an example of a dialogue carried out by two agents in a dialogue game simulator that was developed in the authors' lab. We are told nothing about the expected behaviour of agents. For example, it may be possible to enter into an infinite dialogue using this theory, something that would be unacceptable in the medical domain. Nothing is said about the outcome of the dialogues that are carried out under this theory.

Pasquier and Chaib-draa's theory is, in one sense, very appealing, as it sets out to give a unified approach that addresses several fundamental issues that have been mostly overlooked in the agent dialogue literature. However, it is hard to understand how this theory would be applied to an argumentation-based inquiry dialogue, such as those supported by my system.

	Inquiry protocol	Generative	Formal	Outcome results
Pasquier and Chaib-draa [50, 49, 51, 52]	NO	YES	NO	NO

2.9 Prakken

Another piece of work that was of particular interest to me is Prakken's work on coherence and flexibility in dialogue games [54, 55, 56]. Prakken proposes a formal framework for modelling, but not generating, persuasion dialogues [56]. These take place between two agents that he denotes P (for proponent) and

O (for opponent). P moves first, either claiming that the topic of the dialogue is true, or putting forward an argument for that topic. It is up to P to try and persuade O that the topic of the dialogue holds. Prakken is particularly focussed on persuasion dialogues, as he comes from an artificial intelligence and law background. My framework does not currently deal with persuasion dialogues, although it seems reasonable to expect that it could be extended to do so. However, although Prakken is considering a different dialogue type to me, our systems are similar.

Prakken's framework is modular, allowing the use of different underlying logics, various sets of locutions, different rules for commitment updates, alternative turntaking rules, and various sets of protocol rules. It also allows different rules on whether multiple replies to a move, postponing of replies, and backtracking (returning to earlier choices and making alternative moves) are allowed. My system is also modular, although it was mainly with the intention of altering the protocol and strategy rules that this design choice was made. Prakkken's framework does not provide any strategy rules and this is one of the major differences between Prakken's and my work. Unlike my system, agents cannot automatically generate dialogues using Prakken's framework alone, as there is no strategy component given. This is not an oversight of Prakken, rather it highlights the different background from which we are approaching this problem; Prakken is interested in modelling realistic legal dialogues, whereas it is my intention to provide agents with the ability to generate dialogues.

Prakken's framework distinctively imposes an explicit reply structure on dialogues. Each dialogue move either explicitly attacks or explicitly surrenders to some earlier move of the other participant. In most systems, such as mine, this is not made explicit. Within my framework it is only possible to make attacking moves, and the moves that these are aimed at are left implicit. It may be the case that a move made within my system attacks more than one previous move.

Prakken defines something called the *dialogical status* of a move, which may be either *in* or *out*. This allows Prakken to consider the outcome of dialogues. A move in a dialogue is said to be *surrendered* in a dialogue if and only if either it is a move putting forward an argument A and there is a reply to that move that concedes A's conclusion, or there is a surrendering reply to that move. A move in a dialogue is said to be *in* if and only if either the move is surrendered in the dialogue, or else all the attacking replies to that move are *out*; otherwise the move is *out*. The current winner of a dialogue is defined to be P if and only if the first move of the dialogue is *in*, and is O otherwise.

Prakken goes on to investigate some soundness and fairness results. His concepts of soundness and fairness are similar, but perhaps not as strong, as my concepts of soundness and completeness. Interestingly, Prakken compares the outcome of the dialogue to the dialectical tree that is implicitly constructed during the dialogue. I explicitly define the dialectical tree constructed during a dialogue (which I call the *dialogue tree*), but I use this tree to determine the outcome of the dialogue, and compare this with the dialectical tree constructed from the union of the two participating agents' beliefs.

Let me explore Prakken's soundness and fairness results further. The outcome of one of Prakken's dialogues depends on the status of the first move of the dialogue, which depends on the following moves that have been made. In order to investigate soundness and fairness results of his system, Prakken compares this outcome to a dialectical tree constructed from all of the things asserted so far in the dialogue. So both the dialogue outcome and the dialectical tree are dependent on the moves made, and hence arguments asserted, throughout the dialogue. Prakken is checking whether the labelling function that labels moves *in* or *out* does indeed leave you with the outcome that you would expect, given the arguments that have been moved during the dialogue. I compare my dialogue outcome with that which you would get if you merged the two agents' knowledge and carried out the reasoning internally.

Prakken is particularly interested in how consideration of relevance and focus of moves enforce various degrees of flexibility and coherence of dialogues. He defines a move as being relevant if and only if it changes the status of the initial move of the dialogue. Prakken defines a new version of his protocol which has the added rule that moves made must be relevant. His motivation for this is that if this relevance is not enforced, then we end up with unintuitive dialogues. This is particularly undesirable for Prakken, as he is trying to model natural dialogues. Prakken states that his soundness and fairness results easily translate from his liberal to his relevant dialogues, but he makes no evaluation as to whether the relevant protocol is better in any way.

I have also been considering how to ensure that moves made are relevant. This is evident in some of the rules of my protocols. For example, I restrict an agent to only being able to assert an argument that attacks something that has already been asserted. I have also considered relevance of moves at a strategical level. The motivation behind the pruned tree strategy (defined in Chapter 6) is for the agents to use the most relevant move at any point in a dialogue.

	Inquiry protocol	Generative	Formal	Outcome results
Prakken [54, 55, 56]	NO	NO	YES	YES

2.10 Summary

To conclude this chapter, there are no existing, formally specified dialogue systems that are sufficient to automatically generate inquiry dialogues, and which provide results about the behaviour and outcome of such dialogues. My work addresses this shortcoming.

In the next chapter I present García and Simari's argumentation system [22], which is intended for internal reasoning by a single agent.

Chapter 3

System for internal argumentation within a single agent

In this section I introduce an argumentation system intended for internal reasoning, which is an adaption of García and Simari's Defeasible Logic Programming system (DeLP) [22]. DeLP is a formalism that combines results of logic programming and defeasible argumentation. It allows an agent to reason with inconsistent, uncertain and incomplete knowledge that may change dynamically over time, making it ideal for dealing with knowledge from the medical domain.

In DeLP, a query q is successful when there is a warranted argument A for q. That is to say, there is an argument A whose claim is q, and A is not defeated. In order to determine whether A is defeated or not, we must consider each possible defeater for A in turn. For each of these defeaters, all of their possible defeaters must be considered, and so on. This is made possible by a dialectical analysis of A. DeLP provides us with a warrant procedure that implements this dialectical analysis. Therefore, if the argument A for q is found undefeated by this warrant procedure, then that means that the query q is successful. García and Simari also impose some constraints on the warrant procedure for avoiding undesirable situations, such as producing an infinite sequence of defeaters.

I choose DeLP as the system for internal argumentation as it is sufficient for my needs without being too complicated, and as it provides a nice visualisation for the warrant procedure. I ruled out abstract argumentation systems such as Dung's [13], as my system needs to be able to deal with specific arguments (i.e. I want to generate arguments from a knowledge base so that the support for each argument is a set of formulae). I then considered argumentation systems based on classical logic, such as that of Besnard and Hunter [6, 7]. Classical logic is a rich formalism that is very expressive, and as such it is an attractive option for representing the medical domain. However, my system is a complicated one with several interacting components and processes, and in order to be able to provide concrete results regarding the outcome of dialogues in my system I needed something more restrictive than classical logic.

DeLP restricts the language and types of inference, giving us something less complicated than classical logic. It also appears easier to edit a defeasible logic knowledgebase to provide particular arguments in particular situations [7, Chapter 11], as we might wish to do in a system such as CREDO. The emphasis of this thesis is a general framework for dialogue generation and so it would be interesting to consider the use of other defeasible logics other than DeLP (such as [64] or [57]) and to see what effect this might have on the system. DeLP is a recent proposal that has benefited from being able to incorporate ideas from other proposals for defeasible reasoning. It also has the added advantage of being implemented[1] and so examples can be tested.

The reader should note that this chapter is very closely based on [22]. The system used in my work is a subset of DeLP, with only minor changes from [22]. The only significant difference is that García and Simari assume the existence of facts, strict rules and defeasible rules. I want a system that will deal with knowledge that is entirely defeasible. I have dealt with this by assuming that the set of strict rules is empty and by making facts defeasible.

3.1 Knowledge representation

García and Simari define the DeLP language in terms of three disjoint sets: a set of facts, a set of strict rules and a set of defeasible rules. However, I am dealing with knowledge from the medical domain, that we know to often be incomplete, unreliable and inconsistent, hence I want all of the knowledge to be treated as defeasible. Medical knowledge is constantly expanding. What we might have thought of as a strict rule ten years ago we may now realise is defeasible in certain situations. I want facts to also be considered as defeasible for two reasons. Firstly, it is conceivable that a mistake may have been made, as the medical domain is highly stressful and occasional human error is unavoidable. Secondly, I want it to be possible for a set of facts to be inconsistent, as it is possible that a doctor may have inconsistent beliefs. For example, a patient's set of symptoms might strongly suggest a certain disease but, when the test for that disease is carried out, the results come back negative. The doctor would then have reasons to believe that the patient did have that disease and reasons to believe that she did not.

In order to address this issue, I have left out García and Simari's definitions of a fact and a strict rule, and added a new definition of a defeasible fact. Note that I use a propositional logic in which a literal is either an atom α or a negated atom $\neg\alpha$. This is a slightly simplified version of García and Simari, who assume a first-order logic in which literals are either ground atoms or negated ground atoms. I have chosen to use propositional logic for ease of presentation.

Definition 3.1.1 *A* **defeasible rule** *is an ordered pair, denoted "$Body \rightarrow Head$" whose first member, $Body$, is a finite, non-empty set of literals, and whose second member, $Head$, is a literal. For ease of reading, I will denote a defeasible rule in which $Body$ is $\{\alpha_1, \ldots, \alpha_n\}$ and $Head$ is β as: $\alpha_1 \wedge \ldots \wedge \alpha_n \rightarrow \beta$ (this is slightly different to the notation in [22]).*

Note that the symbols $\wedge$ and $\rightarrow$ are not being used here to represent classical conjunction or implication. They represent meta-relations between sets of literals. In particular, there is no contraposition.

I will now define a defeasible fact. This is simply a literal, as is a fact in García and Simari's system. Unlike García and Simari, however, I consider facts as defeasible.

Definition 3.1.2 *A* **defeasible fact** *is a literal, α.*

[1] http://lidia.cs.uns.edu.ar/delp_client

The warrant procedure defined by García and Simari assumes that a formal criterion exists for comparing two arguments. This criterion allows an agent to decide whether one argument defeats another or not. García and Simari do not specify what this criterion should be, but do give two examples: specificity of an argument, and a preference relation between arguments. I have decided to define a preference relation on agent beliefs for use as the comparison criterion.

The choice of comparison criterion is by no means trivial. It is not the main focus of my PhD and much more time could be spent investigating other criteria. However, preference ordering is a common approach which seems to be suited to the medical domain. Medical knowledge arises as a result of clinical trials, published work, laboratory work etc., and some sources may be regarded to be more reliable than others. For example, if we encode some knowledge from an established clinical guideline, then you might assume this to be preferred to knowledge that has resulted from a small clinical trial, hence we can imagine some sensible way of associating a preference level with a belief that depends on where the belief has arisen from. I will now go on to define a belief. This is not something that appears in García and Simari [22]. It allows us to associate preference levels with defeasible facts and defeasible rules.

A belief is either a defeasible fact or a defeasible rule that has associated with it a preference level. This is an integer that gives an indication of how confident the agent is in the belief. The lower the preference level the more strongly the agent believes the belief. Note that I denote the set of natural numbers as $\mathbb{N} = \{1, 2, 3, \ldots\}$.

Definition 3.1.3 *A* **belief** *is a pair* (ϕ, L) *where* ϕ *is either a defeasible fact or a defeasible rule, and* $L \in \mathbb{N}$ *is a label that denotes the* **preference level** *of the belief. The function* pLevel *returns the preference level of the belief:* $\mathsf{pLevel}((\phi, L)) = L$*. The function* bel *returns the first element of the belief:* $\mathsf{bel}((\phi, L)) = \phi$*. The set of all beliefs is denoted* $\mathcal{B}$*.*

I also make a distinction between beliefs in defeasible facts and beliefs in defeasible rules. Beliefs in defeasible facts are called *state beliefs*, as these are beliefs about the specific state of the environment. Beliefs in defeasible rules are called *domain beliefs*, as these are beliefs about how the domain is expected to behave. Note that all beliefs are defeasible.

Definition 3.1.4 *A* **state belief** *is a belief* (ϕ, L) *where* ϕ *is a defeasible fact. The set of all state beliefs is denoted* $\mathcal{S}$*.*

Definition 3.1.5 *A* **domain belief** *is a belief* (ϕ, L) *where* ϕ *is a defeasible rule. The set of all domain beliefs is denoted* $\mathcal{R}$*.*

I also define associated sets that include only the first element of a belief, i.e. the set of all defeasible rules and defeasible facts, the set of all defeasible facts, and the set of all defeasible rules.

Definition 3.1.6 *The set of all defeasible rules and defeasible facts is denoted* $\mathcal{B}^\star = \{\phi \mid (\phi, L) \in \mathcal{B}\}$*. The set of all defeasible facts is denoted* $\mathcal{S}^\star = \{\phi \mid (\phi, L) \in \mathcal{S}\}$*. The set of all defeasible rules is denoted* $\mathcal{R}^\star = \{\phi \mid (\phi, L) \in \mathcal{R}\}$*.*

Note that $\mathcal{B}^\star = \mathcal{S}^\star \cup \mathcal{R}^\star$, as a belief is either a belief in a defeasible rule or a belief in a defeasible fact.

Each agent is identified by a unique id x taken from a set $\mathcal{I}$. I will refer to an agent that is uniquely identified by x, such that $x \in \mathcal{I}$, simply as x. Each agent has associated with it a, possibly inconsistent, finite belief base. I assume that an agent's belief base is fixed, at least for the duration of a dialogue. That is to say, a belief base does not change over time.

Definition 3.1.7 *A* **belief base** *associated with an agent x is a finite set Σ^x such that $\Sigma^x \subseteq \mathcal{B}$ and $x \in \mathcal{I}$.*

Example 3.1.1 *Consider the following belief base associated with agent x_1.*

$$\Sigma^{x_1} = \left\{ \begin{array}{l} (a,1), (\neg a, 1), (b, 2), (d, 1), \\ (a \rightarrow c, 3), (b \rightarrow \neg c, 2), (d \rightarrow \neg b, 1), (\neg a \rightarrow \neg b, 1) \end{array} \right\}$$

The top four elements are state beliefs and you can see that the agent conflictingly believes strongly in both a and $\neg a$. The bottom four elements are all domain beliefs.
$\mathsf{pLevel}((a,1)) = 1$. $\mathsf{pLevel}((\neg a,1)) = 1$. $\mathsf{pLevel}((b,2)) = 2$. $\mathsf{pLevel}((d,1)) = 1$. $\mathsf{pLevel}((a \rightarrow c,3)) = 3$. $\mathsf{pLevel}((b \rightarrow \neg c,2)) = 2$. $\mathsf{pLevel}((d \rightarrow \neg b,1)) = 1$. $\mathsf{pLevel}((\neg a \rightarrow \neg b,1)) = 1$.
Recall that the lower the pLevel *value, the more preferred the belief.*

I will now define what constitutes a defeasible derivation. This has been adapted from the García and Simari definition in order to deal with my definition of a belief.

Definition 3.1.8 *Let Ψ be a set of beliefs and α a defeasible fact. A* **defeasible derivation** *of α from Ψ, denoted $\Psi \mid\!\sim \alpha$, is a finite sequence $\alpha_1, \alpha_2, \ldots, \alpha_n$ of literals such that α_n is the defeasible fact α and each α_m, $1 \leq m \leq n$, is in the sequence because either*

- *(α_m, L) is a state belief in Ψ, or*
- *there exists a domain belief $(\beta_1 \wedge \ldots \wedge \beta_j \rightarrow \alpha_m, L')$ in Ψ such that every literal β_i, $1 \leq i \leq j$, is a literal α_k of the sequence appearing before α_m ($k < m$).*

It may be possible to defeasibly derive conflicting defeasible facts from the same belief base. This can be seen in the following example, where defeasible derivations of, for instance, b and $\neg b$ have been constructed from the same belief base.

Example 3.1.2 *If we continue with the running example started in Example 3.1.1 we see that the following defeasible derivations exist from Σ^{x_1}.*
a. $\neg a$. b. d. a, c. $b, \neg c$. $d, \neg b$. $\neg a, \neg b$.

I also define a function that takes a set of beliefs Ψ and returns the set of literals that can be defeasibly derived from Ψ.

Definition 3.1.9 *The function* DefDerivations : $\wp(\mathcal{B}) \mapsto \mathcal{S}^\star$ *returns the* **set of literals that can be defeasibly derived** *from a set of beliefs Ψ, $\Psi \subseteq \mathcal{B}$, such that* $\mathsf{DefDerivations}(\Psi) = \{\phi \mid$ *there exists* $\Phi \subseteq \Psi$ *such that* $\Phi \mid\!\sim \phi\}$.

Defeasible derivation is used to infer the claim of an argument from its support, as I will show in the next section.

3.2 Argument structure

In this section, García and Simari's defeasible argumentation formalism will be introduced. First I define an argument as, informally, a minimally consistent set from which the claim can be defeasibly derived.

Definition 3.2.1 *An* **argument** *A constructed from a, possibly inconsistent, belief base Ψ, $\Psi \subseteq \mathcal{B}$, is a tuple $\langle \Phi, \phi \rangle$ where ϕ is a literal and Φ is a set of beliefs such that*

1. *$\Phi \subseteq \Psi$,*
2. *$\Phi \mid\sim \phi$,*
3. *$\forall \phi, \phi'$ such that $\Phi \mid\sim \phi$ and $\Phi \mid\sim \phi'$, it is not the case that $\phi \cup \phi' \vdash \perp$ (where $\vdash$ represents classical implication), and*
4. *Φ is minimal: there is no proper subset Φ' of Φ such that Φ' satisfies conditions (1), (2) and (3).*

Φ is called the **support** *of the argument, and is denoted $\mathsf{Support}(A)$. ϕ is called the* **claim** *of the argument, and is denoted $\mathsf{Claim}(A)$. The set of all arguments that can be constructed from a knowledge base Ψ is denoted $\mathcal{A}(\Psi)$.*

Example 3.2.1 *Continuing the running example, the following arguments can be constructed by the agent.*

$$
\begin{array}{ll}
a_1 = \langle \{(a, 1)\}, a \rangle & a_5 = \langle \{(a, 1), (a \rightarrow c, 3)\}, c \rangle \\
a_2 = \langle \{(\neg a, 1)\}, \neg a \rangle & a_6 = \langle \{(b, 2), (b \rightarrow \neg c, 2)\}, \neg c \rangle \\
a_3 = \langle \{(b, 2)\}, b \rangle & a_7 = \langle \{(d, 1), (d \rightarrow \neg b, 1)\}, \neg b \rangle \\
a_4 = \langle \{(d, 1)\}, d \rangle & a_8 = \langle \{(\neg a, 1), (\neg a \rightarrow \neg b, 1)\}, \neg b \rangle
\end{array}
$$

I now define the sub-argument relation. In the next section, where we consider conflicting arguments, we will see that it is possible for one argument to attack another argument by being in conflict with a sub-argument of that argument.

Definition 3.2.2 *Let A_1 and A_2 be two arguments. A_1 is a* **sub-argument** *of A_2 iff $\mathsf{Support}(A_1) \subseteq \mathsf{Support}(A_2)$. This is denoted $A_1 \sqsubseteq A_2$.*

Example 3.2.2 *Continuing the running example, we see that $a_1 \sqsubseteq a_5$, $a_2 \sqsubseteq a_8$, $a_3 \sqsubseteq a_6$ and $a_4 \sqsubseteq a_7$. Note also that an argument is always a sub-argument of itself so $a_1 \sqsubseteq a_1$, $a_2 \sqsubseteq a_2$ etc.*

If we consider Example 3.2.1 then we see that it is possible to construct two arguments from the same belief base that have contradictory claims, for example $\mathsf{Claim}(a_1) = a$ and $\mathsf{Claim}(a_2) = \neg a$, and so these arguments are in conflict.

3.3 Conflicts and counter-arguments

As Ψ may be inconsistent, there may be conflicts between arguments within $\mathcal{A}(\Psi)$. Two arguments are said to conflict with one another if and only if the union of their claims is inconsistent.

Definition 3.3.1 *Let A_1 and A_2 be two arguments. A_1 is in* **conflict** *with A_2 iff* $\mathsf{Claim}(A_1) \cup \mathsf{Claim}(A_2) \vdash \perp$, *where $\vdash$ represents the classical consequence relation (i.e.* $\mathsf{Claim}(A_1) = \neg\mathsf{Claim}(A_2)$ *as the claim of an argument is always a literal) . This is denoted $A_1 \bowtie A_2$.*

Example 3.3.1 *Continuing the running example, $a_1 \bowtie a_2$, $a_3 \bowtie a_7$, $a_3 \bowtie a_8$, and $a_5 \bowtie a_6$.*

I now use the previous definitions to define an attack relation between arguments. This is slightly different from the definition in [22], as it has been altered to include the sub-argument that is being specifically attacked. This is so that, when we are using the preference levels to determine whether an argument defeats another or not, we can base the comparison on the preference level of the specific sub-argument that is being attacked, as this may differ from the preference level of the entire argument.

Definition 3.3.2 *Let A_1, A_2 and A_3 be arguments. A_1* **attacks** *A_2 at A_3 iff $A_3 \sqsubseteq A_2$ and $A_1 \bowtie A_3$. This is denoted $A_1 \rhd A_2(A_3)$. A_1 is called a* **counter-argument** *for A_2, and A_3 is called the* **disagreement sub-argument** *of A_1 attacking A_2.*

Note that, in the previous definition, the disagreement sub-argument A_3 is unique. A_1 and A_3 conflict ($A_1 \bowtie A_3$), and so the claim of A_1 is the negation of the claim of A_3. Let us assume that there is another disagreement sub-argument such that A_1 attacks A_2 at A_4, A_1 and A_4 conflict, and so the claim of A_1 is the negation of the claim of A_4. As A_1 is a literal, this means that the claim of A_4 is the same as the claim of A_3. As an argument is minimal (Definition 3.2.1, condition 4), this means that A_3 and A_4 must be the same arguments.

Example 3.3.2 *Continuing the running example, the following counter-argument relations hold.*

$a_1 \rhd a_2(a_2)$	$a_1 \rhd a_8(a_2)$	$a_2 \rhd a_1(a_1)$
$a_2 \rhd a_5(a_1)$	$a_3 \rhd a_7(a_7)$	$a_3 \rhd a_8(a_8)$
$a_5 \rhd a_6(a_6)$	$a_6 \rhd a_5(a_5)$	$a_7 \rhd a_3(a_3)$
$a_7 \rhd a_6(a_3)$	$a_8 \rhd a_3(a_3)$	$a_8 \rhd a_6(a_3)$

Let us imagine that an agent can construct two arguments from its set of beliefs, A_1 and A_2, and that A_1 is a counter-argument for A_2. The agent must make a decision about whether A_1 defeats A_2 or not, and so it needs some kind of criterion for judging which is a more powerful argument. García and Simari do not specify a comparison criterion that must be used but suggest two examples, one which relates to the specificity of arguments and one which uses a preference relation defined among defeasible rules. As discussed earlier, I have decided to specify a comparison criterion based on the preference ordering of the beliefs. This is very similar to the comparison of the preference level of rules suggested by García and Simari but also considers the preference level of defeasible facts. My comparison criterion will be introduced in the following section.

3.4 Comparing arguments

When deciding whether one argument defeats another or not, an agent in my system will consider the preference level of each argument. This gives an instantiation of García and Simari's framework. The preference level of an argument is equal to the preference level of the least preferred belief used in its support.

Definition 3.4.1 *Let A be an argument. The* **preference level** *of A, denoted $\mathsf{pLevel}(A)$, is equal to $\mathsf{pLevel}(\phi)$ such that*

1. $\phi \in \mathsf{Support}(A)$
2. *for all* $\phi' \in \mathsf{Support}(A), \mathsf{pLevel}(\phi') \leq \mathsf{pLevel}(\phi)$

Example 3.4.1 *Continuing the running example, the arguments have the following preference levels.*

$$\mathsf{pLevel}(a_1) = 1 \qquad \mathsf{pLevel}(a_2) = 1 \qquad \mathsf{pLevel}(a_3) = 2 \qquad \mathsf{pLevel}(a_4) = 1$$

$$\mathsf{pLevel}(a_5) = 3 \qquad \mathsf{pLevel}(a_6) = 2 \qquad \mathsf{pLevel}(a_7) = 1 \qquad \mathsf{pLevel}(a_8) = 1$$

One argument is strictly preferred to another if the preference level of the first argument is less than that of the second.

Definition 3.4.2 *Let A_1 and A_2 be two arguments. A_1 is* **strictly preferred** *to A_2 iff $\mathsf{pLevel}(A_1) < \mathsf{pLevel}(A_2)$. This is denoted as $A_1 >_p A_2$.*

Example 3.4.2 *Continuing the running example*

$$a_1 >_p a_3 \qquad a_2 >_p a_3 \qquad a_4 >_p a_3 \qquad a_7 >_p a_3 \qquad a_8 >_p a_3$$

$$a_1 >_p a_5 \qquad a_2 >_p a_5 \qquad a_4 >_p a_5 \qquad a_7 >_p a_5 \qquad a_8 >_p a_5$$

$$a_1 >_p a_6 \qquad a_2 >_p a_6 \qquad a_4 >_p a_6 \qquad a_7 >_p a_6 \qquad a_8 >_p a_6$$

$$a_3 >_p a_5 \qquad a_6 >_p a_5$$

Observe that the preference level of an argument may differ from that of its sub-arguments, and so it makes sense to base the comparison between arguments on the preference level of the sub-argument that is being attacked. Assume that given two arguments, A_1 and A_2, we know that A_1 attacks A_2 at disagreement sub-argument A_3. In some cases A_1 defeats A_2, and in some cases A_1 does not defeat A_2. The agent makes the decision as to which of these cases holds depending on the preference levels of A_1 and A_3. In order to defeat the argument A_2, the preference level of A_1 must be the same or less (meaning more preferred) than the preference level of A_3. If it is the same, then A_1 is a *blocking defeater* for A_2. If it is less, then A_1 is a *proper defeater* for A_2.

Definition 3.4.3 *Let A_1, A_2 and A_3 be arguments such that A_3 is the disagreement sub-argument of A_1 attacking A_2. A_1 is a* **proper defeater** *for A_2 iff*

1. $A_1 \rhd A_2(A_3)$,

2. $A_1 >_p A_3$.

This is denoted as $A_1 \Rightarrow_p A_2$.

Definition 3.4.4 *Let* A_1*,* A_2 *and* A_3 *be arguments such that* A_3 *is the disagreement sub-argument of* A_1 *attacking* A_2*.* A_1 *is a* **blocking defeater** *for* A_2 *iff*

1. $A_1 \rhd A_2(A_3)$,

2. $A_1 \not>_p A_3$,

3. $A_3 \not>_p A_1$.

This is denoted as $A_1 \Rightarrow_b A_2$.

Example 3.4.3 *Continuing the running example,* $a_1 \Rightarrow_b a_2$*,* $a_1 \Rightarrow_b a_8$*,* $a_2 \Rightarrow_b a_1$*,* $a_2 \Rightarrow_b a_5$*,* $a_6 \Rightarrow_p a_5$*,* $a_7 \Rightarrow_p a_3$*,* $a_7 \Rightarrow_p a_6$*,* $a_8 \Rightarrow_p a_3$ *and* $a_8 \Rightarrow_p a_6$*.*

If A_1 is a *proper* defeater for A_2, then it is not possible for A_2 to be a defeater of A_1. However, if A_1 is a *blocking* defeater for A_2, then there will always be some sub-argument of A_2 that is a blocking defeater for A_1. This result is not given by García and Simari so I give it here for completeness.

Proposition 3.4.1 *Let* A_1 *and* A_2 *be arguments. If* $A_1 \Rightarrow_b A_2$ *then there exists* A_3 *such that* $A_3 \sqsubseteq A_2$ *and* $A_3 \Rightarrow_b A_1$*.*

Proof: *Assume* $A_1 \Rightarrow_b A_2$*. From this, the definition of blocking defeaters (Definition 3.4.4), and the definition of strict preference (Definition 3.4.2), we get that there exists* $A_3 \sqsubseteq A_2$ *such that* $A_1 \rhd A_2(A_3)$*, and* $\mathsf{pLevel}(A_1) = \mathsf{pLevel}(A_3)$*. From the definition of attack (Definition 3.3.2) and the definition of conflict (Definition 3.3.1), we get that* $\mathsf{Claim}(A_1) \cup \mathsf{Claim}(A_3) \vdash \perp$*.* $A_1 \sqsubseteq A_1$*, from the definition of a sub-argument (Definition 3.2.2), hence, from the definition of attack (Definition 3.3.2), we get that* $A_3 \rhd A_1(A_1)$*. From the definition of a blocking defeater (Definition 3.4.4), we get that* $A_3 \Rightarrow_b A_1$*.* $\square$

In this section I proposed a criterion that allows us to decide whether an argument A_1 that conflicts with an argument A_2 defeats A_2 or not. In the next section I introduce García and Simari's method for visualising defeat relations that occur within a set of interacting arguments.

3.5 Argument trees and acceptable argumentation lines

It is often the case that there are many different interactions within a set of arguments. A useful method for visualising these interactions is to use an *argument tree*. Each node in an argument tree represents an argument, and the branches between the nodes represent the defeat relation (blocking or proper).

Definition 3.5.1 *An* **argument tree***,* T*, is a tree in which every node* N *is labelled with an argument* A*. Any child of* N*,* N_i*, is labelled with an argument* A_i *such that* A_i *is a defeater for* A*. The* **level of a node***,* N*, is the number of arcs on the path from the root to node* N*, and this value is returned by the function* $\mathsf{Level}(N)$*. The* **label of a node** *is returned by the function* Label *such that* $\mathsf{Label}(N) = A$ *iff*

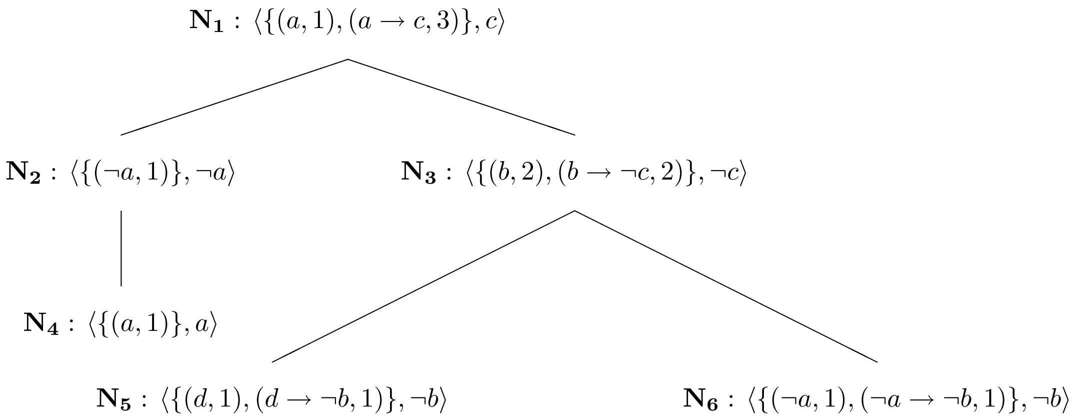


Figure 3.1: An argument tree.

node N is labelled with A. The set of all the nodes in an argument tree T is denoted $\mathsf{Nodes}(T)$*. The* **root node** *of an argument tree T is denoted* $\mathsf{Root}(T)$.

Example 3.5.1 *Continuing the running example, an example of an argument tree is shown in Figure 3.1. For each node, the name of the node is given followed by the argument that the node is labelled with.*
$\mathsf{Level}(N_1) = 0$, $\mathsf{Level}(N_2) = 1$, $\mathsf{Level}(N_3) = 1$, $\mathsf{Level}(N_4) = 2$, $\mathsf{Level}(N_5) = 2$, *and* $\mathsf{Level}(N_6) = 2$.
$\mathsf{Label}(N_1) = \langle\{(a,1),(a \rightarrow c,3)\},c\rangle$, $\mathsf{Label}(N_2) = \langle\{(\neg a,1)\},\neg a\rangle$, $\mathsf{Label}(N_3) = \langle\{(b,2),(b \rightarrow \neg c,2)\},\neg c\rangle$, $\mathsf{Label}(N_4) = \langle\{(a,1)\},a\rangle$, $\mathsf{Label}(N_5) = \langle\{(d,1),(d \rightarrow \neg b,1)\},\neg b\rangle$, *and* $\mathsf{Label}(N_6) = \langle\{(\neg a,1),(\neg a \rightarrow \neg b,1)\},\neg b\rangle$.
$\mathsf{Root}(T) = N_1$. $\mathsf{Nodes}(T) = \{N_1,N_2,N_3,N_4,N_5,N_6\}$.

Note that I will omit the name of the nodes in argument trees that appear henceforth in this thesis as doing so does not cause any confusion.

Each path from the root node to another node in an argument tree is called an *argumentation line*.

Definition 3.5.2 *Let Ψ be a, possibly inconsistent, belief base and A_0 be an argument constructed from Ψ. An* **argumentation line** *for A_0 is a sequence of arguments from Ψ denoted $\Lambda = [A_0, A_1, A_2, \ldots]$ where each element of the sequence A_i, $0 < i$, is a defeater (proper or blocking) of its predecessor A_{i-1}.*

Note that every branch in an argument tree is an argumentation line.

Example 3.5.2 *Continuing the running example, the following are all examples of argumentation lines.*

$$\Lambda_1 = [a_5, a_2, a_1, a_2, a_1, a_2]$$

$$\Lambda_2 = [a_5, a_6, a_7]$$

$$\Lambda_3 = [a_5, a_6, a_8]$$

As each argument in an argumentation line $[A_0, A_1, A_2, \ldots]$ defeats its predecessor, A_1 defeats A_0 and so is an *interfering* argument for the claim of A_0. A_2 defeats A_1 and so is a *supporting* argument for the claim of A_0. If we carry on in this manner then we can split an argumentation line into two sets, supporting arguments for the claim of A_0 and interfering arguments for the claim of A_0.

Definition 3.5.3 *Let* $\Lambda = [A_0, A_1, A_2, \ldots]$ *be an argumentation line. The set of* **supporting** *arguments for* $\mathsf{Claim}(A_0)$ *is* $\Lambda^S = \{A_0, A_2, A_4, \ldots\}$*. The set of* **interfering** *arguments for* $\mathsf{Claim}(A_0)$ *is* $\Lambda^I = \{A_1, A_3, A_5, \ldots\}$.

Example 3.5.3 *Continuing the running example, the following are examples of sets of supporting/interfering arguments.*

$$\Lambda_1^S = \{a_5, a_1\} \quad \Lambda_1^I = \{a_2\}$$
$$\Lambda_2^S = \{a_5, a_7\} \quad \Lambda_2^I = \{a_6\}$$
$$\Lambda_3^S = \{a_5, a_8\} \quad \Lambda_3^I = \{a_6\}$$

As discussed in [22], there are certain undesirable situations in which we may end up with an infinite argumentation line. An example of this is circular argumentation, where an argument is used more than once in an argumentation line to defend itself. In order to avoid these undesirable situations some constraints must be imposed on what is an *acceptable* argumentation line. The interested reader is referred to [22] for a full discussion of such situations.

Before defining what an acceptable argumentation line is, I define what it means for a set of arguments to be concordant. A set of arguments is said to be concordant if and only if the union of the supports of all of the arguments is consistent.

Definition 3.5.4 *A set of arguments* $\{A_1, A_2, \ldots, A_n\}$ *is* **concordant** *iff* $\bigcup_{i=1}^{n} \mathsf{Support}(A_i) \nvdash \bot$.

I now define the four constraints that García and Simari place on an acceptable argumentation line.

Definition 3.5.5 *Let* $\Lambda = [A_0, A_1, A_2, \ldots]$ *be an argumentation line.* Λ *is an* **acceptable argumentation line** *iff*

1. *Λ is a finite sequence,*
2. *the set Λ_S of supporting arguments is concordant, and the set Λ_I of interfering arguments is concordant,*
3. *no argument A_k appearing in Λ, $0 \leq k \leq n$, is a sub-argument of an argument A_j that appears earlier in Λ_i, $j < k$,*
4. *for all i, such that the argument A_i is a blocking defeater for A_{i-1}, if A_{i+1}exists, then A_{i+1} is a proper defeater for A_i.*

The first constraint is not surprising, clearly we cannot easily deal with infinite paths in a tree. García and Simari include the second constraint as they believe, intuitively, that there should be agreement among the set of supporting arguments and among the set of interfering arguments. The third constraint is included to ensure that circular argumentation does not occur. The fourth constraint is included by García and Simari as they wish to avoid the situation in which, given a set of arguments that are either for claim α or claim $\neg\alpha$, if they all have the same preference level and there are more arguments for α than for $\neg\alpha$, then α is preferred to $\neg\alpha$. For a more thourough discussion of these constraints please refer to [22, pages 113–116].

Example 3.5.4 *Continuing the running example, we see that only Λ_2 is an acceptable argumentation line. Λ_1 is not acceptable as it breaks constraints (3) and (4), Λ_3 is not acceptable as it breaks constraint (2).*

In the next section I describe García and Simari's mechanism for determining whether, given a set of possibly inconsistent beliefs, an argument should be considered warranted or not.

3.6 Warrant through dialectical analysis

Given a set of beliefs Ψ and an argument $A_1 \in \mathcal{A}(\Psi)$, in order to know whether A_1 is defeated or not, an agent has to consider each argument from $\mathcal{A}(\Psi)$ that attacks A_1 and decide whether or not it defeats it. However, a defeater of A_1 may itself be defeated by another argument $A_2 \in \mathcal{A}(\Psi)$. Defeaters may also exist for A_2, themselves of which may also have defeaters. Therefore, in order to decide whether A_1 is defeated, an agent has to consider all defeaters for A_1, all of the defeaters of those defeaters, and so on. It does so by constructing a special type of argument tree called a dialectical tree, where all acceptable argumentation lines are represented in the tree.

Definition 3.6.1 *Let Ψ be a, possibly inconsistent, belief base and A_0 be an argument such that $A_0 \in \mathcal{A}(\Psi)$. A* **dialectical tree** *for A_0 constructed from Ψ, denoted $\mathsf{T}^{\Psi}_{A_0}$, is a special type of argument tree that is defined as follows.*

1. *The root of the tree is labelled with A_0.*
2. *Let N be a node of the tree labelled A_n, $n \geq 0$, and let $\Lambda_i = [A_0, \ldots, A_n]$ be the sequence of labels on the path from the root to node N. Let arguments $B_1, B_2, \ldots, B_k$ be all the defeaters for A_n that can be formed from Ψ.*
   For each defeater B_j, $1 \leq j \leq k$, such that the argumentation line $\Lambda'_i = [A_0, \ldots, A_n, B_j]$ is an acceptable argumentation line, then the node N has a child N_j that is labelled B_j.
   If there is no defeater for A_n or there is no B_j such that Λ'_i is acceptable, then N is a leaf node.

Example 3.6.1 *Continuing the running example, the dialectical tree constructed by agent x_1 with the argument a_5 at its root, $\mathsf{T}^{\Sigma^{x_1}}_{a_5}$, is shown in Figure 3.2.*

Note that the example argument tree shown in Figure 3.1 is not an example of a dialectical tree as it contains two unacceptable argumentation lines. The path that leads to $\langle\{(\neg a, 1)\}, \neg a\rangle$ breaks constraints 3 and 4 of the definition of an acceptable argumentation line (Definition 3.5.5) as the subargument $\langle\{(a, 1)\}, a\rangle$ appears twice in the argumentation line and we have a blocking defeater followed by another

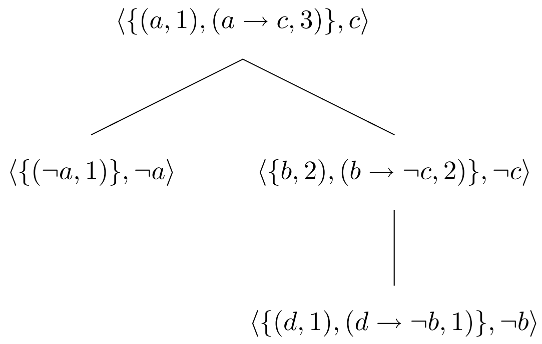


Figure 3.2: A dialectical tree.

blocking defeater. The path that leads to $\langle\{(\neg a,1),(\neg a \to \neg b,1)\},\neg b\rangle$ breaks constraint 2 of the definition of an acceptable argumentation line as the set of supporting arguments is not concordant.

I now define what it means for two dialectical trees to be equal to each other. This definition is used later in the analysis of this system to compare the trees generated by different warrant inquiry dialogues.

Definition 3.6.2 *Let* T^{Δ}_{A} *and* $\mathsf{T}^{\Delta'}_{A'}$ *be two dialectical trees.* T^{Δ}_{A} **equals** $\mathsf{T}^{\Delta'}_{A'}$*, denoted* $\mathsf{T}^{\Delta}_{A} = \mathsf{T}^{\Delta'}_{A'}$*, iff*

1. $\mathsf{Support}(A) = \mathsf{Support}(A')$,
2. $\mathsf{Claim}(A) = \mathsf{Claim}(A')$,
3. *the argumentation line* $[A, A_1, \ldots, A_n]$ *appears in* T^{Δ}_{A} *iff the argumentation line* $[A', A_1, \ldots, A_n]$ *appears in* $\mathsf{T}^{\Delta'}_{A'}$.

Note that a sub-tree of a dialectical tree (i.e. a node with all its descendants) is not always a dialectical tree [22, page 118]. This is because of the constraints on an acceptable argumentation line. Suppose an agent constructs an acceptable argumentation line and the defeater A cannot be included in this line as it would make the line unacceptable. There may be a sub-sequence of this line, however, where A could be included without violating any of the acceptable argumentation line constraints.

Following from [22], in order to determine whether the root of a dialectical tree is undefeated or not, we have to recursively mark each node in the tree as D (defeated) or U (undefeated).

Definition 3.6.3 *Let* T^{Ψ}_{A} *be a dialectical tree. The corresponding* **marked dialectical tree** *of* T^{Ψ}_{A} *is obtained by marking every node in* T^{Ψ}_{A} *as follows.*

1. *All leaves in* T^{Ψ}_{A} *are marked* U.
2. *If* B *is a node of* T^{Ψ}_{A} *and* B *is not a leaf node, then* B *will be marked* U *iff every child of* B *is marked* D*. The node* B *will be marked* D *iff it has at least one child marked* U.

Example 3.6.2 *Following the running example, the corresponding marked dialectical tree of* $\mathsf{T}^{\Sigma^x}_{a_5}$ *is shown in Figure 3.3.*

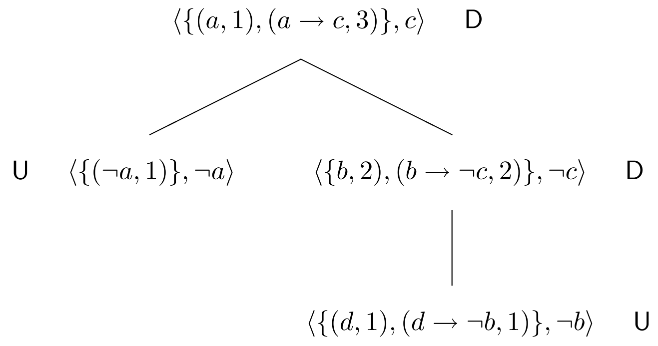


Figure 3.3: A marked dialectical tree.

The function Status takes a node of a dialectical tree and returns either D or U, depending on what the node is marked with in the marked version of the dialectical tree. If the node does not appear in the dialectical tree then this is indicated by the function Status returning $null$.

Definition 3.6.4 *The* **status** *of a node N in a dialectical tree, T_A^Ψ, is returned by the function* Status *such that* $\mathsf{Status}(N, \mathsf{T}_A^\Psi) = \mathsf{U}$ *iff N is marked* U *in the corresponding marked dialectical tree of* T_A^Ψ, *and* $\mathsf{Status}(N, \mathsf{T}_A^\Psi) = \mathsf{D}$ *iff N is marked with* D *in the corresponding marked dialectical tree of* T_A^Ψ, *else* $\mathsf{Status}(N, \mathsf{T}_A^\Psi) = null$.

The claim of an argument is warranted by the belief base if and only if the status of the root of the associated dialectical tree is U. We can see from Figure 3.3 that the claim of argument $\langle\{(a, 1), (a \rightarrow c, 3)\}, c\rangle$ is *not* warranted, as the status of the root of the associated dialectical tree is D.

Definition 3.6.5 *Let A be an argument such that $A \in \mathcal{A}(\Psi)$. We say that the claim of argument A is* **warranted by** Ψ *iff* $\mathsf{Status}(\mathsf{Root}(\mathsf{T}_A^\Psi), \mathsf{T}_A^\Psi) = \mathsf{U}$.

Example 3.6.3 *Continuing the running example, $\langle\{(a, 1), (a \rightarrow c, 3)\}, c\rangle$ is not warranted by Σ^{x_1}.*

The warrant procedure makes it possible for an agent to reason with incomplete, inconsistent, and uncertain knowledge—three of the defining characteristics of knowledge in the medical domain. Another characteristic of medical knowledge is that it is constantly changing. The defeasible nature of the rules makes it possible to deal efficiently with changes in knowledge, by adding new defeasible rules to the knowledge base.

García and Simari also point out that the notions of acceptable argumentation line and the dialectical tree provide a flexible structure for defining different argumentation protocols [22, page 119]. That is to say, this structure allows us to easily consider different strategies for accepting defeaters during argumentation.

3.7 Summary

I have presented García and Simari's argumentation system for internal reasoning in this chapter. I have made two changes to their system. Firstly, I made a trivial generalisation in assuming that their set of strict facts and set of strict rules are always empty. Secondly, I have introduced defeasible facts, which can be thought of as defeasible rules with an empty body. García and Simari state that defeasible rules have a non-empty body, but this is an arbitrary decision that reflects the fact that they do not desire the existence of defeasible facts. I have relaxed this constraint, as I want all knowledge to be defeasible. Neither of these changes have any effect on García and Simari's framework or their results.

In the next chapter, I go on to present my dialogue system, that allows us to apply García and Simari's warrant procedure to inter-agent reasoning between two agents.

Chapter 4

Dialogue system for argumentation between two agents

In this chapter I formally define a novel dialogue system. This system is capable of generating dialogues of different types. A dialogue of one type may be embedded in another dialogue of the same or different type. I define two particular types of dialogue, which I call argument inquiry and warrant inquiry. I provide a protocol for each of these dialogue types that returns the set of legal moves at a point in the dialogue. I also provide a strategy, for use by an agent with these protocols, that returns exactly one of the legal moves at a point in the dialogue. The move returned by the strategy is the move that gets made, hence this system is generative. I conclude this chapter with several dialogue examples.

In my opinion, both the argument inquiry and the warrant inquiry dialogue can be classified as inquiry dialogues according to the Walton and Krabbe typology [65]. Walton and Krabbe define an inquiry dialogue as follows [65, page 66].

Type	*Initial Situation*	*Main Goal*	*Participant's Aims*
Inquiry	General ignorance	Growth of knowledge and agreement	Find a 'proof' or destroy one

No formal definition of inquiry dialogues is given, leaving this classification somewhat open to interpretation. I believe that each of my argument inquiry and warrant inquiry dialogue types fit this general definition. In the warrant inquiry dialogue, the 'proof' takes the form of a dialectical tree, which has an argument at its root whose claim is the topic of the dialogue. In the argument inquiry dialogue, the 'proof' takes the form of an argument whose claim is the topic of the dialogue.

I have chosen to focus on inquiry dialogues for two reasons. Firstly, it is a dialogue type that has received little attention to date in the literature. The only groups that I am aware of who have defined inquiry protocols are Amgoud *et al* [3, 2, 46, 47, 48] and McBurney and Parsons [39, 40]. However, the main reason that I am focussing on inquiry dialogues is because this is a key dialogue type for use in a cooperative domain such as breast cancer care. It is common for doctors to need to share their knowledge in order to jointly make a decision, for example when diagnosing a condition whose symptoms span more than one medical speciality. These dialogue types are also sufficient for the referral agent scenario described in Section 1.2.1.

The warrant inquiry dialogue allows two agents to share parts of their knowledge in order to jointly decide whether a certain argument with a certain claim is warranted. It effectively allows us to take the dialectical reasoning mechanism given by García and Simari [22], which is intended for internal reasoning by one agent, and to use this mechanism for inter-agent reasoning between two agents.

The topic of a warrant inquiry dialogue is a literal. One agent will open a warrant inquiry dialogue about a certain topic and then both agents will share knowledge relevant to this topic. They exchange arguments that are used to construct a dialectical tree that has an argument for the topic at the root. In order to do this, the agents frequently need to share beliefs in order to jointly find arguments for a particular claim, and to do so they enter into an embedded argument inquiry dialogue that has that claim as its topic. The argument inquiry dialogue allows two agents to share parts of their knowledge, with the specific purpose of entailing new information. It is particularly useful in cooperative domains, where agents can be trusted to give truthful information and are willing to enter into such a dialogue.

A key feature of this work that sets it apart from much of the existing literature is the agents' capability to automatically generate dialogues. Most existing work is intended only to model dialogues (e.g. [48, 56]), that is to say they provide a protocol that tells us what moves are legal at a particular point, but not which in particular of these legal moves an agent should make. My framework provides intelligent strategies, for use with different dialogue types, that allow an agent to select exactly one of the set of legal moves at any point in a dialogue. Hence, my system allows agents to automatically generate dialogues. As far as I am aware, mine is the only example of a system that incorporates one or more intelligent strategies capable of generating inquiry dialogues. The set of moves used in a dialogue is defined in the next section.

4.1 The moves

The communicative acts in a dialogue are called moves. My move structure follows the normal conventions of agent communication languages in that it follows Austin and Searle's speech act theory [4, 62]. Austin and Searle propose that a speech act can be classified by its illocutionary force, that is the type of effect that the speaker hopes the speech act will have. For example, a speech act may be a request, where the speaker wishes the hearer to carry out an action, or an assertion, where the speaker wishes the hearer to be aware that the speaker believes a certain thing.

I assume that there are always exactly two agents, called participants, taking part in a dialogue, each with its own identifier taken from the set $\mathcal{I}$. Each participant takes it in turn to make a move to the other participant. For a dialogue involving participants $x_1, x_2 \in \mathcal{I}$, I also refer to participants using the meta-variables P and $\overline{P}$, such that if P is x_1 then $\overline{P}$ is x_2 and if P is x_2 then $\overline{P}$ is x_1.

A move in my system is a tuple of the form $\langle Agent, Act, Content \rangle$. $Agent$ is the identifier of the agent to which the move is addressed (the receiver of the move), Act is the type of move, and the $Content$ gives the details of the move. The format for moves used in warrant inquiry and argument inquiry dialogues is shown in Table 4.1. Note that the system allows for other types of dialogues to be

Move	**Format**
$open$	$\langle x, open, dialogue(\theta, \gamma)\rangle$
$assert$	$\langle x, assert, \langle \Phi, \phi \rangle\rangle$
$close$	$\langle x, close, dialogue(\theta, \gamma)\rangle$

Table 4.1: The format for moves used in dialogues, where x is an agent id ($x \in \mathcal{I}$), $\langle \Phi, \phi \rangle$ is an argument, and either $\theta = wi$ (for warrant inquiry) and $\gamma \in \mathcal{S}^\star$ (i.e. γ is a defeasible fact), or $\theta = ai$ (for argument inquiry) and $\gamma \in \mathcal{R}^\star$ (i.e. γ is a defeasible rule). The set of all moves meeting this format is denoted $\mathcal{M}$.

generated and these might require the addition of extra moves.

The open move is used to open a dialogue of a certain type with a certain topic. If an open move is made within a dialogue then it causes an embedded dialogue to be opened. Note that a warrant inquiry dialogue takes a defeasible fact as its topic, whilst an argument inquiry dialogue takes a defeasible rule as its topic. The assert move is used to exchange arguments. The close move is used to terminate a dialogue. Note that the two participants must be in agreement in order to terminate a dialogue, that is to say they must both make a close move one after the other.

The following function returns the receiver of a move.

Definition 4.1.1 *The* **receiver** *of a move* $\langle Agent, Act, Content\rangle$ *is returned by* $\mathsf{Receiver} : \mathcal{M} \mapsto \mathcal{I}$ *such that* $\mathsf{Receiver}(\langle Agent, Act, Content\rangle) = Agent$*.*

In the next section I give the general definition of a dialogue.

4.2 The general dialogue

A dialogue is simply a sequence of moves, each of which is made from one participant to the other. As a dialogue progresses over time, I denote each timepoint by a natural number. Each move is indexed by the timepoint when the move was made. Exactly one move is made at each timepoint. The dialogue itself is indexed with two timepoints, indexing the first and last moves of the dialogue. Although I am only considering argument inquiry and warrant inquiry type dialogues here, this is a general definition that is sufficient for dialogues of other types that one may want to specify.

Definition 4.2.1 *A* **dialogue***, denoted* D_r^t*, is a sequence of moves of the form* $[m_r, \ldots, m_t]$ *involving two participants* x_1 *and* x_2 *such that* $x_1, x_2 \in \mathcal{I}$*,* $x_1 \neq x_2$*,* $r, t \in \mathbb{N}$*,* $r \leq t$ *and the following conditions hold*

1) m_r *is a move of the form* $\langle P, open, dialogue(\theta, \gamma)\rangle$*,*
2) *for all* s *such that* $r \leq s \leq t$*,* $\mathsf{Receiver}(m_s) \in \{x_1, x_2\}$*,*
3) *for all* s *such that* $r \leq s < t$*,* $\mathsf{Receiver}(m_s) \neq \mathsf{Receiver}(m_{s+1})$*.*

The **topic** *of the dialogue is returned by the function* $\mathsf{Topic}(D_r^t)$ *such that* $\mathsf{Topic}(D_r^t) = \gamma$*. The* **type** *of the dialogue is returned by the function* Type *such that* $\mathsf{Type}(D_r^t) = \theta$*. The set of all dialogues is denoted* $\mathcal{D}$*.*

The first move of a dialogue D_r^t must always be an open move (condition (1) of the definition above), every move of the dialogue must be made to a participant of the dialogue (condition (2)), and the agents take it in turns to receive moves (condition (3)).

Example 4.2.1

1. Let $[m_1, m_2, m_3, m_4]$ *be a sequence of moves such that*

$$m_1 = \langle x_1, open, dialogue(\theta, \gamma)\rangle$$

$$m_2 = \langle x_2, assert, \langle \Phi, \phi\rangle\rangle$$

$$m_3 = \langle x_1, close, dialogue(\theta', \gamma')\rangle$$

$$m_4 = \langle x_2, assert, \langle \Phi, \phi\rangle\rangle$$

$[m_1, m_2, m_3, m_4]$ *is a dialogue according to the above definition.*

2. Let $[m_1, m_2, m_3]$ *be a sequence of moves such that*

$$m_1 = \langle x_1, open, dialogue(\theta, \gamma)\rangle$$

$$m_2 = \langle x_2, open, dialogue(\theta', \gamma')\rangle$$

$$m_3 = \langle x_2, assert, \langle \Phi, \phi\rangle\rangle$$

$[m_1, m_2, m_3]$ *is not a dialogue according to the above definition, as it breaks condition (3).*

3. Let $[m_3, m_4, m_5, m_6, m_7]$ *be a sequence of moves such that*

$$m_3 = \langle x_1, open, dialogue(\theta, \gamma)\rangle$$

$$m_4 = \langle x_2, open, dialogue(\theta, \gamma)\rangle$$

$$m_5 = \langle x_1, open, dialogue(\theta, \gamma)\rangle$$

$$m_6 = \langle x_2, open, dialogue(\theta, \gamma)\rangle$$

$$m_7 = \langle x_1, open, dialogue(\theta, \gamma)\rangle$$

$[m_3, m_4, m_5, m_6, m_7]$ *is a dialogue according to the above definition.*

4. Let $[m_1, m_2, m_3]$ *be a sequence of moves such that*

$$m_1 = \langle x_1, open, dialogue(\theta, \gamma)\rangle$$

$$m_2 = \langle x_2, close, dialogue(\theta, \gamma)\rangle$$

$$m_3 = \langle x_3, close, dialogue(\theta, \gamma)\rangle$$

$[m_1, m_2, m_3]$ *is not a dialogue according to the above definition, as it breaks condition (2).*

Note that if $r = 1$, then this indicates that the dialogue is a *top-level dialogue* (i.e. a dialogue that is not embedded within another dialogue). If $r \neq 1$ then this indicates that the dialogue must be embedded within one or more other dialogues.

Definition 4.2.2 *Let* D_r^t *be a dialogue.* D_r^t *is a* **top-level dialogue** *iff* $r = 1$. *The set of all top-level dialogues is denoted* $\mathcal{D}_{top}$.

I now define some extra terminology to allow us to talk about relationships between dialogues. The first of these is *extends*. The dialogue $D_r^{t_1}$ extends the dialogue D_r^t if and only if the sequence that is D_r^t appears at the beginning of the sequence that is $D_r^{t_1}$.

Definition 4.2.3 *Let D_r^t be a dialogue that is a sequence of n moves, $n = t - (r - 1)$. For any dialogue $D_r^{t_1}$, $D_r^{t_1}$* **extends** *D_r^t iff $t \leq t_1$ and the first n moves of $D_r^{t_1}$ are the sequence of moves that is D_r^t.*

Example 4.2.2 *Let $D_r^t = [m_r, m_{r+1}, m_t]$ be a dialogue (note $t = r + 2$).*
1. Let $D_{r'}^{t'} = [m_{r'}, m_{r'+1}, m_{r'+2}, m_{t'}]$ be a dialogue ($t' = r' + 3$) where $m_{r'} = m_r$, $m_{r'+1} = m_{r+1}$ and $m_{r'+2} = m_t$. $D_{r'}^{t'}$ extends D_r^t.
2. Let $D_{r'}^{t'} = [m_{r'}, m_{r'+1}, m_{t'}]$ be a dialogue ($t' = r' + 2$) where $m_{r'} = m_r$, $m_{r'+1} = m_{r+1}$ and $m_{t'} = m_t$. $D_{r'}^{t'}$ extends D_r^t.
3. Let $D_{r'}^{t'} = [m_{r'}, m_{r'+1}, m_{r'+2}, m_{r'+3}, m_{t'}]$ be a dialogue ($t' = r' + 4$) where $m_{r'+1} = m_r$, $m_{r'+2} = m_{r+1}$ and $m_{r'+3} = m_{r+2}$. $D_{r'}^{t'}$ does not extend D_r^t.

I now define an operator that appends a move to a dialogue sequence, to give us a dialogue *extension*.

Definition 4.2.4 *Let $D_r^t = [m_r, m_{r+1}, \ldots, m_t]$ be a dialogue. The* **extension** *of dialogue D_r^t by move m_{t+1} is denoted $D_r^t + m_{t+1}$ such that $D_r^t + m_{t+1} = [m_r, m_{r+1}, \ldots, m_t, m_{t+1}]$.*

Example 4.2.3 *Let $D_1^3 = [\langle x_1, open, dialogue(\theta, \gamma)\rangle, \langle x_2, assert, \langle \Phi, \phi\rangle\rangle, \langle x_1, assert, \langle \Phi', \phi'\rangle\rangle]$ be a dialogue. The extension of dialogue D_1^3 by move $m_4 = \langle x_2, close, dialogue(\theta, \gamma)\rangle$ is the dialogue $D_1^4 = [\langle x_1, open, dialogue(\theta, \gamma)\rangle, \langle x_2, assert, \langle \Phi, \phi\rangle\rangle, \langle x_1, assert, \langle \Phi', \phi'\rangle\rangle, \langle x_2, close, dialogue(\theta, \gamma)\rangle]$ (i.e. $D_1^3 + m_4 = D_1^4$).*

I also define the sub-dialogue relation. The dialogue $D_{r_1}^{t_1}$ is a sub-dialogue of the dialogue D_r^t if and only if the sequence that is $D_{r_1}^{t_1}$ is a sub-sequence of the sequence that is D_r^t. If a dialogue $D_{r_1}^{t_1}$ is a sub-dialogue of a dialogue D_r^t then this means that $D_{r_1}^{t_1}$ is embedded within D_r^t. A dialogue may be embedded within several other dialogues, creating a nesting of dialogues and this will be illustrated in a later example (Example 4.2.5).

Definition 4.2.5 *Let $D_r^t = [m_r, m_{r+1}, \ldots, m_{t-1}, m_t]$ be a dialogue that is a sequence of n moves, $n = t - (r - 1)$. Let $D_{r_1}^{t_1}$ be a dialogue that is a sequence of m moves, $m = t_1 - (r_1 - 1)$. $D_{r_1}^{t_1}$ is a* **sub-dialogue** *of D_r^t iff $r < r_1$, $t_1 \leq t$, and the sequence $D_{r_1}^{t_1}$ is a sub-sequence appearing in the sequence D_r^t.*

Example 4.2.4 *Let $D_r^t = [m_r, m_{r+1}, m_{r+2}, m_{r+3}, m_t]$ be a dialogue ($t = r + 4$).*
1. Let $D_{r'}^{t'} = [m_{r'}, m_{r'+1}, m_{t'}]$ be a dialogue ($t' = r' + 2$) where $m_{r'} = m_{r+1}$, $m_{r'+1} = m_{r+2}$ and $m_{t'} = m_{r+3}$. $D_{r'}^{t'}$ is a sub-dialogue of D_r^t.
2. Let $D_{r'}^{t'} = [m_{r'}, m_{r'+1}, m_{t'}]$ be a dialogue ($t' = r' + 2$) where $m_{r'} = m_r$, $m_{r'+1} = m_{r+1}$ and $m_{t'} = m_{r+2}$. $D_{r'}^{t'}$ is not a sub-dialogue of D_r^t.
3. Let $D_{r'}^{t'} = [m_{r'}, m_{r'+1}, m_{r'+2}, m_{t'}]$ be a dialogue ($t' = r' + 3$) where $m_{r'} = m_{r+1}$, $m_{r'+1} = m_{r+2}$, $m_{r'+1} = m_{r+3}$ and $m_{t'} = m_t$. $D_{r'}^{t'}$ is a sub-dialogue of D_r^t.

4. Let $D^{t'}_{r'} = [m_{r'}, m_{r'+1}, m_{t'}]$ be a dialogue ($t' = r' + 2$) where $m_{r'} = m_{r+3}$ and $m_{r'+1} = m_t$. $D^{t'}_{r'}$ is not a sub-dialogue of D^t_r.

A top-dialogue of a dialogue is any top-level dialogue that it appears in. If a dialogue is not a top-level dialogue then this means that it must be embedded within one or more other dialogues. A top-dialogue of a dialogue D^t_r is the outermost dialogue in which D^t_r is embedded. (Note that the following four definitions will be illustrated shortly in Example 4.2.5).

Definition 4.2.6 *Let D^t_1 be a top-level dialogue. D^t_1 is a* **top-dialogue** *of the dialogue D^t_r iff the participants of D^t_1 are the same as the participants of D^t_r, and either the sequence that is D^t_1 is the same as the sequence that is D^t_r or D^t_r is a sub-dialogue of D^t_1.*

In order to terminate a dialogue, two close moves must appear next to each other in the sequence, called a matched-close.

Definition 4.2.7 *Let D^t_r be a dialogue with participants x_1 and x_2 such that $\mathsf{Topic}(D^t_r) = \gamma$ and $\mathsf{Type}(D^t_r) = \theta$. We say that m_s, $r < s \leq t$, is a* **matched-close for** *D^t_r iff $m_{s-1} = \langle P, close, dialogue(\theta, \gamma)\rangle$ and $m_s = \langle \overline{P}, close, dialogue(\theta, \gamma)\rangle$.*

So a matched-close will terminate a dialogue D^t_r (condition (1) of the following definition), but only if D^t_r has not already terminated (condition (2) of the following definition) and any sub-dialogues that are embedded within D^t_r are also terminated (condition (3) of the following definition). This ensures that a dialogue does not terminate after an inner embedded dialogue has been opened but not yet terminated.

Definition 4.2.8 *Let D^t_r be a dialogue. D^t_r* **terminates at** *t iff the following conditions hold:*

1. m_t is a matched-close for D^t_r,

2. there does not exist $D^{t_1}_r$ such that $D^{t_1}_r$ terminates at t_1 and D^t_r extends $D^{t_1}_r$,

3. for all $D^{t_1}_{r_1}$, if $D^{t_1}_{r_1}$ is a sub-dialogue of D^t_r,
then there exists $D^{t_2}_{r_1}$ such that $D^{t_2}_{r_1}$ terminates at t_2
and [either $D^{t_2}_{r_1}$ extends $D^{t_1}_{r_1}$ or $D^{t_1}_{r_1}$ extends $D^{t_2}_{r_1}$]
and $D^{t_2}_{r_1}$ is a sub-dialogue of D^t_r.

As we are often dealing with multiple nested dialogues, it is sometimes useful to refer to the current dialogue, which is the innermost dialogue that has not yet terminated. The first condition of the following definition states that the current dialogue must start with an open move (necessary for the current dialogue to be a dialogue). The second condition states that all sub-dialogues of the current dialogue have terminated, and the third condition states that the current dialogue has not yet terminated.

Definition 4.2.9 *Let D^t_r be a dialogue. The* **current dialogue** *is returned by $\mathsf{Current}(D^t_r)$ such that $\mathsf{Current}(D^t_r) = D^t_{r_1}$ where $1 \leq r \leq r_1 \leq t$ and the following conditions hold:*

1. $m_{r_1} = \langle x, open, dialogue(\theta, \gamma)\rangle$ for some $x \in \mathcal{I}$, some $\gamma \in \mathcal{B}$ and $\theta \in \{wi, ai\}$,

2. *for all* $D^{t_1}_{r_2}$, *if* $D^{t_1}_{r_2}$ *is a sub-dialogue of* $D^t_{r_1}$,

   then there exists $D^{t_2}_{r_2}$ *such that either* $D^{t_2}_{r_2}$ *extends* $D^{t_1}_{r_2}$ *or* $D^{t_1}_{r_2}$ *extends* $D^{t_2}_{r_2}$

   and $D^{t_2}_{r_2}$ *is a sub-dialogue of* $D^t_{r_1}$

   and $D^{t_2}_{r_2}$ *terminates at* t_2,

3. *there does not exist* $D^{t_3}_{r_1}$ *such that* $D^t_{r_1}$ *extends* $D^{t_3}_{r_1}$ *and* $D^{t_3}_{r_1}$ *terminates at* t_3.

If the above conditions do not hold then $\mathsf{Current}(D^t_r) = null$.
The **topic of the current dialogue** *is returned by* cTopic *such that* $\mathsf{cTopic}(D^t_r) = \mathsf{Topic}(\mathsf{Current}(D^t_r))$.
The **type of the current dialogue** *is returned by* cType *such that* $\mathsf{cType}(D^t_r) = \mathsf{Type}(\mathsf{Current}(D^t_r))$.
Note, $\mathsf{Topic}(null) = null$ *and* $\mathsf{Type}(null) = null$.

I now give a schematic example of nested dialogues.

Example 4.2.5 *An example of nested dialogues is shown in Figure 4.1. In this example,* D^t_1 *is a top-level dialogue that has not yet terminated.* D^t_i *is a sub-dialogue of* D^t_1 *that terminates at t.* D^k_j *is a sub-dialogue of both* D^t_1 *and* D^t_i*, that terminates at k.*
D^t_1 *is a top-dialogue of* D^t_1. D^k_1 *is a top-dialogue of* D^k_j. D^t_1 *is a top-dialogue of* D^t_i. D^k_1 *is a top-dialogue of* D^k_1.
$\mathsf{Current}(D^t_1) = D^t_1$. $\mathsf{Current}(D^{t-1}_1) = D^{t-1}_i$. $\mathsf{Current}(D^k_1) = D^k_i$. $\mathsf{Current}(D^{k-1}_1) = D^{k-1}_j$. $\mathsf{Current}(D^t_i) = null$. $\mathsf{Current}(D^{t-1}_i) = D^{t-1}_i$. $\mathsf{Current}(D^k_i) = D^k_i$. $\mathsf{Current}(D^{k-1}_i) = D^{k-1}_j$. $\mathsf{Current}(D^k_j) = null$. $\mathsf{Current}(D^{k-1}_j) = D^{k-1}_j$.

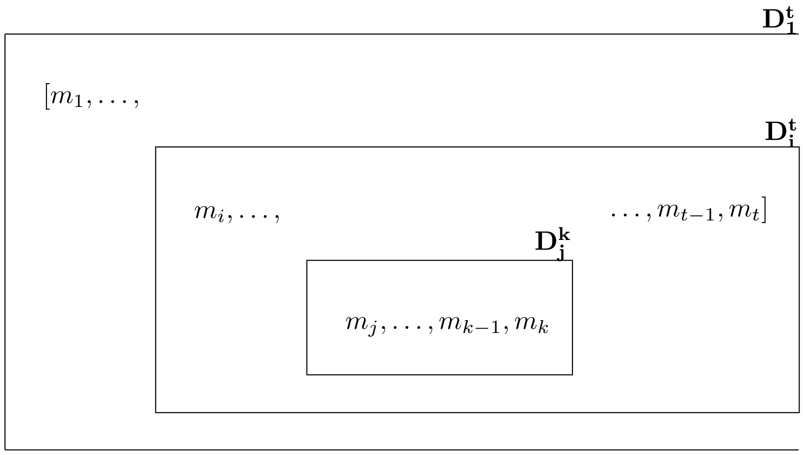
I adopt the standard approach of associating a *commitment store* with each agent participating in a dialogue. A commitment store is a set of beliefs that the agent has asserted so far in the course of the dialogue. As a commitment store consists of things that the agent has already publicly declared, its contents are visible to the other agent participating in the dialogue. For this reason, when constructing an argument, an agent may make use of not only its own beliefs, but also those from the other agent's commitment store.

Definition 4.2.10 *A* **commitment store** *associated with an agent* x *at a timepoint t, denoted* CS^t_x*, where* $x \in \mathcal{I}$ *and* $t \in \mathbb{N}$*, is a set of beliefs,* $CS^t_x \subseteq \mathcal{B}$.

An agent's commitment store grows monotonically over time. An agent's commitment store is updated when the agent makes an assert move. If an agent makes a move asserting an argument, every element of the support is added to the agent's commitment store. This is the only time the commitment store is updated.

Definition 4.2.11 (Commitment store update) *Let* D^t_r *be the current dialogue such that* $\mathsf{Receiver}(m_t) = \overline{P}$.

$$CS^t_P = \begin{cases} \emptyset & \text{iff } t = 0, \\ CS^{t-1}_P \cup \Phi & \text{iff } m_t = \langle \overline{P}, assert, \langle \Phi, \phi \rangle \rangle, \\ CS^{t-1}_P & \text{otherwise.} \end{cases}$$



$$1 < i < j < k < t - 1$$

$$D_1^t = [m_1, \ldots, m_i, \ldots, m_j, \ldots, m_k, m_{k+1}, \ldots, m_{t-1}, m_t]$$

$$m_1 = \langle P_1, open, dialogue(\theta_1, \phi_1)\rangle, m_i = \langle P_i, open, dialogue(\theta_i, \phi_i)\rangle$$

$$m_j = \langle P_j, open, dialogue(\theta_j, \phi_j)\rangle, m_{k-1} = \langle P_{k-1}, close, dialogue(\theta_k, \phi_k)\rangle$$

$$m_k = \langle P_k, close, dialogue(\theta_k, \phi_k)\rangle, m_{t-1} = \langle P_{t-1}, close, dialogue(\theta_t, \phi_t)\rangle$$

$$m_t = \langle P_t, close, dialogue(\theta_t, \phi_t)\rangle$$

Figure 4.1: Nested dialogues. D_1^t is a top level dialogue that has not yet terminated. D_i^t is a sub-dialogue of D_1^t that terminates at t. D_j^k is a sub-dialogue of both D_1^t and D_i^t, that terminates at k.

Example 4.2.6 *Let D_1^7 be a dialogue that terminates at 7 with participants x_1 and x_2 such that*

$$m_1 = \langle x_2, open, dialogue(\theta, \gamma)\rangle$$
$$m_2 = \langle x_1, assert, \langle\{(a, 2), (a \rightarrow b, 1)\}, b\rangle\rangle$$
$$m_3 = \langle x_2, assert, \langle\{(c, 1), (c \rightarrow \neg a, 1)\}, \neg a\rangle\rangle$$
$$m_4 = \langle x_1, close, dialogue(\theta, \gamma)\rangle$$
$$m_5 = \langle x_2, assert, \langle\{(d, 1), (d \rightarrow \neg b, 2)\}, \neg b\rangle\rangle$$
$$m_6 = \langle x_1, close, dialogue(\theta, \gamma)\rangle$$
$$m_7 = \langle x_2, close, dialogue(\theta, \gamma)\rangle$$

Recall that the first element of a move denotes the agent that is receiving the move, hence agent x_1 makes the first move, agent x_2 makes the second move and so on. The commitment stores for each time point in the dialogue are as follows

$$CS^1_{x_1} = \emptyset \qquad CS^1_{x_2} = \emptyset$$
$$CS^2_{x_1} = \emptyset \qquad CS^2_{x_2} = \{(a, 2), (a \rightarrow b, 1)\}$$
$$CS^3_{x_1} = \{(c, 1), (c \rightarrow \neg a, 1)\} \qquad CS^3_{x_2} = \{(a, 2), (a \rightarrow b, 1)\}$$
$$CS^4_{x_1} = \{(c, 1), (c \rightarrow \neg a, 1)\} \qquad CS^4_{x_2} = \{(a, 2), (a \rightarrow b, 1)\}$$
$$CS^5_{x_1} = \{(c, 1), (c \rightarrow \neg a, 1), (d, 1), (d \rightarrow \neg b, 2)\} \qquad CS^5_{x_2} = \{(a, 2), (a \rightarrow b, 1)\}$$
$$CS^6_{x_1} = \{(c, 1), (c \rightarrow \neg a, 1), (d, 1), (d \rightarrow \neg b, 2)\} \qquad CS^6_{x_2} = \{(a, 2), (a \rightarrow b, 1)\}$$
$$CS^7_{x_1} = \{(c, 1), (c \rightarrow \neg a, 1), (d, 1), (d \rightarrow \neg b, 2)\} \qquad CS^7_{x_2} = \{(a, 2), (a \rightarrow b, 1)\}$$

Each dialogue type has a specific protocol associated with it. A protocol is a function that returns the set of moves that are legal for an agent to make at a particular point in a particular type of dialogue. In the following sections, I will give specific protocols for argument inquiry and warrant inquiry dialogues, but the definition given here is a general one. A protocol function takes the top-level dialogue that two agents are participating in and the identifier of the agent whose turn it is to move, and returns the set of legal moves that the agent may make.

Definition 4.2.12 *A* **protocol**, Π_θ, *is a function such that* $\theta \in \{ai, wi\}$ *and* $\Pi_\theta : \mathcal{D}_{top} \times \mathcal{I} \mapsto \wp(\mathcal{M})$.

Note that in order for it to be easily possible to check an agent's conformance with a protocol, the protocol should only refer to public elements of the dialogue and not, for example, to an agent's private beliefs.

I believe that it is important to consider properties such as soundness and completeness if we are to understand the behaviour of dialogues. In order to consider such properties, I must first define what the outcome of a dialogue is and then later define a benchmark to compare such dialogue outcomes to (Section 5.5). The outcome function returns the outcome at any particular point in a particular type of dialogue. Specific outcome functions for different dialogue types will come later.

Definition 4.2.13 *The* **outcome** *of a dialogue,* $\mathsf{Outcome}_\theta$, *is a function such that* $\theta \in \{wi, ai\}$ *and* $\mathsf{Outcome}_\theta : \mathcal{D} \mapsto \wp(\mathcal{A}(\mathcal{B}))$.

Along with the definition of a specific strategy function (that will come later in Section 4.5), the specification of specific outcome functions for different dialogue types (that will also come later in Sections 4.3 and 4.4) sets this theory apart from most other similar systems. Most do not define the outcome of a dialogue, with a notable exception being [47], making it hard to analyse the behaviour of such systems.

This completes the general definition of my dialogue framework. In the next sections I will give the details that specify the argument inquiry and warrant inquiry dialogue types.

4.3 The argument inquiry dialogue

The goal of an argument inquiry dialogue is to jointly construct an argument for a particular claim. It is easy to imagine an example of a situation in which neither of the two participants could construct a particular argument independently, but if they cooperated and pooled their knowledge then they would be able to do so. For example, when called in to help diagnose a patient, a consultant may have the specialist knowledge about which combinations of symptoms suggest which diseases but, having not yet examined the patient, may not know what the patient's particular symptoms are. A nurse working on the same ward as the patient will not have the specialist knowledge that the consultant has, but will know all about the patient's symptoms. The nurse and the specialist could then enter into an argument inquiry dialogue to try and jointly construct arguments for different diagnoses.

An argument inquiry dialogue has a set associated with it called a *question store*. This set is used to keep track of literals that, if known to be true, would allow an argument for the consequent of the topic of the dialogue to be constructed. An argument inquiry dialogue is initiated when an agent wants to construct an argument for a certain claim, let us say ϕ, that it cannot do so alone. If the agent knows of a domain belief whose consequent is that claim, let us say $(\alpha_1 \wedge \ldots \wedge \alpha_n \rightarrow \phi, L)$, then the agent will open an argument inquiry dialogue with $\alpha_1 \wedge \ldots \wedge \alpha_n \rightarrow \phi$ as its topic. If, between them, the two participating agents could provide arguments for each of the elements α_i, $1 \leq i \leq n$, in the antecedent of the topic, then it would be possible for an argument for ϕ to be constructed. The question store is used to keep track of these elements.

When an argument inquiry dialogue is opened with topic $\alpha_1 \wedge \ldots \wedge \alpha_n \rightarrow \phi$, a question store associated with that dialogue is created whose content is $\{\alpha_1, \ldots, \alpha_n, \phi\}$. Throughout the dialogue, the participating agents will both try and provide arguments for the elements in the question store. This may lead them to open further nested argument inquiry dialogues that have as a topic a rule whose consequent is an element in the question store.

Definition 4.3.1 *For a dialogue D_r^t with participants x_1 and x_2, a* **question store***, denoted QS_r, is a finite set of literals such that*

$$QS_r = \begin{cases} \{\alpha_1, \ldots, \alpha_n, \beta\} & \textit{iff } m_r = \langle P, open, dialogue(ai, \alpha_1 \wedge \ldots \wedge \alpha_n \rightarrow \beta)\rangle \ (1 \leq n), \\ \emptyset & \textit{otherwise.} \end{cases}$$

The **question store of the current dialogue** *is returned by* cQS *such that* $\mathsf{cQS}(D_r^t) = QS_{r_1}$ *iff* $\mathsf{Current}(D_r^t) = D_{r_1}^t$.

I will now give an example of an argument inquiry dialogue illustrating the intuition behind the protocol which I will define shortly (Definition 4.3.2).

Example 4.3.1 *Let D_1^{11} be an argument inquiry dialogue that terminates at 11 with participants x_1 and x_2 such that*

$$\Sigma^{x_1} = \{(a \wedge b \rightarrow c, 1), (e \rightarrow b, 2)\}$$

$$\Sigma^{x_2} = \{(d, 1), (e, 1), (d \rightarrow a, 1)\}$$

Agent x_1 wishes to construct an argument for c and believes that agent x_2 may be able to help.

$$m_1 = \langle x_2, open, dialogue(ai, a \wedge b \rightarrow c)\rangle$$
$$m_2 = \langle x_1, assert, \langle\{(d, 1), (d \rightarrow a, 1)\}, a\rangle\rangle$$
$$m_3 = \langle x_2, open, dialogue(ai, e \rightarrow b)\rangle$$
$$m_4 = \langle x_1, assert, \langle\{(e, 1)\}, e\rangle\rangle$$
$$m_5 = \langle x_2, assert, \langle\{(e, 1), (e \rightarrow b, 2)\}, b\rangle\rangle$$
$$m_6 = \langle x_1, close, dialogue(ai, e \rightarrow b)\rangle$$
$$m_7 = \langle x_2, close, dialogue(ai, e \rightarrow b)\rangle$$
$$m_8 = \langle x_1, close, dialogue(ai, a \wedge b \rightarrow c)\rangle$$
$$m_9 = \langle x_2, assert, \langle\{(e, 1), (e \rightarrow b, 2), (d, 1), (d \rightarrow a, 1), (a \wedge b \rightarrow c, 1)\}, c\rangle\rangle$$
$$m_{10} = \langle x_1, close, dialogue(ai, a \wedge b \rightarrow c)\rangle$$
$$m_{11} = \langle x_2, close, dialogue(ai, a \wedge b \rightarrow c)\rangle$$

Timepoint 1. *When agent x_1 makes the move m_1 it opens the argument inquiry dialogue D_1^{11} that terminates at 11. Agent x_1 wishes to collaborate with agent x_2 to try and construct an argument for c, and it has a belief in the defeasible rule $a \wedge b \rightarrow c$ that may help in the construction of an argument for c. The question store associated with D_1^{11} is $QS_1 = \{a, b, c\}$. If x_1 and x_2 can jointly construct arguments for a and b then x_1 will be able to use these along with its belief in $a \wedge b \rightarrow c$ to construct an argument for c. The element c is also included in the question store as it may be the case that x_1 and x_2 are not able to jointly construct arguments for x_1 and x_2, but one of them may know of a different defeasible rule that may allow them to jointly construct an argument for c (e.g. $f \wedge g \rightarrow c$). Both commitment stores are empty, $CS_{x_1}^1 = CS_{x_2}^1 = \emptyset$.*

Timepoint 2. *When making the move m_2, x_2 examines the question store for the current argument inquiry dialogue ($QS_1 = \{a, b, c\}$) and sees if it can provide an argument for any of the elements in it. It finds that it can provide an argument for a and so asserts this argument. This causes the support of this argument to be added to x_2's commitment store, $CS_{x_2}^2 = \{(d, 1), (d \rightarrow a, 1)\}$. x_1's commitment store is still empty, $CS_{x_1}^2 = \emptyset$.*

Timepoint 3. *Agent x_1 now examines the current question store ($QS_1 = \{a, b, c\}$) and although it cannot provide an argument for any of its elements, it does know of a defeasible rule which may help to construct an argument for an element of the current question store, $e \rightarrow b$. Agent x_1 makes the move m_3, opening the embedded argument inquiry dialogue D_3^7 that terminates at 7. The question store associated with D_3^7 is $QS_3 = \{e, b\}$. No assert move has been made so the commitment stores remain the same,*

$CS^3_{x_1} = \emptyset$, $CS^3_{x_2} = \{(d, 1), (d \rightarrow a, 1)\}$.

Timepoint 4. *The current dialogue is D^4_3 and so agent x_2 examines the question store associated with this dialogue ($QS_3 = \{e, b\}$) and finds that it can provide an argument for one of its elements, e. Agent x_2 makes move m_4 asserting this argument, causing the support of this argument to be added to its public commitment store. $CS^4_{x_2} = \{(d, 1), (d \rightarrow a, 1), (e, 1)\}$. x_1's commitment store does not change, $CS^4_{x_1} = \emptyset$.*

Timepoint 5. *Agent x_1 is now able to use the elements just added to x_2's commitment store along with its belief in $e \rightarrow b$ to construct an argument for b (an element of the current question store, $QS_3 = \{e, b\}$), and so asserts this argument in move m_5. This causes the support of this argument to be added to x_1's commitment store, $CS^5_{x_1} = \{(e, 1), (e \rightarrow b, 2)\}$. x_2's commitment store does not change, $CS^5_{x_2} = \{(d, 1), (d \rightarrow a, 1), (e, 1)\}$.*

Timepoint 6. *Agent x_2 checks to see whether it is able to construct any arguments for an element of the current question store ($QS_3 = \{e, b\}$) that cannot already be constructed from the union of the two public commitment stores. It cannot and so makes the move m_6 to close the current dialogue. The commitment stores do not change, $CS^6_{x_1} = \{(e, 1), (e \rightarrow b, 2)\}$, $CS^6_{x_2} = \{(d, 1), (d \rightarrow a, 1), (e, 1)\}$.*

Timepoint 7. *Agent x_1 now checks to see whether it is able to construct any arguments for an element of the current question store ($QS_3 = \{e, b\}$) that cannot already be constructed from the union of the two public commitment stores. It cannot and so makes the move m_7 to close the current dialogue. As m_7 is a matched-close for D^7_3 and there are not any dialogues embedded within D^7_3 that have not yet terminated, D^7_3 terminates at 7. The commitment stores do not change, $CS^7_{x_1} = \{(e, 1), (e \rightarrow b, 2)\}$, $CS^7_{x_2} = \{(d, 1), (d \rightarrow a, 1), (e, 1)\}$.*

Timepoint 8. *As D^7_3 has terminated the current dialogue is now D^8_1, and so the current question store is $QS_1 = \{a, b, c\}$. Agent x_2 checks to see whether it can construct any arguments for an element of QS_1 that cannot already be constructed from the union of the two public commitment stores. It cannot and so makes the move m_8 to close the current dialogue. The commitment stores do not change, $CS^8_{x_1} = \{(e, 1), (e \rightarrow b, 2)\}$, $CS^8_{x_2} = \{(d, 1), (d \rightarrow a, 1), (e, 1)\}$.*

Timepoint 9. *Agent x_1 checks to see whether it can construct any arguments for an element of the current question store ($QS_1 = \{a, b, c\}$) that cannot already be constructed from the union of the two public commitment stores. It can use the elements added to the commitment stores at timepoints 4 and 5, along with its belief in $a \wedge b \rightarrow c$, to construct an argument for c and so asserts this argument in move m_9. This causes the support of the argument to be added to x_1's commitment store, $CS^9_{x_1} = \{(e, 1), (e \rightarrow b, 2), (d, 1), (d \rightarrow a, 1), (a \wedge b \rightarrow c, 1)\}$. x_1's commitment store does not change $CS^9_{x_2} = \{(d, 1), (d \rightarrow a, 1), (e, 1)\}$.*

Timepoint 10. *Agent x_1 checks to see whether it is able to construct any arguments for an element of the current question store ($QS_1 = \{a, b, c\}$) that cannot already be constructed from the union of the two public commitment stores. It cannot and so makes the move m_{10} to close the current dialogue. The commitment stores do not change, $CS^{10}_{x_1} = \{(e, 1), (e \rightarrow b, 2), (d, 1), (d \rightarrow a, 1), (a \wedge b \rightarrow c, 1)\}$, $CS^{10}_{x_2} = \{(d, 1), (d \rightarrow a, 1), (e, 1)\}$.*

Timepoint 11. *Agent x_1 now checks to see whether it is able to construct any arguments for an element of the current question store ($QS_1 = \{a, b, c\}$) that cannot already be constructed from the union of the two public commitment stores. It cannot and so makes the move m_{11} to close the current dialogue. As m_{11} is a matched-close for D_1^{11} and there are not any dialogues embedded within D_1^{11} that have not yet terminated, D_1^{11} terminates at 11. The commitment stores do not change, $CS_{x_1}^{11} = \{(e, 1), (e \rightarrow b, 2), (d, 1), (d \rightarrow a, 1), (a \wedge b \rightarrow c, 1)\}$, $CS_{x_2}^{11} = \{(d, 1), (d \rightarrow a, 1), (e, 1)\}$.*
The two agents have successfully jointly constructed an argument for c, $\langle\{(e, 1), (e \rightarrow b, 2), (d, 1), (d \rightarrow a, 1), (a \wedge b \rightarrow c, 1)\}, c\rangle$.

I now give the specific protocol function for argument inquiry dialogues. It takes the top-level dialogue that the agents are participating in and the identifier of the agent whose turn it is to move, and returns the set of legal moves that the agent may make. Note that it is straightforward to check conformance with the argument inquiry protocol, as it only refers to public elements of the dialogue. That is to say, it does not refer to either participating agent's beliefs.

Definition 4.3.2 *The* **argument inquiry protocol** *is a function $\Pi_{ai} : \mathcal{D}_{top} \times \mathcal{I} \mapsto \wp(\mathcal{M})$. If D_1^t is a top-level dialogue with participants x_1 and x_2,* $\mathsf{Receiver}(m_t) = P$, $1 \leq t$ *and* $\mathsf{cTopic}(D_1^t) = \gamma$, *then* $\Pi_{ai}(D_1^t, P)$ *is*

$$\Pi_{ai}^{assert}(D_1^t, P) \cup \Pi_{ai}^{open}(D_1^t, P) \cup \{\langle \overline{P}, close, dialogue(ai, \gamma)\rangle\}$$

where

$$\begin{aligned} \Pi_{ai}^{assert}(D_1^t, P) = \{\langle \overline{P}, assert, \langle \Phi, \phi\rangle\rangle \mid \\ &\textit{(1) } \phi \in \mathsf{cQS}(D_1^t), \\ &\textit{(2) } \Phi \not\subseteq CS_P^t \cup CS_{\overline{P}}^t\} \end{aligned}$$

$$\begin{aligned} \Pi_{ai}^{open}(D_1^t, P) = \{\langle \overline{P}, open, dialogue(ai, \beta_1 \wedge \ldots \wedge \beta_n \rightarrow \alpha)\rangle \mid \\ &\textit{(1) } \alpha \in \mathsf{cQS}(D_1^t), \\ &(2) \textit{ there does not exist } t' \textit{ such that } 1 < t' \leq t \\ &\quad \textit{and } QS_{t'} = \{\beta_1, \ldots, \beta_n, \alpha\}\} \end{aligned}$$

As previously remarked, the argument inquiry protocol refers only to public elements of the dialogue, this is deliberate to ensure that conformance with the protocol can be checked. The first condition of a legal assert move ($\phi \in \mathsf{cQS}(D_1^t)$) states that the claim of the argument being asserted must be present in the current question store. This helps to ensure the focus of the dialogue; the participating agents of an argument inquiry dialogue are trying to provide arguments for the elements in the question store, as if they do so then they will be able to form an argument with the desired claim. They should not waste time by asserting arguments that will not help to do this. The second condition of a legal assert move ($\Phi \not\subseteq CS_P^t \cup CS_{\overline{P}}^t$) states that the support of the argument being asserted is not already present in the union of the commitment stores. This ensures that assert moves are not repeated and also that agents do not waste time asserting arguments which can already be constructed by both participating agents.

The first condition of a legal open move ($\alpha \in \mathsf{cQS}(D_1^t)$) states that the consequent of the topic of the argument inquiry dialogue being opened with the move must be present in the current question store. Again, this is to maintain the focus of the dialogue. If the argument inquiry dialogue being opened is successful in finding an argument for the consequent of the topic, then this may help the agents to construct an argument for the consequent of the topic of the dialogue in which the argument inquiry dialogue being opened is embedded. The second condition of a legal open move (there does not exist t' such that $1 < t' \leq t$ and $QS_{t'} = \{\beta_1, \ldots, \beta_n, \alpha\}$) ensures that open moves are not repeated by an agent. It also ensures that an agent will not open a new argument inquiry dialogue with topic $\beta_1 \wedge \ldots \wedge \beta_n \rightarrow \alpha$ if an argument inquiry dialogue has previously been opened whose topic has an antecedent that is some permutation of $\beta_1 \wedge \ldots \wedge \beta_n$, e.g. if an argument inquiry dialogue with topic $\beta_1 \wedge \beta_2 \wedge \beta_3 \rightarrow \alpha$ had previously been opened then it would not be possible to open a new argument inquiry dialogue with $\beta_2 \wedge \beta_3 \wedge \beta_1 \rightarrow \alpha$ as its topic. This helps avoid redundancy, as two such argument inquiry dialogues would lead to the same outcome.

A well-formed argument inquiry dialogue is a dialogue whose first move is an open move that has $dialogue(ai, \phi)$ as its content where ϕ is a defeasible rule (condition (1) of the following definition). Condition (2) of the following definition ensures that there must be a continuation of the dialogue that terminates (ensuring that outer dialogues are not closed after an inner one has been opened but not yet closed), and that all the moves in the terminating continuation of the dialogue are legal according to the argument inquiry protocol.

Definition 4.3.3 *A* **well-formed argument inquiry dialogue** *is a dialogue of the form* $D_r^t = [m_r, \ldots, m_t]$ *with participants* x_1 *and* x_2 *such that*

1. $m_r = \langle P, open, dialogue(ai, \phi)\rangle$ *where* $P \in \{x_1, x_2\}$ *and* $\phi \in \mathcal{R}^\star$ *(i.e.* ϕ *is a defeasible rule),*

2. *there exists* t' *such that* $t \leq t'$, $D_r^{t'}$ *extends* D_r^t*, and* $D_r^{t'}$ *terminates at* t'*, and*
   for all s *such that* $r \leq s < t'$ *and* $D_r^{t'}$ *extends* D_r^s*, if* $D_1^{t'}$ *is a top-dialogue of* $D_r^{t'}$ *and* D_1^s *is a top-dialogue of* D_r^s *and* $D_1^{t'}$ *extends* D_1^s *and* $\mathsf{Receiver}(m_s) = P'$ *(where* $P' \in \{x_1, x_2\}$*), then* $m_{s+1} \in \Pi_{ai}(D_1^s, \overline{P'})$.

The set of all well-formed argument inquiry dialogues is denoted $\mathcal{D}_{ai}$.

I define the outcome of an argument inquiry dialogue as the set of all arguments that can be constructed from the union of the commitment stores and whose claims are the consequent of the topic of the dialogue.

Definition 4.3.4 *The* **argument inquiry outcome** *of a dialogue is a function* $\mathsf{Outcome}_{ai} : \mathcal{D}_{ai} \mapsto \wp(\mathcal{A}(\mathcal{B}))$*. If* D_r^t *is a well-formed argument inquiry dialogue with participants* x_1 *and* x_2*, then* $\mathsf{Outcome}_{ai}(D_r^t) = \{\langle \Phi, \phi\rangle \mid \mathsf{Topic}(D_r^t) = \alpha_1 \wedge \ldots \wedge \alpha_n \rightarrow \phi$ *and* $\langle \Phi, \phi\rangle \in \mathcal{A}(CS_{x_1}^t \cup CS_{x_2}^t)\}$

Example 4.3.2 *Considering the dialogue in Example 4.3.1, the outcome of the top-level argument inquiry dialogue* D_1^{11} *is* $\{\langle\{(e,1),(e \rightarrow b,2),(d,1),(d \rightarrow a,1),(a \wedge b \rightarrow c,1)\},c\rangle\}$.

$$\mathsf{Outcome}_{ai}(D_1^{11}) = \{\langle\{(e,1),(e \rightarrow b,2),(d,1),(d \rightarrow a,1),(a \wedge b \rightarrow c,1)\},c\rangle\}$$

The outcome of the embedded argument inquiry dialogue D_3^7 *is* $\{\langle\{(e,1),(e \rightarrow b,2)\},b\rangle\}$.

$$\mathsf{Outcome}_{ai}(D_3^7) = \{\langle\{(e,1),(e \rightarrow b,2)\},b\rangle\}$$

Although in this example, the outcome of each dialogue only contains one element, we will see in later examples (Section 4.7) that the outcome of an argument inquiry dialogue may sometimes be the empty set if the agents fail to jointly construct an argument for the desired claim, or may sometimes contain more than one argument if the agents can jointly construct more than one argument for the desired claim.

In this section, I have given the specific protocol that allows us to model argument inquiry dialogues. In the next section I will give details relating to warrant inquiry dialogues. In the section following that, I will provide a specific strategy for use by agents with either the argument inquiry or warrant inquiry protocol.

4.4 The warrant inquiry dialogue

The goal of a warrant inquiry dialogue is to jointly arrive at a warrant for an argument for a particular claim, which is the topic of the dialogue. This warrant takes the form of a dialectical tree. The participants take it in turn to exchange arguments that they believe to have some bearing on the status of the argument for the topic. A warrant inquiry dialogue would be opened if an agent believed that another agent had some knowledge that it was missing, and felt that this information could be relevant for consideration. For example, if a trainee doctor needed to make a critical diagnosis then they may enter into a warrant inquiry dialogue with a consultant, who the trainee would expect to have extra specialist knowledge that could help in the diagnosis. Or, as in the referral agent scenario that I described in Section 1.2.1, the knowledge of the two agents may be completely distinct (with one knowing only state beliefs and the other only domain beliefs) and so neither would be able to construct the relevant dialectical tree alone.

As two agents participating in a warrant inquiry dialogue exchange arguments, a dialectical tree is constructed that has an argument for the topic at the root, called the *root argument*. As it may be the case that more than one argument for the topic are asserted during the dialogue, the root argument is the first argument for the topic that gets asserted. If the root argument is $null$ then this means that no argument for the topic has been asserted yet.

Definition 4.4.1 *The function* $\mathsf{RootArg} : \mathcal{D} \mapsto \mathcal{A}(\mathcal{B}) \cup \{null\}$ *returns the* **root argument** *of a warrant* *inquiry dialogue. Let D_r^t be a warrant inquiry dialogue with participants x_1 and x_2.*

$$\mathsf{RootArg}(D_r^t) = \begin{cases} \langle \Gamma, \gamma \rangle & \text{if there exists an } s \text{ such that } r < s \leq t \text{ and} \\ & m_s = \langle P, assert, \langle \Gamma, \gamma \rangle \rangle \text{ and } \mathsf{Topic}(D_r^t) = \gamma \text{ and } P \in \{x_1, x_2\} \text{ and} \\ & \text{there does not exist } s' \text{ such that } r < s' < s \text{ and} \\ & \qquad \text{there exists } \Gamma' \text{ such that } m_{s'} = \langle P', assert, \langle \Gamma', \gamma \rangle \rangle \\ & \qquad \text{and } P' \in \{x_1, x_2\}, \\ & \text{else} \\ null & \text{otherwise.} \end{cases}$$

As discussed earlier, the goal of a warrant inquiry dialogue is to find a warrant for an argument for a particular claim in the form of a dialectical tree. As the agents exchange arguments, they construct a special type of dialectical tree (Definition 3.6.1) called a dialogue tree. This is simply a dialectical tree that has the root argument of the dialogue at its root, and is constructed from the contents of the two commitment stores. The dialogue tree at the end of a warrant inquiry dialogue is the warrant for the topic of the dialogue if and only if the status of the root of the dialogue tree is U. If the root argument is $null$ (meaning no argument for the topic has been asserted yet), then the dialogue tree is also $null$ (meaning that the tree is empty).

Definition 4.4.2 *A* **dialogue tree** *associated with a warrant inquiry dialogue D_r^t is a special type of dialectical tree that is denoted* $\mathsf{DialogueTree}(D_r^t)$. *If* $\mathsf{RootArg}(D_r^t) = A$ *and* $\Phi = CS_P^t \cup CS_{\overline{P}}^t$*, then* $\mathsf{DialogueTree}(D_r^t)$ *is the dialectical tree* T_A^Φ*. Otherwise, if* $\mathsf{RootArg}(D_r^t) = null$*, then* $\mathsf{DialogueTree}(D_r^t) = null$*.*

I will now give an example of a warrant inquiry dialogue illustrating the intuition behind the protocol which I will define shortly (Definition 4.4.3).

Example 4.4.1 *Let D_1^{14} be a warrant inquiry dialogue that terminates at 14 with participants x_1 and x_2 such that*

$$\Sigma^{x_1} = \{(a, 2), (c \rightarrow \neg a, 1), (c, 2), (d, 1)\}$$

$$\Sigma^{x_2} = \{(a \rightarrow b, 1), (d \rightarrow \neg c, 1)\}$$

Agent x_1 wishes to search for a warrant for an argument for b and believes that agent x_2 may be able to *help.*

$$m_1 = \langle x_2, open, dialogue(wi, b)\rangle$$
$$m_2 = \langle x_1, open, dialogue(ai, a \to b)\rangle$$
$$m_3 = \langle x_2, assert, \langle\{(a, 2)\}, a\rangle\rangle$$
$$m_4 = \langle x_1, assert, \langle\langle\{(a, 2), (a \to b, 1)\}, b\rangle\rangle$$
$$m_5 = \langle x_2, close, dialogue(ai, a \to b)\rangle$$
$$m_6 = \langle x_1, close, dialogue(ai, a \to b)\rangle$$
$$m_7 = \langle x_2, assert, \langle\{(c, 2), (c \to \neg a, 1)\}\rangle\rangle$$
$$m_8 = \langle x_1, open, dialogue(ai, d \to \neg c)\rangle$$
$$m_9 = \langle x_2, assert, \langle\{(d, 1)\}, d\rangle\rangle$$
$$m_{10} = \langle x_1, assert, \langle\langle\{(d, 1), (d \to \neg c, 1)\}, \neg c\rangle\rangle$$
$$m_{11} = \langle x_2, close, dialogue(ai, d \to \neg c)\rangle$$
$$m_{12} = \langle x_1, close, dialogue(ai, d \to \neg c)\rangle$$
$$m_{13} = \langle x_2, close, dialogue(wi, b)\rangle$$
$$m_{14} = \langle x_1, close, dialogue(wi, b)\rangle$$

Timepoint 1. *When agent x_1 makes the move m_1 it opens the warrant inquiry dialogue D_1^{10} that terminates at 10. Agent x_1 wishes to collaborate with agent x_2 to try and construct a warrant (in the form of a dialectical tree) for an argument for b. Both commitment stores are empty, $CS_{x_1}^1 = CS_{x_2}^1 = \emptyset$. The root argument is currently $null$, meaning no argument for b has been asserted yet, and so the dialogue tree is $null$,* $\mathsf{DialogueTree}(D_1^1) = null$*, meaning the current dialogue tree is empty.*

Timepoint 2. *As no argument for the topic of the dialogue has been asserted yet, agent x_2 checks to see if it can construct such an argument from its beliefs and the commitment stores. It cannot, but it does know of a defeasible rule whose consequent is the topic of the dialogue, $a \to b$, and so it opens an embedded argument inquiry dialogue D_2^6 that terminates at 6. The question store associated with this argument inquiry dialogue is $QS_2 = \{a, b\}$. Both commitment stores are still empty, $CS_{x_1}^2 = CS_{x_2}^2 = \emptyset$. As the root argument is still $null$, the dialogue tree associated with the top-level warrant inquiry dialogue (a top-dialogue of D_2^6) is still empty,* $\mathsf{DialogueTree}(D_1^2) = null$*.*

Timepoint 3. *The current dialogue is now the embedded argument inquiry dialogue D_2^3 and so x_1 examines the current question store ($QS_2 = \{a, b\}$) and sees if it can provide an argument for any of its elements. It can provide an argument for a and so asserts this argument, causing the support of this argument to be added to its commitment store, $CS_{x_1}^3 = \{(a, 2)\}$. x_2's commitment store does not change, $CS_{x_2}^3 = \emptyset$. It is still the case that no argument for the topic of the top-level warrant inquiry dialogue have yet been asserted and so the dialogue tree is still empty,* $\mathsf{DialogueTree}(D_1^3) = null$*.*

Timepoint 4. *Agent x_2 is now able to use elements from x_1's public commitment store along with its belief in $a \to b$ to construct an argument for b (an element of the current question store, $QS_2 = \{a, b\}$), and so asserts this argument. This causes the support of this argument to be added to x_2's commitment store, $CS_{x_2}^4 = \{a, 2), (a \to b, 1)\}\}$. x_1's commitment store does not change, $CS_{x_1}^4 = \{(a, 2)\}$. It is now the case that within the top-level warrant inquiry dialogue, D_1^4, an argument for the topic, $\langle\{(a, 2), (a \to b, 1)\}, b\rangle$, has been asserted and so this is the root argument and the dialogue tree*

$\mathsf{DialogueTree}(D_1^3)$ *is as follows*

U $\langle\{(a, 2), (a \rightarrow b, 1)\}, b\rangle$

Timepoint 5. *As the current dialogue is still the embedded argument inquiry dialogue D_2^5, x_1 now examines the current question store ($QS_2 = \{a, b\}$) to see if it can construct an argument for any of its elements that cannot already be constructed from the union of the commitment stores. It cannot and so makes a move to close the embedded argument inquiry dialogue. The commitment stores do not change, $CS_{x_2}^5 = \{a, 2), (a \rightarrow b, 1)\}\}$, $CS_{x_1}^5 = \{(a, 2)\}$. As the construction of the dialogue tree depends on the root argument and the contents of the commitment stores, the dialogue tree also does not change.*

Timepoint 6. *We are still in the embedded argument inquiry dialogue and so x_2 now examines the current question store ($QS_2 = \{a, b\}$) to see if it can construct an argument for any of its elements that cannot already be constructed from the union of the commitment stores. It cannot and so makes a move to close the embedded argument inquiry dialogue. As m_6 is a matched-close for D_2^6 and there are not any dialogues embedded within D_2^6 that have not yet terminated, D_2^6 terminates at 6. The commitment stores do not change, $CS_{x_2}^6 = \{a, 2), (a \rightarrow b, 1)\}\}$, $CS_{x_1}^6 = \{(a, 2)\}$. The dialogue tree does not change.*

Timepoint 7. *The current dialogue is now the top-level warrant inquiry dialogue D_1^7. As the dialogue tree is not empty, x_1 examines the current dialogue tree to see if it can construct any arguments which, if asserted, would cause some change to the dialogue tree. It does find such an argument, $\langle\{(c, 2), (c \rightarrow \neg a, 1)\}\rangle$, and so asserts this argument. The causes the support of the argument to be added to x_1's commitment store, $CS_{x_1}^7 = \{(a, 2), (c, 2), (c \rightarrow \neg a, 1)\}$. x_2's commitment store does not change, $CS_{x_2}^7 = \{a, 2), (a \rightarrow b, 1)\}\}$. The change to the commitment stores causes a new node to be added to the dialogue tree,* $\mathsf{DialogueTree}(D_1^7)$, *as follows*

U $\langle\{(c, 2), (c \rightarrow \neg a, 1)\}, \neg a\rangle$

|

D $\langle\{(a, 2), (a \rightarrow b, 1)\}, b\rangle$

Timepoint 8. *x_2 now checks to see if it can construct any argument which would change the current dialogue tree. It cannot but it does know of a defeasible rule with the consequent $\neg c$. If an argument for $\neg c$ were found then this may cause a new node to be added to the dialogue tree (dependent on whether the conditions of an acceptable argumentation line are met), and so x_2 makes a move to open an embedded argument inquiry dialogue with this defeasible rule as its topic, causing the question store $QS_8 = \{d, \neg c\}$ to be created. The commitment stores do not change $CS_{x_1}^8 = \{(a, 2), (c, 2), (c \rightarrow \neg a, 1)\}$, $CS_{x_2}^8 = \{a, 2), (a \rightarrow b, 1)\}\}$. The dialogue tree does not change.*

Timepoint 9. *x_1 checks the current question store ($QS_8 = \{d, \neg c\}$) to see if it can construct an argument for any of its elements. It finds that it can, $\langle\{(d, 1)\}, d\rangle$, and so asserts this argument. This causes the sup-*

port of this argument to be added to x_1's commitment store $CS^9_{x_1} = \{(a,2),(c,2),(c \rightarrow \neg a,1),(d,1)\}$. x_2's commitment store does not change, $CS^9_{x_2} = \{a,2),(a \rightarrow b,1)\}\}$. Although the union of the commitment stores have changed, the dialogue tree has not.

Timepoint 10. *x_2 checks the current question store ($Q_8 = \{d, \neg c\}$) to see if it can construct an argument for any of its elements. It finds that it can, $\langle\{(d,1),(d \rightarrow \neg c,1)\}, \neg c\rangle$, and so asserts this argument. This causes the support of the argument to be added to x_2's commitment store, $CS^{10}_{x_2} = \{a,2),(a \rightarrow b,1),(d,1),(d \rightarrow \neg c,1)\}\}$. x_1's commitment store does not change, $CS^{10}_{x_1} = \{(a,2),(c,2),(c \rightarrow \neg a,1),(d,1)\}$. This change to the commitment stores has caused a new node to be added to the dialogue tree as follows*

U $\langle\{(d,1),(d \rightarrow \neg c,1)\}, \neg c\rangle$

|

D $\langle\{(c,2),(c \rightarrow \neg a,1)\}, \neg a\rangle$

|

U $\langle\{(a,2),(a \rightarrow b,1)\}, b\rangle$

Timepoint 11. *x_1 checks the current question store ($QS_8 = \{d, \neg c\}$) to see if it can construct an argument for any of its elements that cannot already be constructed from the union of the public commitment stores. It cannot and so makes a move to close the embedded argument inquiry dialogue. The commitment stores do not change, $CS^{11}_{x_2} = \{a,2),(a \rightarrow b,1),(d,1),(d \rightarrow \neg c,1)\}\}$, $CS^{11}_{x_1} = \{(a,2),(c,2),(c \rightarrow \neg a,1),(d,1)\}$, and the dialogue tree does not change.*

Timepoint 12. *x_2 is also unable to construct any new arguments for an element of the current question store and so also moves to close the dialogue. As m_{12} is a matched-close for D^{12}_8 and there are not any dialogues embedded within D^{12}_8 that have not yet terminated, D^{12}_8 terminates at 12. The commitment stores do not change, $CS^{12}_{x_2} = \{a,2),(a \rightarrow b,1),(d,1),(d \rightarrow \neg c,1)\}\}$, $CS^{12}_{x_1} = \{(a,2),(c,2),(c \rightarrow \neg a,1),(d,1)\}$, and the dialogue tree does not change.*

Timepoint 13. *The current dialogue is now the top-level warrant inquiry dialogue D^{13}_1 and so x_1 checks to see if there are any arguments that it could assert which would cause the dialogue tree to change. There are not and so it checks to see whether it knows of any defeasible rules that have as a consequent something which is in conflict with an element of the union of the commitment stores (as if an argument for such a consequent could be found during an embedded argument inquiry dialogue then it may cause a new node to be added to the dialogue tree). It does not find any such defeasible rules and so moves to close the dialogue. The commitment stores do not change, $CS^{13}_{x_2} = \{a,2),(a \rightarrow b,1),(d,1),(d \rightarrow \neg c,1)\}\}$, $CS^{13}_{x_1} = \{(a,2),(c,2),(c \rightarrow \neg a,1),(d,1)\}$, and the dialogue tree does not change.*

Timepoint 14. *x_2 now performs the same checks that x_1 did at timepoint 13, but also finds that there*

are no useful moves it could make and so also moves to close the dialogue. As m_{14} is a matched close for D_1^{14}, D_1^{14} terminates at 14. The commitment stores do not change, $CS_{x_2}^{14} = \{a, 2), (a \rightarrow b, 1), (d, 1), (d \rightarrow \neg c, 1)\}\}$, $CS_{x_1}^{14} = \{(a, 2), (c, 2), (c \rightarrow \neg a, 1), (d, 1)\}$, and the dialogue tree does not change.
The final dialogue tree is

$$\begin{array}{cc} \textbf{U} & \langle\{(d,1),(d \rightarrow \neg c,1)\},\neg c\rangle \\ & | \\ \textbf{D} & \langle\{(c,2),(c \rightarrow \neg a,1)\},\neg a\rangle \\ & | \\ \textbf{U} & \langle\{(a,2),(a \rightarrow b,1)\},b\rangle \end{array}$$

and so the agents have successfully found a warrant for an argument for b.

I now give the specific protocol and outcome functions that make up the warrant inquiry dialogue. The protocol function tells us which moves are legal at a point in a warrant inquiry dialogue. Note that, as the only type of open move that it is legal to make within a warrant inquiry dialogue is one that opens an embedded argument inquiry dialogue, the only type of dialogue that can be embedded within a warrant inquiry dialogue is an argument inquiry dialogue. Recall that argument inquiry dialogue may themselves have argument inquiry dialogues embedded within them, and so it is possible to have multiple nested argument inquiry dialogues embedded within a warrant inquiry dialogue. However, it is not possible to nest warrant inquiry dialogues within other warrant inquiry dialogues or within argument inquiry dialogues. Also note that, like with the argument inquiry protocol, the warrant inquiry protocol only depends on public elements of the dialogue, hence there is no problem in checking an agent's conformance with the protocol.

Definition 4.4.3 *A* **warrant inquiry protocol** *is a function $\Pi_{wi} : \mathcal{D}_{top} \times \mathcal{I} \mapsto \wp(\mathcal{M})$. If D_1^t is a top-level dialogue with participants x_1 and x_2,* $\mathsf{Receiver}(m_t) = P$, $1 \leq t$ *and* $\mathsf{cTopic}(D_1^t) = \gamma$*, then* $\Pi_{wi}(D_1^t, P)$ *is*

$$\Pi_{wi}^{assert}(D_1^t, P) \cup \Pi_{wi}^{open}(D_1^t, P) \cup \{\langle \overline{P}, close, dialogue(wi, \gamma)\rangle\}$$

where

$$\Pi_{wi}^{assert}(D_1^t, P) = \{\langle \overline{P}, assert, \langle \Phi, \phi \rangle \rangle | \\ \mathsf{DialogueTree}(\mathsf{Current}(D_1^t) + \langle \overline{P}, assert, \langle \Phi, \phi \rangle \rangle) \neq \mathsf{DialogueTree}(\mathsf{Current}(D_1^t))\}$$

$$\Pi_{wi}^{open}(D_1^t, P) = \{\langle \overline{P}, open, dialogue(ai, \beta_1 \wedge \ldots \wedge \beta_n \rightarrow \alpha) \rangle | \\ \textit{(1)}\ \neg\alpha \in \mathsf{DefDerivations}(CS_P^t \cup CS_{\overline{P}}^t)\ \textit{or} \\ \mathsf{RootArg}(\mathsf{Current}(D_1^t)) = null\ \textit{and}\ \alpha = \gamma, \\ (2)\ \textit{there does not exist}\ t'\ \textit{such that}\ 1 < t' \leq t \\ \textit{and}\ QS_{t'} = \{\beta_1, \ldots, \beta_n, \alpha\}\}$$

An assert move is only legal if it changes the dialogue tree in some way ($\mathsf{DialogueTree}(D_1^t + \langle \overline{P}, assert, \langle \Phi, \phi \rangle \rangle) \neq \mathsf{DialogueTree}(D_1^t)$), i.e. adding the argument to the commitment stores causes a new node to be added to the dialogue tree, it does not necessarily change the status of the root node. This ensures that the dialogue stays focussed on only exchanging arguments that may have some bearing on the status of the root argument. It also ensures that agents do not repeat assert moves.

The first condition of a legal move opening an embedded argument inquiry dialogue with topic $\beta_1 \wedge \ldots \wedge \beta_n \rightarrow \alpha$ states that one of the following must hold.

- It must be possible to defeasibly derive $\neg\alpha$ from the union of the commitment stores ($\neg\alpha \in \mathsf{DefDerivations}(CS_P^t \cup CS_{\overline{P}}^t)$). This condition is again intended to ensure that the focus of the dialogue stays relevant. If an agent can form an argument for the negation of something that can be derived from the commitment stores, then that means that the argument conflicts with something that has been previously asserted, a necessity for an argument that is going to alter the dialogue tree.

- The root argument for the current dialogue is null (i.e. no argument whose claim is the topic of the current dialogue has yet been asserted during the current dialogue) and α (the consequent of the topic of the argument inquiry dialogue being opened) is the topic of the current dialogue ($\mathsf{RootArg}(\mathsf{Current}(D_1^t)) = null$ and $\alpha = \gamma$). This condition allows agents to open nested argument inquiry dialogues which will not lead to arguments that will conflict with things that have already been asserted, but which may lead to an argument for the topic of the warrant inquiry dialogue in which it is embedded, if no such argument has previously been asserted.

The second condition of a legal open move (there does not exist t' such that $1 < t' \leq t$ and $QS_{t'} = \{\beta_1, \ldots, \beta_n, \alpha\}$) ensures that open moves are not repeated by an agent (as it does in the argument inquiry dialogue). It also ensures that an agent will not open a new argument inquiry dialogue with topic $\beta_1 \wedge \ldots \wedge \beta_n \rightarrow \alpha$ if an argument inquiry dialogue has previously been opened whose topic has an antecedent that is some permutation of $\beta_1 \wedge \ldots \wedge \beta_n$, e.g. if an argument inquiry dialogue with topic $\beta_1 \wedge \beta_2 \wedge \beta_3 \rightarrow \alpha$ had previously been opened then it would not be possible to open a new argument inquiry dialogue with $\beta_2 \wedge \beta_3 \wedge \beta_1 \rightarrow \alpha$ as its topic. This helps avoid redundancy, as two such argument inquiry dialogues would lead to the same outcome.

A well-formed warrant inquiry dialogue is a dialogue whose first move is an open move that has $dialogue(wi, \phi)$ as its content where ϕ is a defeasible fact (condition (1) of the following definition). Condition (2) of the following definition ensures that there must be a continuation of the dialogue that terminates (ensuring that outer dialogues are not closed after an inner one has been opened but not yet closed), and that all the moves in the terminating continuation of the dialogue are legal according to the protocol for the relevant current dialogue.

Definition 4.4.4 *A* **well-formed warrant inquiry dialogue** *is a dialogue of the form* $D_r^t = [m_r, \ldots, m_t]$ *with participants* x_1 *and* x_2 *such that*

1. $m_r = \langle P, open, dialogue(wi, \phi)\rangle$ *where* $P \in \{x_1, x_2\}$ *and* $\phi \in \mathcal{S}^\star$ *(i.e.* ϕ *is a defeasible fact),*
2. *there exists* t' *such that* $t \leq t'$, $D_r^{t'}$ *extends* D_r^t*, and* $D_r^{t'}$ *terminates at* t'*, and*
   for all s *such that* $r \leq s < t'$ *and* $D_r^{t'}$ *extends* D_r^s*, if* $D_1^{t'}$ *is a top-dialogue of* $D_r^{t'}$ *and* D_1^s *is a top-dialogue of* D_r^s *and* $D_1^{t'}$ *extends* D_1^s *and* $\mathsf{Receiver}(m_s) = P'$ *(where* $P' \in \{x_1, x_2\}$*) and* $\mathsf{cType}(D_r^s) = \theta$*, then* $m_{s+1} \in \Pi_\theta(D_1^s, \overline{P'})$.

The set of all well-formed warrant inquiry dialogues is denoted $\mathcal{D}_{wi}$.

Note, if I refer simply to a well-formed dialogue, then I mean either well-formed argument inquiry dialogue or a well-formed warrant inquiry dialogue.

Definition 4.4.5 *A* **well-formed dialogue** *is either a well-formed argument inquiry dialogue or a well-formed warrant inquiry dialogue.*

The outcome of a warrant inquiry dialogue is determined by its dialogue tree. If the root argument is undefeated in the dialogue tree then a warranted argument for the topic of the dialogue has successfully been found and the set containing the root argument is the outcome, otherwise the outcome is the empty set.

Definition 4.4.6 *The* **warrant inquiry outcome** *of a dialogue is a special type of outcome function* $\mathsf{Outcome}_{wi}$ *such that* $\mathsf{Outcome}_{wi} : \mathcal{D}_{wi} \mapsto \wp(\mathcal{A}(\mathcal{B}))$*. Let* D_r^t *be a well formed warrant inquiry dialogue.*

$$\mathsf{Outcome}_{wi}(D_r^t) = \begin{cases} \{\mathsf{RootArg}(D_r^t)\} & \textit{if } \mathsf{Status}(\mathsf{Root}(\mathsf{DialogueTree}(D_r^t)), \mathsf{DialogueTree}(D_r^t)) \\ & \quad = \mathsf{U}, \textit{ else} \\ \emptyset & \textit{if } \mathsf{Status}(\mathsf{Root}(\mathsf{DialogueTree}(D_r^t)), \mathsf{DialogueTree}(D_r^t)) \\ & \quad = \mathsf{D} \textit{ or } \mathsf{RootArg}(D_r^T) = null. \end{cases}$$

Example 4.4.2 *Considering the dialogue in Example 4.4.1, the outcome of the top-level warrant inquiry dialogue* D_1^{14} *is* $\{\langle\{(a, 2), (a \rightarrow b, 1)\}, b\rangle\}$.

$$\mathsf{Outcome}_{wi}(D_1^{14}) = \{\langle\{(a, 2), (a \rightarrow b, 1)\}, b\rangle\}$$

The outcome of the embedded argument inquiry dialogue D_2^6 *is* $\{\langle\{(a,2),(a \rightarrow b,1)\},b\rangle\}$.

$$\mathsf{Outcome}_{ai}(D_2^6) = \{\langle\{(a,2),(a \rightarrow b,1)\},b\rangle\}$$

The outcome of the embedded argument inquiry dialogue D_8^{12} *is* $\{\langle\{(d,1),(d \rightarrow \neg c,1)\},\neg c\rangle\}$.

I have now defined the specific details of the argument inquiry and warrant inquiry dialogue, however this is not yet sufficient for generating inquiry dialogues, as there is no mechanism for selecting exactly one of the legal moves returned by a protocol. I address this in the next section by providing a specific agent strategy for use in a dialogue.

4.5 The exhaustive strategy

I now define a strategy for use by agents participating in a well-formed, top-level dialogue. This is a function that returns exactly one of the legal moves at a particular point in a dialogue. It is this function that sets my system apart from many of the comparable existing systems, as it allows the actual generation of dialogues. Most systems only go so far as to provide something equivalent to my protocol function (e.g. [48, 56]). Such systems are intended for modelling legal dialogues, whilst my system allows automatic generation of dialogues, by providing specific strategy functions that allow agents to intelligently select one specific, legal move to make. A strategy function takes the top-level dialogue that two agents are participating in, and the identifier of the agent whose turn it is to move, and returns exactly one move to be made.

Definition 4.5.1 *A* **strategy**, Ω_θ, *is a function such that* $\theta \in \{wi, ai\}$ *and* $\Omega_\theta : \mathcal{D}_{top} \times \mathcal{I} \mapsto \mathcal{M}$.

I will now define a specific strategy called the *exhaustive strategy*. A strategy is personal to an agent, as the move that it returns depends on the agent's private beliefs. The exhaustive strategy states that if there are any legal moves that assert an argument which can be constructed by the agent, then the most preferred of these moves is selected, else if there are any legal open moves with a defeasible rule as their content that is in the agent's beliefs, then the most preferred of these moves is selected. If there are no such moves then a close move is made.

In order to select the most preferred of the legal assert or open moves, I assign a unique number to the move content and carry out a comparison of these numbers. Let us assume that $\mathcal{B}^\star$ is composed of a finite number Z of atoms. Let us also assume that there is a registration function μ over these atoms: so, for a literal α, $\mu(\alpha)$ returns a unique single digit number base Z (this number is only like an id number and can be arbitrarily assigned). For a rule $\alpha_1 \wedge \ldots \wedge \alpha_n \rightarrow \alpha_{n+1}$, $\mu(\alpha_1 \wedge \ldots \wedge \alpha_n \rightarrow \alpha_{n+1})$ is an $n+1$ digit number of the form $\mu(\alpha_1)\ldots\mu(\alpha_n)\mu(\alpha_{n+1})$. This gives a unique base Z number for each formula in $\mathcal{B}^\star$ and allows an agent to choose the most preferred open move using the natural ordering relation $<$ over base Z numbers.

Definition 4.5.2 *Let* $\Xi = \{\langle P, open, dialogue(\theta_1,\phi_1)\rangle, \ldots, \langle P, open, dialogue(\theta_k,\phi_k)\rangle\}$ *be a set of open moves made by agent* $\overline{P}$. *The function* $\mathsf{Pref_o}$ *returns the* **preferred open move** *to make.* $\mathsf{Pref_o}(\Xi) = \langle P, open, dialogue(\theta_i,\phi_i)\rangle$, $1 \le i \le k$, *such that for all* j, $1 \le j \le k$, *if* $i \neq j$, *then* $\mu(\phi_i) < \mu(\phi_j)$.

If the set Ξ taken by the function $\mathsf{Pref_o}$ is not the empty set then $\mathsf{Pref_o}$ always returns a unique open move. I now give an example to illustrate the function $\mathsf{Pref_o}$.

Example 4.5.1 *Let us assume that we have a set of possible open moves Ξ as follows.*

$$\Xi = \{\langle P, open, dialogue(ai, b \wedge a \rightarrow c)\rangle, \langle P, open, dialogue(ai, c \rightarrow a)\rangle, \\ \langle P, open, dialogue(ai, a \wedge b \rightarrow c)\rangle\}$$

Let us also assume that the registration function μ arbitrarily assigns a single digit number base Z as follows.

$$\mu(a) = 1, \quad \mu(b) = 2, \quad \mu(c) = 3$$

This gives us the following unique base 3 numbers for the defeasible rules that appear in the above open moves.

$$\mu(b \wedge a \rightarrow c) = 213, \quad \mu(c \rightarrow a) = 31, \quad \mu(a \wedge b \rightarrow c) = 123$$

As $\mu(c \rightarrow a) < \mu(a \wedge b \rightarrow c) < \mu(b \wedge a \rightarrow c)$, we get $\mathsf{Pref_o}(\Xi) = \langle P, open, dialogue(ai, c \rightarrow a)\rangle$.

I similarly assign a number to each argument in $\mathcal{A}(\mathcal{B})$ using a registration function λ together with μ. For an argument $\langle\{(\phi_1, L_1), \ldots, (\phi_n, L_n)\}, \phi_{n+1}\rangle$, $\lambda(\langle\{(\phi_1, L_1), \ldots, (\phi_n, L_n)\}, \phi_{n+1}\rangle) = \langle d_1, \ldots d_n, d_{n+1}\rangle$ where $d_1 < \ldots < d_n < d_{n+1}$ and $\langle d_1, \ldots, d_n, d_{n+1}\rangle$ is a permutation of $\langle\mu(\phi_1), \ldots, \mu(\phi_n), \mu(\phi_{n+1})\rangle$, (where μ is the registration function for $\mathcal{B}$). The function λ returns a unique tuple of base Z numbers for each argument. I use a standard lexicographical comparison, denoted $\prec_{lex}$, of these tuples of numbers to chose the most preferred move content (i.e. the maximum element in the lexicographical ordering).

Definition 4.5.3 *Let $\Xi = \{\langle P, assert, \langle\Phi_1, \phi_1\rangle\rangle, \ldots, \langle P, assert, \langle\Phi_k, \phi_k\rangle\rangle\}$ be a set of assert moves made by agent $\overline{P}$. The function* $\mathsf{Pref_a}$ *returns the* **preferred assert move** *to make.* $\mathsf{Pref_a}(\Xi) = \langle P, assert, \langle\Phi_i, \phi_i\rangle\rangle$, *$1 \leq i \leq k$, such that for all j, $1 \leq j \leq k$, if $i \neq j$, then $\lambda(\langle\Phi_i, \phi_i\rangle) \prec_{lex} \lambda(\langle\Phi_j, \phi_j\rangle)$.*

If the set Ξ taken by the function $\mathsf{Pref_a}$ is not the empty set then $\mathsf{Pref_a}$ always returns a unique assert move. Again, I give an example to illustrate this function.

Example 4.5.2 *Let us assume that we have a set of possible assert moves Ξ as follows.*

$$\Xi = \{\langle P, assert, \langle\{(b,1), (a,1), (b \wedge a \rightarrow c, 1)\}, c\rangle\rangle, \langle P, assert, \langle\{(c,1), (c \rightarrow a, 1)\}, a\rangle\rangle, \\ \langle P, assert, \langle\{(a,1), (b,1), (a \wedge b \rightarrow c, 1)\}, c\rangle\rangle\}$$

Let us also assume that the registration function μ arbitrarily assigns a single digit number base Z as follows.

$$\mu(a) = 1, \quad \mu(b) = 2, \quad \mu(c) = 3$$

This gives us the following unique tuples of base 3 numbers for the arguments that appear in the above assert moves.

$$\lambda(\langle\{(b,1), (a,1), (b \wedge a \rightarrow c, 1)\}, c\rangle) = \langle 1, 2, 213\rangle$$

$$\lambda(\langle\{(c,1),(c \to a,1)\}, a\rangle) = \langle 3, 31\rangle$$

$$\lambda(\langle\{(a,1),(b,1),(a \wedge b \to c,1)\}, c\rangle) = \langle 1, 2, 123\rangle$$

As $\lambda(\langle\{(a,1),(b,1),(a \wedge b \to c,1)\}, c\rangle) \prec_{lex} \lambda(\langle\{(b,1),(a,1),(b \wedge a \to c,1)\}, c\rangle) \prec_{lex} \lambda(\langle\{(c,1),(c \to a,1)\}, a\rangle)$*, we get* $\mathsf{Pref_a}(\Xi) = \langle P, assert, \langle\{(a,1),(b,1),(a \wedge b \to c,1)\}, c\rangle\rangle$.

I now define the exhaustive strategy. This defines a set of legal assert moves and a set of legal open moves. If the set of legal assert moves is not empty, then the most preferred of these moves is made, else if the set of legal open moves is not empty, then the most preferred of these is made, else a close move is made.

Definition 4.5.4 *The* **exhaustive strategy** *is a function* $\Omega_{exh} : \mathcal{D}_{top} \times \mathcal{I} \mapsto \mathcal{M}$*, where* $\mathsf{cTopic}(D_1^t) = \gamma$*,* $\mathsf{cType}(D_1^t) = \theta$ *and*

$$\Omega_{exh}(D_1^t, P) = \begin{cases} \mathsf{Pref_a}(\mathsf{Asserts}_{exh}(D_1^t, P)) & \textit{iff } \mathsf{Asserts}_{exh}(D_1^t, P) \neq \emptyset \\ \mathsf{Pref_o}(\mathsf{Opens}_{exh}(D_1^t, P)) & \textit{iff } \mathsf{Asserts}_{exh}(D_1^t, P) = \emptyset \textit{ and } \mathsf{Opens}_{exh}(D_1^t, P) \neq \emptyset \\ \langle P, close, dialogue(\theta, \gamma)\rangle & \textit{iff } \mathsf{Asserts}_{exh}(D_1^t, P) = \emptyset \textit{ and } \mathsf{Opens}_{exh}(D_1^t, P) = \emptyset \end{cases}$$

where

$$\mathsf{Asserts}_{exh}(D_1^t, P) = \{\langle \overline{P}, assert, \langle\Phi, \phi\rangle\rangle \in \Pi_\theta^{assert}(D_1^t, P) \mid \langle\Phi, \phi\rangle \in \mathcal{A}(\Sigma^P \cup CS_{\overline{P}}^t)\}$$

$$\mathsf{Opens}_{exh}(D_1^t, P) = \{\langle \overline{P}, open, dialogue(ai, \psi)\rangle \in \Pi_\theta^{open}(D_1^t, P) \mid (\psi, L) \in \Sigma^P\}$$

This strategy is called the exhaustive strategy as it ensures that all moves which might have a bearing on the outcome of the dialogue will get made. This is because an agent will not make a close move unless it cannot make any more assert or open moves and both agents must make a close move in order to terminate the dialogue.

Note the restrictions on the sets of legal moves from which an agent can pick a next move. As I am considering a cooperative domain, an agent will only assert an argument that it can construct from the union of its beliefs and the other agent's commitment store (and so will not make arguments up or deliberately deceive). Agents are restricted to only opening a new argument inquiry dialogue with topic ϕ if they have a belief (ϕ, L). This prevents an agent from opening a nested sub-dialogue unless it at least knows of a rule that might help construct the desired argument.

I now define a well-formed exhaustive dialogue. This is a well-formed dialogue that is generated by two agents who both follow the exhaustive strategy at all times.

Definition 4.5.5 *A* **well-formed exhaustive dialogue** *is a well-formed dialogue* D_r^t *with participants* x_1 *and* x_2 *such that*

for all s *such that* $r \leq s < t$ *and* D_r^t *extends* D_r^s*,*
if D_1^t *is a top-dialogue of* D_r^t *and* D_1^s *is a top-dialogue of* D_r^s
and D_1^t *extends* D_1^s *and* $\mathsf{Receiver}(m_s) = P$ *(where* $P \in \{x_1, x_2\}$*)*
and $\mathsf{cType}(D_r^s) = \theta$*,*
then $\Omega_{exh}(D_1^s, P) = m_{s+1}$

In the next section I give an algorithm to demonstrate how an agent uses the exhaustive strategy to generate a well-formed exhaustive dialogue.

4.6 Dialogue behaviour

In order to demonstrate how the agents actually behave during a dialogue, I will give an algorithm that simulates a top level dialogue, shown in Figure 4.2. The algorithm simulates two agents, P (who makes the first move opening the dialogue) and $\overline{P}$, that enter into a top-level dialogue of type θ with topic ϕ, and who are both using the exhaustive strategy.

It is important to note that I assume some higher-level planning component, beyond the scope of this work, that has led to P's decision to enter into a top level dialogue of type θ with topic ϕ, i.e. an agent P has some mechanism which determines when it should make the move $m_1 = \langle \overline{P}, open, dialogue(\theta, \phi) \rangle$ to agent $\overline{P}$. The agents then take it in turns to apply the exhaustive strategy, that in turn calls the relevant protocol for the current dialogue, in order to select the next move to make. If an assert move is made then the relevant commitment store is updated.

In the following section I give examples of well-formed exhaustive dialogues generated in this way.

4.7 Dialogue examples

In this section I give examples of well-formed exhaustive dialogues that take place between two agents, x_1 and x_2. Throughout all the examples in this section, I will assume that $\mu(a) = 1$, $\mu(\neg a) = 2$, $\mu(b) = 3$, $\mu(\neg b) = 4$, $\mu(c) = 5$, $\mu(\neg c) = 6$, $\mu(d) = 7$ etc.

I represent a top-level dialogue as a table, the first column of which gives the value of t, the second column gives the commitment store of agent x_1, the third column gives the move m_t, the fourth column gives agent x_2's commitment store, and the fifth column gives the details of any question stores that are not equal to the empty set.

4.7.1 Argument inquiry dialogue example 1

In the following example we have an agent x_1 who wishes to enter into a dialogue with agent x_2 in order to try to find an argument for d. We have

$$\Sigma^{x_1} = \{(b \wedge c \rightarrow d, 1)\} \qquad \Sigma^{x_2} = \{(c, 1), (b, 1)\}$$

Agent x_1 is aware of a defeasible rule whose consequent is d, and so opens an argument inquiry dialogue with this defeasible rule as its topic. The dialogue proceeds as in Table 4.2.

Note that at $t = 3$ agent x_1 makes a move to close the dialogue, as it cannot assert any arguments for an element in the commitment store, nor can it open any new argument inquiry dialogues. However, this does not close the dialogue, as each agent must make a close move in succession in order to close the dialogue.

As the agents successfully find an argument for d, the outcome of the dialogue is this argument.

$topDialogue(P, \overline{P}, \theta, \phi)$:

$t = 1;$

$CS_P^t = \{\}$

$CS_{\overline{P}}^t = \{\}$

$m_t = \langle \overline{P}, open, dialogue(\theta, \phi) \rangle$

$D_1^t = [m_t]$

while D_1^t is not terminated

$t = t + 1$

$m_t = \Omega_{exh}(D_1^{t-1}, \overline{P})$

$D_1^t = D_1^{t-1} + m_t$

$CS_P^t = CS_P^{t-1}$

if $m_t = \langle P, assert, \langle \Psi, \psi \rangle \rangle$

then $CS_{\overline{P}}^t = CS_{\overline{P}}^{t-1} \cup \Psi$

else $CS_{\overline{P}}^t = CS_{\overline{P}}^{t-1}$

if D_1^t is not terminated **then**

$t = t + 1$

$m_t = \Omega_{exh}(D_1^{t-1}), P)$

$D_1^t = D_1^{t-1} + m_t$

$CS_{\overline{P}}^t = CS_{\overline{P}}^{t-1}$

if $m_t = \langle \overline{P}, assert, \langle \Psi, \psi \rangle \rangle$

then $CS_P^t = CS_P^{t-1} \cup \Psi$

else $CS_P^t = CS_P^{t-1}$

return D_1^t

Figure 4.2: An algorithm that simulates the behaviour of a top-level dialogue. This algorithm takes as input the identifier of the agent who opens the dialogue ($P \in \mathcal{I}$), the identifier of the other agent participating in the dialogue ($\overline{P} \in \mathcal{I}$), the type of the dialogue ($\theta \in \{wi, ai\}$), and the topic of the dialogue ($\phi \in \mathcal{B}^\star$, i.e. ϕ is either a defeasible rule if $\theta = ai$, or a defeasible fact if $\theta = wi$). It returns a well-formed top-level exhaustive dialogue.

t	$CS^t_{x_1}$	m_t	$CS^t_{x_2}$	QS_t
1		$\langle x_2, open, dialogue(ai, b \wedge c \rightarrow d)\rangle$		$QS_1 = \{b, c, d\}$
2		$\langle x_1, assert, \langle\{(b, 1)\}, b\rangle\rangle$	$(b, 1)$	
3		$\langle x_2, close, dialogue(ai, b \wedge c \rightarrow d)\rangle$		
4		$\langle x_1, assert, \langle\{(c, 1)\}, b\rangle\rangle$	$(c, 1)$	
5	$(b, 1), (c, 1)$	$\langle x_2, assert, \langle\{(b, 1), (c, 1), (b \wedge c \rightarrow d, 1)\}, d\rangle\rangle$		
	$(b \wedge c \rightarrow d, 1)$			
6		$\langle x_1, close, dialogue(ai, b \wedge c \rightarrow d)\rangle$		
7		$\langle x_2, close, dialogue(ai, b \wedge c \rightarrow d)\rangle$		

Table 4.2: Argument inquiry dialogue example 1.

t	$CS^t_{x_1}$	m_t	$CS^t_{x_2}$	QS_t
1		$\langle x_2, open, dialogue(ai, b \wedge c \rightarrow d)\rangle$		$QS_1 = \{b, c, d\}$
2		$\langle x_1, assert, \langle\{(c, 1)\}, c\rangle\rangle$	$(c, 1)$	
3		$\langle x_2, close, dialogue(ai, b \wedge c \rightarrow d)\rangle$		
4		$\langle x_1, close, dialogue(ai, b \wedge c \rightarrow d)\rangle$		

Table 4.3: Argument inquiry dialogue example 2.

$$\mathsf{Outcome}_{ai}(D_1^7) = \{\langle\{(b, 1), (c, 1), (b \wedge c \rightarrow d, 1)\}, d\rangle\}$$

Note that the commitment stores build up monotonically, so, for example, $CS^1_{x_2} = \emptyset$, $CS^2_{x_2} = \{(b, 1)\}$ and $CS^4_{x_2} = \{(b, 1), (c, 1)\}$.

4.7.2 Argument inquiry dialogue example 2

In the following example we have an agent x_1 who wishes to enter into a dialogue with agent x_2 in order to try to find an argument for d. We have

$$\Sigma^{x_1} = \{(b \wedge c \rightarrow d, 1)\} \qquad \Sigma^{x_2} = \{(c, 1)\}$$

Agent x_1 is aware of a defeasible rule whose consequent is d, and so opens an argument inquiry dialogue with this defeasible rule as its topic. The dialogue proceeds as in Table 4.3.

As the agents do not find an argument for d, the outcome of the dialogue is the empty set.

$$\mathsf{Outcome}_{ai}(D_1^4) = \emptyset$$

4.7.3 Argument inquiry dialogue example 3

In the following example we have an agent x_1 who wishes to enter into a dialogue with agent x_2 in order to try to find an argument for d. We have

t	$CS^t_{x_1}$	m_t	$CS^t_{x_2}$	QS_t
1		$\langle x_2, open, dialogue(ai, b \wedge c \rightarrow d)\rangle$		$QS_1 = \{b, c, d\}$
2		$\langle x_1, assert, \langle\{(c, 1)\}, c\rangle\rangle$	$(c, 1)$	
3	$(b, 1)$	$\langle x_2, assert, \langle\{(b, 1)\}, b\rangle\rangle$		
4		$\langle x_1, close, dialogue(ai, b \wedge c \rightarrow d)\rangle$		
5	$(c, 1)$	$\langle x_2, assert, \langle\{(b, 1), (c, 1), (b \wedge c \rightarrow d, 1)\}, d\rangle\rangle$		
	$(b \wedge c \rightarrow d, 1)$			
6		$\langle x_1, close, dialogue(ai, b \wedge c \rightarrow d)\rangle$		
7		$\langle x_2, close, dialogue(ai, b \wedge c \rightarrow d)\rangle$		

Table 4.4: Argument inquiry dialogue example 3.

$$\Sigma^{x_1} = \{(b \wedge c \rightarrow d, 1), (b, 1)\} \qquad \Sigma^{x_2} = \{(c, 1)\}$$

Agent x_1 is aware of a defeasible rule whose consequent is d and so opens an argument inquiry dialogue with this defeasible rule as its topic. The dialogue proceeds as in Table 4.4.

As the agents successfully find an argument for d, the outcome of the dialogue is this argument.

$$\mathsf{Outcome}_{ai}(D_1^7) = \{\langle\{(b, 1), (c, 1), (b \wedge c \rightarrow d, 1)\}, d\rangle\}$$

4.7.4 Argument inquiry dialogue example 4

In the following example we have an agent x_1 who wishes to enter into a dialogue with agent x_2 in order to try to find an argument for c. We have

$$\Sigma^{x_1} = \{(b \rightarrow c, 1), (a, 1)\} \qquad \Sigma^{x_2} = \{(a \rightarrow b, 1)\}$$

Agent x_1 is aware of a defeasible rule whose consequent is c and so opens an argument inquiry dialogue with this defeasible rule as its topic. The dialogue proceeds as in Table 4.5.

As the agents successfully find an argument for c, the outcome of the dialogue is this argument.

$$\mathsf{Outcome}_{ai}(D_1^9) = \{\langle\{(a, 1), (a \rightarrow b, 1), (b \rightarrow c, 1)\}, c\rangle\}$$

There is also an nested argument inquiry sub-dialogue, D_2^6, that terminates at 6 and whose topic is $a \rightarrow b$. As the agents successfully find an argument for b, the outcome of the dialogue is this argument.

$$\mathsf{Outcome}_{ai}(D_2^6) = \{\langle\{(a, 1), (a \rightarrow b, 1)\}, b\rangle\}$$

4.7.5 Argument inquiry dialogue example 5

In the following example we have an agent x_1 who wishes to enter into a dialogue with agent x_2 in order to try to find an argument for e. We have

t	$CS^t_{x_1}$	m_t	$CS^t_{x_2}$	QS_t
1		$\langle x_2, open, dialogue(ai, b \rightarrow c)\rangle$		$QS_1 = \{b, c\}$
2		$\langle x_1, open, dialogue(ai, a \rightarrow b)\rangle$		$QS_2 = \{a, b\}$
3	$(a, 1)$	$\langle x_2, assert, \langle\{(a, 1)\}, a\rangle\rangle$		
4		$\langle x_1, assert, \langle\{(a, 1), (a \rightarrow b, 1)\}, b\rangle$	$(a, 1)$, $(a \rightarrow b, 1)$	
5		$\langle x_2, close, dialogue(ai, a \rightarrow b)\rangle$		
6		$\langle x_1, close, dialogue(ai, a \rightarrow b)\rangle$		
7	$(a \rightarrow b, 1)$ $(b \rightarrow c, 1)$	$\langle x_2, assert, \langle\{(a, 1), (a \rightarrow b, 1), (b \rightarrow c, 1)\}, c\rangle\rangle$		
8		$\langle x_1, close, dialogue(ai, b \rightarrow c)\rangle$		
9		$\langle x_2, close, dialogue(ai, b \rightarrow c)\rangle$		

Table 4.5: Argument inquiry dialogue example 4.

$$\Sigma^{x_1} = \{(b \wedge d \rightarrow e, 1), (c \rightarrow d, 1)\} \qquad \Sigma^{x_2} = \{(a \rightarrow b, 1), (a, 1)\}$$

Agent x_1 is aware of a defeasible rule whose consequent is e and so opens an argument inquiry dialogue with this defeasible rule as its topic. The dialogue proceeds as in Table 4.6.

As the agents do not successfully find an argument for e, the outcome of the top-level dialogue is the empty set.

$$\mathsf{Outcome}_{ai}(D_1^{10}) = \emptyset$$

Note that there are two nested argument inquiry dialogues that appear within the top level dialogue D_1^{10}: D_3^5 that terminates at 5 and whose topic is $c \rightarrow d$, and D_6^8 that terminates at 8 and whose topic is $a \rightarrow b$. As the agents do not successfully find an argument for d, the outcome of D_3^5 is also the empty set.

$$\mathsf{Outcome}_{ai}(D_3^5) = \emptyset$$

As there is an argument for b in the union of the commitment stores, the outcome of D_6^8 is the set containing this argument.

$$\mathsf{Outcome}_{ai}(D_6^8) = \{\langle\{(a, 1), (a \rightarrow b, 1)\}, b\rangle\}$$

t	$CS^t_{x_1}$	m_t	$CS^t_{x_2}$	QS_t
1		$\langle x_2, open, dialogue(ai, b \wedge d \rightarrow e)\rangle$		$QS_1 = \{b, d, e\}$
2		$\langle x_1, assert, \langle\{(a,1), (a \rightarrow b, 1)\}, b\rangle\rangle$	$(a,1)$ $(a \rightarrow b, 1)$	
3		$\langle x_2, open, dialogue(ai, c \rightarrow d)\rangle$		$QS_3 = \{c, d\}$
4		$\langle x_1, close, dialogue(ai, c \rightarrow d)\rangle$		
5		$\langle x_2, close, dialogue(ai, c \rightarrow d)\rangle$		
6		$\langle x_1, open, dialogue(ai, a \rightarrow b)\rangle$		$QS_6 = \{a, b\}$
7		$\langle x_2, close, dialogue(ai, a \rightarrow b)\rangle$		
8		$\langle x_1, close, dialogue(ai, a \rightarrow b)\rangle$		
9		$\langle x_2, close, dialogue(ai, b \wedge d \rightarrow e)\rangle$		
10		$\langle x_1, close, dialogue(ai, b \wedge d \rightarrow e)\rangle$		

Table 4.6: Argument inquiry dialogue example 5.

4.7.6 Argument inquiry dialogue example 6

In the following example we have an agent x_1 who wishes to enter into a dialogue with agent x_2 in order to try to find an argument for c. We have

$$\Sigma^{x_1} = \{(d,1), (b \rightarrow c, 1)\} \qquad \Sigma^{x_2} = \{(e,1), (d \wedge e \rightarrow b, 1), (a \rightarrow b, 1)\}$$

Agent x_1 is aware of a defeasible rule whose consequent is c and so opens an argument inquiry dialogue with this defeasible rule as its topic. The dialogue proceeds as in Table 4.7.

As the agents successfully find an argument for c, the outcome of the top-level dialogue, D_1^{13}, is this argument.

$$\mathsf{Outcome}_{ai}(D_1^{13}) = \{\langle\{(d,1), (e,1), (d \wedge e \rightarrow b, 1), (b \rightarrow c, 1)\}, c\rangle\}$$

Note that there are two nested argument inquiry dialogues that appears within the top-level dialogue: D_2^{10} that terminates at 10 whose topic is $a \rightarrow b$; and D_4^8 that terminates at 8 and whose topic is $d \wedge e \rightarrow b$.

$$\mathsf{Outcome}_{ai}(D_2^{10}) = \{\langle\{(d,1), (e,1), (d \wedge e \rightarrow b, 1)\}, b\rangle\}$$

$$\mathsf{Outcome}_{ai}(D_4^{8}) = \{\langle\{(d,1), (e,1), (d \wedge e \rightarrow b, 1)\}, b\rangle\}$$

t	$CS^t_{x_1}$	m_t	$CS^t_{x_2}$	QS_t
1		$\langle x_2, open, dialogue(ai, b \rightarrow c)\rangle$		$QS_1 = \{b, c\}$
2		$\langle x_1, open, dialogue(ai, a \rightarrow b)\rangle$		$QS_2 = \{a, b\}$
3		$\langle x_2, close, dialogue(ai, a \rightarrow b)\rangle$		
4		$\langle x_1, open, dialogue(ai, d \wedge e \rightarrow b)\rangle$		$QS_4 = \{d, e, b\}$
5	$(d, 1)$	$\langle x_2, assert, \langle\{(d, 1)\}, d\rangle\rangle$		
6		$\langle x_1, assert, \langle\{(d, 1), (e, 1), (d \wedge e \rightarrow b, 1)\}, b\rangle\rangle$	$(d, 1), (e, 1)$ $(d \wedge e \rightarrow b, 1)$	
7		$\langle x_2, close, dialogue(ai, d \wedge e \rightarrow b)\rangle$		
8		$\langle x_1, close, dialogue(ai, d \wedge e \rightarrow b)\rangle$		
9		$\langle x_2, close, dialogue(ai, a \rightarrow b)\rangle$		
10		$\langle x_1, close, dialogue(ai, a \rightarrow b)\rangle$		
11	$(e, 1)$ $(d \wedge e \rightarrow b, 1)$ $(b \rightarrow c, 1)$	$\langle x_2, assert, \langle\{(d, 1), (e, 1), (d \wedge e \rightarrow b, 1), (b \rightarrow c, 1)\}, c\rangle$		
12		$\langle x_1, close, dialogue(ai, b \rightarrow c)\rangle$		
13		$\langle x_2, close, dialogue(ai, b \rightarrow c)\rangle$		

Table 4.7: Argument inquiry dialogue example 6.

4.7.7 Argument inquiry dialogue example 7

In the following example we have an agent x_1 who wishes to enter into a dialogue with agent x_2 in order to try to find an argument for d. We have

$$\Sigma^{x_1} = \{(c \rightarrow d, 1), (b \rightarrow c, 1), (a \rightarrow b, 1)\} \qquad \Sigma^{x_2} = \{(a, 1), (b, 1)\}$$

Agent x_1 is aware of a defeasible rule whose consequent is d and so opens an argument inquiry dialogue with this defeasible rule as its topic. The dialogue proceeds as in Table 4.8.

As there are two arguments for d that can be constructed at the end of the top-level dialogue D_1^{17}, the outcome of this dialogue is the set of these two arguments.

$$\begin{aligned}\mathsf{Outcome}_{ai}(D_1^{17}) = \{&\langle\{(a, 1), (a \rightarrow b, 1), (b \rightarrow c, 1), (c \rightarrow d, 1)\}, d\rangle,\\ &\langle\{(b, 1), (b \rightarrow c, 1), (c \rightarrow d, 1)\}, d\rangle\}\end{aligned}$$

There are two sub-dialogues of D_1^{17}: D_3^{13} that terminates at 13 and has topic $b \rightarrow c$; and D_7^{11} that terminates at 11 and has topic $a \rightarrow b$. D_7^{11} is also a sub-dialogue of of D_3^{13}.

$$\mathsf{Outcome}_{ai}(D_3^{13}) = \{\langle\{(a, 1), (a \rightarrow b, 1), (b \rightarrow c, 1)\}, c\rangle, \langle\{(b, 1), (b \rightarrow c, 1)\}, c\rangle\}$$

$$\mathsf{Outcome}_{ai}(D_7^{11}) = \{\langle\{(a, 1), (a \rightarrow b, 1)\}, b\rangle, \langle\{(b, 1)\}, b\rangle\}$$

4.7.8 Argument inquiry dialogue example 8

In the following example we have an agent x_1 who wishes to enter into a dialogue with agent x_2 in order to try to find an argument for b. We have

$$\Sigma^{x_1} = \{(a \rightarrow b, 1)\} \qquad \Sigma^{x_2} = \{(b \rightarrow a)\}$$

This is an interesting example to consider as one might worry that this would lead to an infinite dialogue, however, as agents may not repeat moves that have already been made, this is not the case.

Agent x_1 is aware of a defeasible rule whose consequent is b and so opens an argument inquiry dialogue with this defeasible rule as its topic. The dialogue proceeds as in Table 4.9.

As there are no arguments for b that can be constructed from the union of the commitment stores at the end of the top-level dialogue D_1^6, the outcome of this dialogue is the emptyset. The outcome of the embedded dialogue D_2^4 is also the emptyset as no arguments for a can be constructed.

$$\mathsf{Outcome}_{ai}(D_1^6) = \mathsf{Outcome}_{ai}(D_2^4) = \emptyset$$

4.7.9 Warrant inquiry dialogue example 1

In the following example we have an agent x_1 who wishes to enter into a dialogue with agent x_2 in order to try to find a warrant for an argument for b. We have

t	$CS^t_{x_1}$	m_t	$CS^t_{x_2}$	QS_t
1		$\langle x_2, open, dialogue(ai, c \rightarrow d)\rangle$		$QS_1 = \{c, d\}$
2		$\langle x_1, close, dialogue(ai, c \rightarrow d)\rangle$		
3		$\langle x_2, open, dialogue(ai, b \rightarrow c)\rangle$		$QS_3 = \{b, c\}$
4		$\langle x_1, assert, \langle\{(b, 1)\}, b\rangle\rangle$	$(b, 1)$	
5	$(b, 1)$ $(b \rightarrow c, 1)$	$\langle x_2, assert, \langle\{(b, 1), (b \rightarrow c, 1)\}, c\rangle\rangle$		
6		$\langle x_1, close, dialogue(ai, b \rightarrow c)\rangle$		
7		$\langle x_2, open, dialogue(ai, a \rightarrow b)\rangle$		$QS_7 = \{a, b\}$
8		$\langle x_1, assert, \langle\{(a, 1)\}, a\rangle\rangle$	$(a, 1)$	
9	$(a, 1)$ $(a \rightarrow b, 1)$	$\langle x_2, assert, \langle\{(a, 1), (a \rightarrow b, 1)\}, b\rangle\rangle$		
10		$\langle x_1, close, dialogue(ai, a \rightarrow b)\rangle$		
11		$\langle x_2, close, dialogue(ai, a \rightarrow b)\rangle$		
12		$\langle x_1, close, dialogue(ai, b \rightarrow c)\rangle$		
13		$\langle x_2, close, dialogue(ai, b \rightarrow c)\rangle$		
14		$\langle x_1, close, dialogue(ai, c \rightarrow d)\rangle$		
15	$(c \rightarrow d, 1)$	$\langle x_2, assert, \langle\{(a, 1), (a \rightarrow b, 1), (b \rightarrow c, 1), (c \rightarrow d, 1)\}, d\rangle\rangle$		
16		$\langle x_2, close, dialogue(ai, c \rightarrow d)\rangle$		
17		$\langle x_1, close, dialogue(ai, c \rightarrow d)\rangle$		

Table 4.8: Argument inquiry dialogue example 7.

t	$CS^t_{x_1}$	m_t	$CS^t_{x_2}$	QS_t
1		$\langle x_2, open, dialogue(ai, a \rightarrow b)\rangle$		$QS_1 = \{a, b\}$
2		$\langle x_1, open, dialogue(ai, b \rightarrow a)\rangle$		$QS_= \{b, a\}$
3		$\langle x_2, close, dialogue(ai, b \rightarrow a)\rangle$		
4		$\langle x_1, close, dialogue(ai, b \rightarrow a)\rangle$		
5		$\langle x_2, close, dialogue(ai, a \rightarrow b)\rangle$		
6		$\langle x_1, close, dialogue(ai, a \rightarrow b)\rangle$		

Table 4.9: Argument inquiry dialogue example 8.

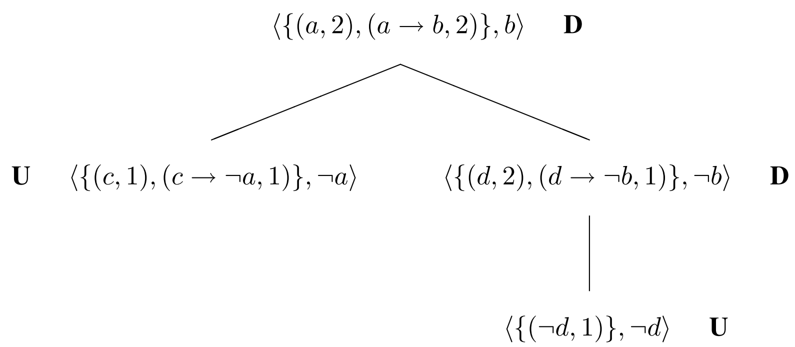


Figure 4.3: The marked dialogue tree for warrant inquiry dialogue example 1.

$$\Sigma^{x_1} = \{(c, 1), (c \rightarrow \neg a, 1), (\neg d, 1)\}$$

$$\Sigma^{x_2} = \{(a, 2), (a \rightarrow b, 2), (d, 2), (d \rightarrow \neg b, 1), (\neg c, 2)\}$$

Agent x_1 opens a warrant inquiry dialogue with b as its topic. The dialogue proceeds as in Table 4.10.

The outcome of the top-level warrant inquiry dialogue D^{16}_1 depends on the dialogue tree $\mathsf{DialogueTree}(D^{16}_1)$. The corresponding marked dialogue tree is shown in Figure 4.3.

As the root argument of the dialogue tree is defeated, the outcome of the dialogue is the empty set.

$$\mathsf{Outcome}_{wi}(D^{16}_1) = \emptyset$$

Note that agent x_2 cannot assert the argument $\langle\{(\neg c, 2)\}, \neg c\rangle$ at $t = 4$, even though it conflicts with the argument $\langle\{(c, 1), (c \rightarrow \neg a, 1)\}, \neg a\rangle$. This is because doing so would not alter the dialogue tree, as $\langle\{(\neg c, 2)\}, \neg c\rangle$ only has a preference level of 2 and so is not a defeater for $\langle\{(c, 1), (c \rightarrow \neg a, 1)\}, \neg a\rangle$.

t	$CS^t_{x_1}$	m_t	$CS^t_{x_2}$	QS_t
1		$\langle x_2, open, dialogue(wi, b)\rangle$		
2		$\langle x_1, assert, \langle\{(a, 2), (a \rightarrow b, 2)\}, b\rangle\rangle$	$(a, 2)$ $(a \rightarrow b, 2)$	
3	$(c, 1)$ $(c \rightarrow \neg a, 1)$	$\langle x_2, assert, \langle\{(c, 1), (c \rightarrow \neg a, 1)\}, \neg a\rangle\rangle$		
4		$\langle x_1, assert, \langle\{(d, 2), (d \rightarrow \neg b, 1)\}, \neg b\rangle\rangle$	$(d, 2)$ $(d \rightarrow \neg b, 1)$	
5	$(\neg d, 1)$	$\langle x_2, assert, \langle\{(\neg d, 1)\}, \neg d\rangle\rangle$		
6		$\langle x_1, open, dialogue(ai, a \rightarrow b)\rangle$		$QS_6 = \{a, b\}$
7		$\langle x_2, close, dialogue(ai, a \rightarrow b)\rangle$		
8		$\langle x_1, close, dialogue(ai, a \rightarrow b)\rangle$		
9		$\langle x_2, open, dialogue(ai, c \rightarrow \neg a)\rangle$		$QS_9 = \{c, \neg a\}$
10		$\langle x_1, close, dialogue(ai, c \rightarrow \neg a)\rangle$		
11		$\langle x_2, close, dialogue(ai, c \rightarrow \neg a)\rangle$		
12		$\langle x_1, open, dialogue(ai, d \rightarrow \neg b)\rangle$		$QS_{12} = \{a, b\}$
13		$\langle x_2, close, dialogue(ai, d \rightarrow \neg b)\rangle$		
14		$\langle x_1, close, dialogue(ai, d \rightarrow \neg b)\rangle$		
15		$\langle x_2, close, dialogue(wi, b)\rangle$		
16		$\langle x_1, close, dialogue(wi, b)\rangle$		

Table 4.10: Warrant inquiry dialogue example 1.

U $\langle\{(a, 2), (a \to b, 1)\}, b\rangle$

|

D $\langle\{(c, 1), (c \to \neg a, 1)\}, \neg a\rangle$

|

U $\langle\{(\neg c, 1)\}, \neg c\rangle$

Figure 4.4: The marked dialogue tree for warrant inquiry dialogue example 2.

Also note that there are three nested argument inquiry dialogues that appear in D_1^{16} (D_6^8, D_9^{11} and D_{12}^{14}) none of which bring any new information to either agent.

$$\mathsf{Outcome}_{ai}(D_6^8) = \emptyset$$

$$\mathsf{Outcome}_{ai}(D_9^{11}) = \emptyset$$

$$\mathsf{Outcome}_{ai}(D_{12}^{14}) = \emptyset$$

4.7.10 Warrant inquiry dialogue example 2

In the following example we have an agent x_1 who wishes to enter into a dialogue with agent x_2 in order to try to find a warrant for an argument for b. We have

$$\Sigma^{x_1} = \{(a, 2), (\neg c, 1)\} \qquad \Sigma^{x_2} = \{(a \to b, 1), (c \to \neg a, 1), (c, 1)\}$$

Agent x_1 opens a warrant inquiry dialogue with b as its topic. The dialogue proceeds as in Table 4.11.

The outcome of the top-level warrant inquiry dialogue D_1^{14} depends on the dialogue tree $\mathsf{DialogueTree}(D_1^{14})$. The corresponding marked dialogue tree is shown in Figure 4.4.

As the root node of the dialogue tree is marked as undefeated, the outcome of the dialogue D_1^{11} is the argument at the root of the tree.

$$\mathsf{Outcome}_{wi}(D_1^{14}) = \{\langle\{(a, 2), (a \to b, 1)\}, b\rangle\}$$

There are two nested argument inquiry dialogues that are sub-dialogues of D_1^{14}: D_2^6 and D_{10}^{12}.

$$\mathsf{Outcome}_{ai}(D_2^6) = \{\langle\{(a, 2), (a \to b, 1)\}, b\rangle\}$$

$$\mathsf{Outcome}_{ai}(D_2^6) = \emptyset$$

t	$CS^t_{x_1}$	m_t	$CS^t_{x_2}$	QS_t
1		$\langle x_2, open, dialogue(wi, b)\rangle$		
2		$\langle x_1, open, dialogue(ai, a \rightarrow b)\rangle$		$QS_2 = \{a, b\}$
3	$(a, 2)$	$\langle x_2, assert, \langle\{(a, 2)\}, a\rangle\rangle$		
4		$\langle x_1, assert, \langle\{(a, 2), (a \rightarrow b, 1)\}, b\rangle\rangle$	$(a, 2)$ $(a \rightarrow b, 1)$	
5		$\langle x_2, close, dialogue(ai, a \rightarrow b)\rangle$		
6		$\langle x_1, close, dialogue(ai, a \rightarrow b)\rangle$		
7		$\langle x_2, close, dialogue(wi, b)\rangle$		
8		$\langle x_1, assert, \langle\{(c, 1), (c \rightarrow \neg a, 1)\}, \neg a\rangle\rangle$	$(c, 1)$ $(c \rightarrow \neg a, 1)$	
9	$(\neg c, 1)$	$\langle x_2, assert, \langle\{(\neg c, 1)\}\neg c\rangle\rangle$		
10		$\langle x_1, open, dialogue(ai, c \rightarrow \neg a)\rangle$		$QS_{10} = \{c, \neg a\}$
11		$\langle x_2, close, dialogue(ai, c \rightarrow \neg a)\rangle$		
12		$\langle x_1, close, dialogue(ai, c \rightarrow \neg a)\rangle$		
13		$\langle x_1, close, dialogue(wi, b)\rangle$		
14		$\langle x_2, close, dialogue(wi, b)\rangle$		

Table 4.11: Warrant inquiry dialogue example 2.

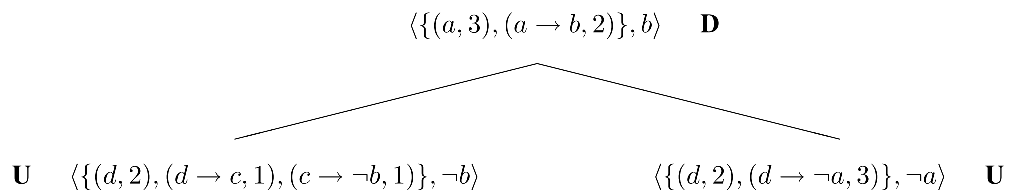


Figure 4.5: The marked dialogue tree for warrant inquiry dialogue example 3.

4.7.11 Warrant inquiry dialogue example 3

In the following example we have an agent x_1 who wishes to enter into a dialogue with agent x_2 in order to try to find a warrant for an argument for b. We have

$$\Sigma^{x_1} = \{(a, 3), (d, 2)\}$$

$$\Sigma^{x_2} = \{(a \to b, 2), (d \to c, 1), (c \to \neg b, 1), (d \to \neg a, 3)\}$$

Agent x_1 opens a warrant inquiry dialogue with b as its topic. The dialogue proceeds as in Table 4.12.

The outcome of the top-level warrant inquiry dialogue D_1^{22} depends on the dialogue tree $\mathsf{DialogueTree}(D_1^{22})$. The corresponding marked dialogue tree is shown in Figure 4.5.

As the root node of the dialogue tree is marked as defeated, the outcome of the dialogue D_1^{26} is the empty set.

$$\mathsf{Outcome}_{wi}(D_1^{26}) = \emptyset$$

There are four nested argument inquiry dialogues that are sub-dialogues of D_1^{26}: D_2^6, D_8^{18}, D_{10}^{14} and D_{22}^{24}. D_{10}^{14} is also a sub-dialogue of D_8^{18}.

$$\mathsf{Outcome}_{ai}(D_2^6) = \{\langle\{(a, 3), (a \to b, 2)\}, b\rangle\}$$

$$\mathsf{Outcome}_{ai}(D_8^{18}) = \{\langle\{(d, 2), (d \to c, 1), (c \to \neg b, 1)\}, \neg b\rangle\}$$

$$\mathsf{Outcome}_{ai}(D_{10}^{14}) = \{\langle\{(d, 2), (d \to c, 1)\}, c\rangle\}$$

$$\mathsf{Outcome}_{ai}(D_{22}^{24}) = \emptyset$$

t	$CS^t_{x_1}$	m_t	$CS^t_{x_2}$	QS_t
1		$\langle x_2, open, dialogue(wi, b)\rangle$		
2		$\langle x_1, open, dialogue(ai, a \rightarrow b)\rangle$		$QS_2 = \{a, b\}$
3	$(a, 3)$	$\langle x_2, assert, \langle\{(a, 3)\}, a\rangle\rangle$		
4		$\langle x_1, assert, \langle\{(a, 3), (a \rightarrow b, 2)\}, b\rangle\rangle$	$(a, 3)$ $(a \rightarrow b, 2)$	
5		$\langle x_2, close, dialogue(ai, a \rightarrow b)\rangle$		
6		$\langle x_1, close, dialogue(ai, a \rightarrow b)\rangle$		
7		$\langle x_2, close, dialogue(wi, b)\rangle$		
8		$\langle x_1, open, dialogue(ai, c \rightarrow \neg b)\rangle$		$QS_8 = \{c, \neg b\}$
9		$\langle x_2, close, dialogue(ai, c \rightarrow \neg b)\rangle$		
10		$\langle x_1, open, dialogue(ai, d \rightarrow c)\rangle$		$QS_{10} = \{d, c\}$
11	$(d, 2)$	$\langle x_2, assert, \langle\{(d, 2)\}, d\rangle\rangle$		
12		$\langle x_1, assert, \langle\{(d, 2), (d \rightarrow c, 1)\}, c\rangle\rangle$	$(d, 2), (d \rightarrow c, 1)$	
13		$\langle x_2, close, dialogue(ai, d \rightarrow c)\rangle$		
14		$\langle x_1, close, dialogue(ai, d \rightarrow c)\rangle$		
15		$\langle x_2, close, dialogue(ai, c \rightarrow \neg b)\rangle$		
16		$\langle x_1, assert, \langle\{(d, 2), (d \rightarrow c, 1), (c \rightarrow \neg b, 1)\}, \neg b\rangle\rangle$	$(c \rightarrow \neg b, 1)$	
17		$\langle x_2, close, dialogue(ai, c \rightarrow \neg b)\rangle$		
18		$\langle x_1, close, dialogue(ai, c \rightarrow \neg b)\rangle$		
19		$\langle x_2, close, dialogue(wi, b)\rangle$		
20		$\langle x_1, assert, \langle\{(d, 2), (d \rightarrow \neg a, 3)\}, \neg a\rangle\rangle$	$(d \rightarrow \neg a, 3)$	
21		$\langle x_2, close, dialogue(wi, b)\rangle$		
22		$\langle x_1, open, dialogue(ai, d \rightarrow \neg a)\rangle$		$QS_{22} = \{d, \neg a\}$
23		$\langle x_2, close, dialogue(ai, d \rightarrow \neg a)\rangle$		
24		$\langle x_1, close, dialogue(ai, d \rightarrow \neg a)\rangle$		
25		$\langle x_2, close, dialogue(wi, b)\rangle$		
26		$\langle x_1, close, dialogue(wi, b)\rangle$		

Table 4.12: Warrant inquiry dialogue example 3.

t	$CS^t_{x_1}$	m_t	$CS^t_{x_2}$	QS_t
1		$\langle x_2, open, dialogue(wi, b)\rangle$		
2		$\langle x_1, close, dialogue(wi, b)\rangle$		
3		$\langle x_2, assert, \langle\{(a, 2), (a \to b, 1)\}, b\rangle\rangle$	$(a, 2)$	
			$(a \to b, 1)$	
4		$\langle x_1, open, dialogue(ai, c \to \neg a)\rangle$		$QS_4 = \{c, \neg a\}$
5		$\langle x_2, assert, \langle\{(c, 1)\}, c\rangle\rangle$	$(c, 1)$	
6		$\langle x_1, assert, \langle\{(c, 1), (c \to \neg a, 1)\}, \neg a\rangle\rangle$	$(c \to \neg a, 1)$	
7		$\langle x_2, close, dialogue(ai, c \to \neg a)\rangle$		
8		$\langle x_1, close, dialogue(ai, c \to \neg a)\rangle$		
9	$(\neg c, 1)$	$\langle x_2, assert, \langle\{(\neg c, 1)\}, \neg c\rangle\rangle$		
10		$\langle x_1, open, dialogue(wi, b)\rangle$		
11		$\langle x_2, close, dialogue(wi, b)\rangle$		

Table 4.13: Warrant inquiry dialogue example 4.

4.7.12 Warrant inquiry dialogue example 4

In the following example we have an agent x_1 who wishes to enter into a dialogue with agent x_2 in order to try to find a warrant for an argument for b. We have

$$\Sigma^{x_1} = \{(a, 2), (a \to b, 1), (c, 1), (\neg c, 1)\} \qquad \Sigma^{x_2} = \{(c \to \neg a, 1)\}$$

Agent x_1 opens a warrant inquiry dialogue with b as its topic. The dialogue proceeds as in Table 4.13.

The outcome of the top-level warrant inquiry dialogue D^{11}_1 depends on the dialogue tree $\mathsf{DialogueTree}(D^{11}_1)$. The corresponding marked dialogue tree is shown in Figure 4.6.

As the root argument of the dialogue tree is undefeated, the outcome of the dialogue is the argument at the root of the tree.

$$\mathsf{Outcome}_{wi}(D^{11}_1) = \{\langle\{(a, 2), (a \to b, 1)\}, b\rangle\}$$

Note that there is a nested argument inquiry dialogue that appears as a sub-dialogue of D^{11}_1: D^8_4.

$$\mathsf{Outcome}_{ai}(D^8_4) = \{\langle\{(c, 1), (c \to \neg a, 1)\}, \neg a\rangle\}$$

Also observe that the constraints on acceptable argumentation lines (Definition 3.5.5) means that the argument $\langle\{(c, 1)\}, c\rangle$ is not added as a leaf of the dialogue tree. In fact, doing so would violate two constraints: that no argument A_k in an argumentation line is a sub-argument of an argument A_j

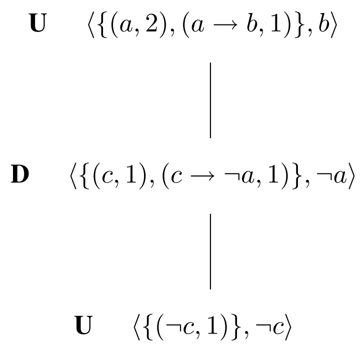


Figure 4.6: The marked dialogue tree for warrant inquiry dialogue example 4.

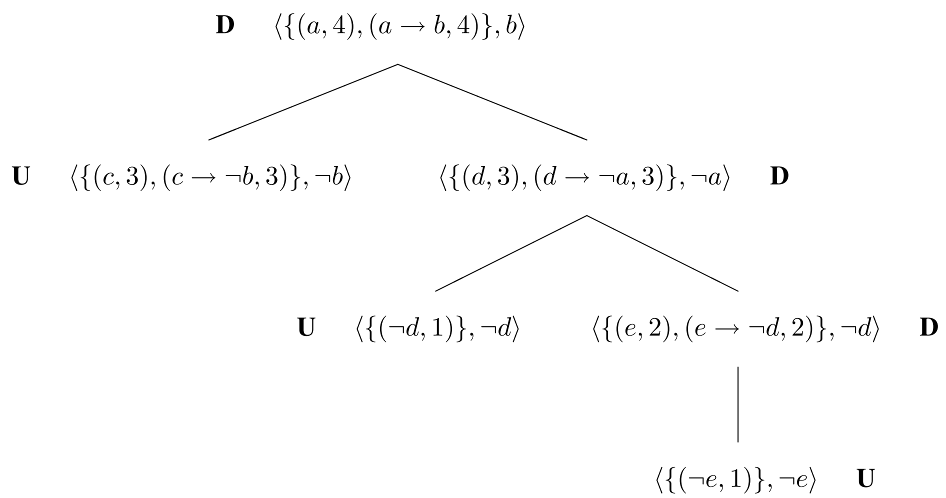


Figure 4.7: The marked dialogue tree for warrant inquiry dialogue example 5.

appearing earlier in the argumentation line ($j < k$); and that for all i, such that the argument A_i is a blocking defeater for A_{i-1}, if A_{i+1}exists then A_{i+1} is a proper defeater for A_i.

4.7.13 Warrant inquiry dialogue example 5

In the following example we have an agent x_1 who wishes to enter into a dialogue with agent x_2 in order to try to find a warrant for an argument for b. We have

$$\Sigma^{x_1} = \{(a, 4), (a \rightarrow b, 4), (c, 3), (c \rightarrow \neg b, 3), (e, 2)\}$$

$$\Sigma^{x_2} = \{(d, 3), (d \rightarrow \neg a, 3), (\neg d, 1), (e \rightarrow \neg d, 2), (\neg e, 1)\}$$

Agent x_1 opens a warrant inquiry dialogue with b as its topic. The dialogue proceeds as in Table 4.14.

The outcome of the top-level warrant inquiry dialogue D_1^{23} depends on the dialogue tree $\mathsf{DialogueTree}(D_1^{23})$. The corresponding marked dialogue tree is shown in Figure 4.7.

As the root argument of the dialogue tree is defeated, the outcome of the dialogue is the empty set.

t	$CS^t_{x_1}$	m_t	$CS^t_{x_2}$	QS_t
1		$\langle x_2, open, dialogue(wi, b)\rangle$		
2		$\langle x_1, close, dialogue(wi, b)\rangle$		
3	$(a, 4)$	$\langle x_2, assert, \langle\{(a, 4), (a \rightarrow b, 4)\}, b\rangle\rangle$		
	$(a \rightarrow b, 4)$			
4		$\langle x_1, assert, \langle\{(d, 3), (d \rightarrow \neg a, 3)\}, \neg a\rangle\rangle$	$(d, 3)$	
			$(d \rightarrow \neg a, 3)$	
5	$(c, 3)$	$\langle x_2, assert, \langle\{(c, 3), (c \rightarrow \neg b, 3)\}, \neg b\rangle\rangle$		
	$(c \rightarrow \neg b, 3)$			
6		$\langle x_1, assert, \langle\{(\neg d, 1)\}, \neg d\rangle\rangle$	$(\neg d, 3)$	
7		$\langle x_2, open, dialogue(ai, a \rightarrow b)\rangle$		$QS_7 = \{a, b\}$
8		$\langle x_1, close, dialogue(ai, a \rightarrow b)\rangle$		
9		$\langle x_2, close, dialogue(ai, a \rightarrow b)\rangle$		
10		$\langle x_1, open, dialogue(ai, d \rightarrow \neg a)\rangle$		$QS_{10} = \{d, \neg a\}$
11		$\langle x_2, close, dialogue(ai, d \rightarrow \neg a)\rangle$		
12		$\langle x_1, close, dialogue(ai, d \rightarrow \neg a)\rangle$		
13		$\langle x_2, open, dialogue(ai, c \rightarrow \neg b)\rangle$		$QS_{13} = \{c, \neg b\}$
14		$\langle x_1, close, dialogue(ai, c \rightarrow \neg b)\rangle$		
15		$\langle x_2, close, dialogue(ai, c \rightarrow \neg b)\rangle$		
16		$\langle x_1, open, dialogue(ai, e \rightarrow \neg d)\rangle$		$QS_{16} = \{e, \neg d\}$
17	$(e, 2)$	$\langle x_2, assert, \langle\{(e, 2)\}, e\rangle\rangle$		
18		$\langle x_1, assert, \langle\{(e, 2), (e \rightarrow \neg d, 2)\}, \neg d\rangle\rangle$	$(e, 2)$	
			$(e \rightarrow \neg d, 2)$	
19		$\langle x_2, close, dialogue(ai, e \rightarrow \neg d)\rangle$		
20		$\langle x_1, close, dialogue(ai, e \rightarrow \neg d)\rangle$		
21		$\langle x_2, close, dialogue(wi, b)\rangle$		
22		$\langle x_1, assert, \langle\{(\neg e, 1)\}, \neg e\rangle\rangle$	$(\neg e, 1)$	
22		$\langle x_2, close, dialogue(wi, b)\rangle$		
23		$\langle x_1, close, dialogue(wi, b)\rangle$		

Table 4.14: Warrant inquiry dialogue example 5.

$$\mathsf{Outcome}_{wi}(D_1^{23}) = \emptyset$$

Note that there are four nested argument inquiry dialogue that appears as sub-dialogues of D_1^{23}: D_7^9, D_{10}^{12}, D_{13}^{15} and D_{16}^{20}.

$$\mathsf{Outcome}_{ai}(D_7^9) = \emptyset$$

$$\mathsf{Outcome}_{ai}(D_{10}^{12}) = \emptyset$$

$$\mathsf{Outcome}_{ai}(D_{13}^{15}) = \emptyset$$

$$\mathsf{Outcome}_{ai}(D_{16}^{20}) = \{\langle\{(e, 2), (e \rightarrow \neg d, 2)\}, \neg d\rangle\}$$

4.8 Summary

In this chapter I have formally defined my novel dialogue system. I have provided protocols for the argument inquiry dialogue and the warrant inquiry dialogue, and proposed a strategy to be used to generate such dialogues, which I call the exhaustive strategy. I have provided several examples of dialogues generated by this strategy. In the next chapter, I will give an analysis of my system and the dialogues that the exhaustive strategy produces.

Chapter 5

Analysis of dialogue system with exhaustive strategy

In this chapter I give results about the dialogue system that I have defined and, in particular, about dialogues produced by the exhaustive strategy (i.e. well-formed exhaustive dialogues). In the first section, I discuss results relating to the system in general that hold regardless of what strategy is being followed (i.e. hold for all well-formed dialogues), regarding the sets of arguments that can be formed during a dialogue. In the later sections, I consider the specific dialogue behaviour when the participating agents are following the exhaustive strategy. I give results about the commitment stores generated during a dialogue produced by two agents following the exhaustive strategy, results about the moves that get made in such a dialogue, and results about the dialogue trees produced by such dialogues. Finally, I define soundness and completeness properties, and show that all well-formed exhaustive dialogues produced by my system are sound and complete. Recall, if I refer to a well-formed exhaustive dialogue then the reader is to assume that I am referring to either a well-formed argument inquiry dialogue that has been generated by the exhaustive strategy or a well-formed warrant inquiry dialogue that has been generated by the exhaustive strategy (Definition 4.5.5).

The reader should note that some of the lemmas included in this chapter are very simple (i.e. follow directly from definitions) but they are included here as they are useful building blocks for the main results that come later in this chapter.

5.1 Results about arguments

This section gives results about relationships between sets of arguments produced by the argumentation system defined in Chapter 3.

The first lemma states that if we have a set Υ that is a subset of a set of beliefs Ψ, then the set of arguments that can be constructed from Υ is a subset of the set of arguments that can be constructed from Ψ.

Lemma 5.1.1 *Let $\Upsilon \subseteq \mathcal{B}$ and $\Psi \subseteq \mathcal{B}$ be two sets. If $\Upsilon \subseteq \Psi$, then $\mathcal{A}(\Upsilon) \subseteq \mathcal{A}(\Psi)$.*

Proof: *Let us assume that $\Upsilon \subseteq \Psi$ and $\langle \Phi, \phi \rangle$ is an argument such that $\langle \Phi, \phi \rangle \in \mathcal{A}(\Upsilon)$. From the definition of an argument (Definition 3.2.1), we see that $\Phi \subseteq \Upsilon$. As $\Phi \subseteq \Upsilon$ and $\Upsilon \subseteq \Psi$, and the* *subset relationship is transitive, we get* $\Phi \subseteq \Psi$*. Hence, we see (from the definition of an argument, Definition 3.2.1) that* $\langle \Phi, \phi \rangle \in \mathcal{A}(\Psi)$*. Hence, if* $\Upsilon \subseteq \Psi$ *and* $A \in \mathcal{A}(\Upsilon)$*, then* $A \in \mathcal{A}(\Psi)$*. Hence, if* $\Upsilon \subseteq \Psi$*, then* $\mathcal{A}(\Upsilon) \subseteq \mathcal{A}(\Psi)$*.* $\square$

The next lemma states that if a set of beliefs is finite, then the set of defeasible facts that can be defeasibly derived from that set is also finite.

Lemma 5.1.2 *If* $\Psi \subseteq \mathcal{B}$ *is a finite set, then the set returned by* $\mathsf{DefDerivations}(\Psi)$ *is also finite.*
Proof: *From the definition of* DefDerivations *(Definition 3.1.9) and the definition of defeasible derivation (Definition 3.1.8), we see that if* $\phi \in \mathsf{DefDerivations}(\Psi)$*, then there either exists a state belief* $(\phi, L) \in \Psi$*, or there exists a domain belief* $(\alpha_1 \wedge \ldots \wedge \alpha_n \rightarrow \phi, L') \in \Psi$*. As* Ψ *is a finite set,* $\mathsf{DefDerivations}(\Psi)$ *is also finite.* $\square$

The final lemma in this section states that the set of arguments that can be constructed from a finite set is also finite.

Lemma 5.1.3 *If* $\Psi \subseteq \mathcal{B}$ *is a finite set, then the set* $\mathcal{A}(\Psi)$ *is also finite.*
Proof: *Consider the definition of an argument (Definition 3.2.1). It states that if* $\langle \Phi, \phi \rangle \in \mathcal{A}(\Psi)$*, then*

1. $\Phi \subseteq \Psi$,
2. $\Phi \mid\sim \phi$,
3. $\forall \phi, \phi'$ *s.t.* $\Phi \mid\sim \phi$ *and* $\Phi \mid\sim \phi'$*, it is not the case that* $\phi \cup \phi' \vdash \perp$ *(where* $\vdash$ *represents classical implicaion), and*
4. Φ *is minimal: there is no proper subset* Φ' *of* Φ *such that* Φ' *satisfies conditions (1), (2) and (3).*

As Ψ *is finite, there are only a finite number of sets* Φ *that satisfy condition 1. From Lemma 5.1.2, we see that the set* $\mathsf{DefDerivations}(\Phi)$ *(i.e. the set* $\{\phi \mid \Phi \mid\sim \phi\}$*, Definition 3.1.8) is finite. Hence, the set* $\mathcal{A}(\Psi)$ *is also finite.* $\square$

The results about arguments given in this section hold regardless of what strategy is being followed, as they relate only to the system for internal argumentation given in Chapter 3. In the next section I give results about commitment stores constructed during a dialogue in which both participants are following the exhaustive strategy (i.e. a well-formed exhaustive dialogue).

5.2 Results about commitment stores

This section gives results about the contents of commitment stores that are constructed during a well-formed exhaustive dialogue.

The first lemma states that if the exhaustive strategy selects a move that asserts an argument, then it will be possible to construct that argument from the union of the agent making the move's beliefs and the other participating agent's commitment store. This is clear from the definition of the exhaustive strategy (Definition 4.5.4) but is included as a lemma so it can be easily referred to in further results.

Lemma 5.2.1 *Let D_r^t be a well-formed exhaustive dialogue with participants x_1 and x_2 such that D_1^t is a top-dialogue of D_r^t. If $\Omega_{exh}(D_1^t, \overline{P}) = \langle P, assert, \langle \Phi, \phi \rangle \rangle$, then $\langle \Phi, \phi \rangle \in \mathcal{A}(\Sigma^{\overline{P}} \cup CS_P^t)$.*

Proof: *The definition of the exhaustive strategy (Definition 4.5.4) ensures that if the assert move $\langle P, assert, \langle \Phi, \phi \rangle \rangle$ is selected, then it will be the case that $\langle \Phi, \phi \rangle \in \mathcal{A}(\Sigma^{\overline{P}} \cup CS_P^t)$.* $\square$

The next lemma states that an agent's commitment store is always a subset of the union of the agents' beliefs. This is because the commitment stores are empty at the beginning of a dialogue and the only way that the commitment stores change are if an argument gets asserted, in which case the support of the argument (which must be a subset of the agent asserting the argument's beliefs) gets added to the relevant commitment store.

Lemma 5.2.2 *If D_r^t is a well-formed exhaustive dialogue with participants x_1 and x_2, then $CS_{x_1}^t \cup CS_{x_2}^t \subseteq \Sigma^{x_1} \cup \Sigma^{x_2}$.*

Proof: *Commitment stores are updated as follows (Definition 4.2.11).*

$$CS_P^t = \begin{cases} \emptyset & \textit{iff } t = 0, \\ CS_P^{t-1} \cup \Phi & \textit{iff } m_t = \langle \overline{P}, assert, \langle \Phi, \phi \rangle \rangle, \\ CS_P^{t-1} & \textit{otherwise.} \end{cases}$$

Hence, the only time that a commitment store is changed is when an agent P makes the move $\langle \overline{P}, assert, \langle \Phi, \phi \rangle \rangle$. From Lemma 5.2.1, we see that for $\langle \overline{P}, assert, \langle \Phi, \phi \rangle \rangle$ to be a move made at point $t+1$ in a dialogue, the condition $\langle \Phi, \phi \rangle \in \mathcal{A}(\Sigma^P \cup CS_{\overline{P}}^t)$ must hold, hence $\Phi \subseteq \Sigma^P \cup CS_{\overline{P}}^t$ (from the definition of an argument, Definition 3.2.1). As a commitment store is empty when $t = 0$, all elements of the commitment stores must be an element of the agents' beliefs, hence $CS_{x_1}^t \cup CS_{x_2}^t \subseteq \Sigma^{x_1} \cup \Sigma^{x_2}$. $\square$

The next lemma states that commitment stores are always finite. This is based on the assumption that an agent's belief base is finite.

Lemma 5.2.3 *Let D_r^t be a well-formed exhaustive dialogue with participants x_1 and x_2. The sets $CS_{x_1}^t$ and $CS_{x_2}^t$ are both finite.*

Proof: *From Lemma 5.2.2 we know that $CS_{x_1}^t \cup CS_{x_2}^t \subseteq \Sigma^{x_1} \cup \Sigma^{x_2}$. As the belief bases Σ^{x_1} and Σ^{x_2} are each assumed to be finite we know that the sets $CS_{x_1}^t$ and $CS_{x_2}^t$ are both finite.* $\square$

The next lemma states that commitment stores grow monotonically throughout a well-formed exhaustive dialogue. Note that this holds regardless of what strategy is being followed and is due to the fact that the only time a commitment store changes is when the support of an argument that has been asserted is added.

Lemma 5.2.4 *Let D_r^t be a well-formed exhaustive dialogue with participants x_1 and x_2. For all s such that $r \leq s \leq t$, if D_r^t extends D_r^s, then $CS_P^s \subseteq CS_P^t$ for $P \in \{x_1, x_2\}$.*

Proof: *According to the definition of commitment store update (Definition 4.2.11), $CS_P^t = \emptyset$ iff $t = 0$, else $CS_P^t = CS_P^{t-1} \cup \Phi$ iff $m_t = \langle \overline{P}, assert, \langle \Phi, \phi \rangle \rangle$, else $CS_P^t = CS_P^{t-1}$ otherwise. Hence, the only time the contents of a commitment store CS_P^t change are if $m_t = \langle \overline{P}, assert, \langle \Phi, \phi \rangle \rangle$, in which case the commitment store grows with the inclusion of Φ. As CS_P^0 is empty it must be the case that a commitment store grows monotonically throughout a dialogue, hence $CS_P^s \subseteq CS_P^t$.* $\square$

In this section, I have given results about commitment stores and the relationship between them and the participating agents' beliefs. In the next section, I give some results about moves made in a well-formed exhaustive dialogue and, in particular, define an upper bound on the moves that get made in a dialogue.

5.3 Results about moves

In this section I propose some sets that are intended to act as upper bounds on the set of moves made during a dialogue. I go on to define the sets of different types of moves made during a dialogue where the agents are following the exhaustive strategy, and show that these are subsets of the upper bounds. First, I consider assert moves.

I now define the set of possible assert moves for a well-formed dialogue, that I will go on to show acts as an upper bound on the moves made in a well-formed exhaustive dialogue. I define the set of possible assert moves as the set of all moves that assert an argument which can be constructed from the union of the two participating agents' beliefs. Note that the set of possible assert moves is defined for any well-formed dialogue, and not just one in which the participating agents are following the exhaustive strategy.

Definition 5.3.1 *Let D_r^t be a well-formed dialogue with participants x_1 and x_2. The set of* **possible assert moves** *for D_r^t is denoted* $\mathsf{PossAsserts}(D_r^t)$ *such that*

$$\mathsf{PossAsserts}(D_r^t) = \{\langle X, assert, \langle \Phi, \phi \rangle \rangle \mid \quad X \in \{x_1, x_2\} \text{ and } \langle \Phi, \phi \rangle \in \mathcal{A}(\Sigma^{x_1} \cup \Sigma^{x_2})\}$$

I will now show that the set of possible assert moves remains static for the duration of the dialogue. This rests on the assumption that an agent's beliefs remain static throughout any well-formed dialogue.

Lemma 5.3.1 *Let D_r^t be a well-formed dialogue. For all s, $1 \leq s \leq t$, if D_r^t extends D_r^s, then* $\mathsf{PossAsserts}(D_r^t) = \mathsf{PossAsserts}(D_r^s)$.

Proof: *The set of possible assert moves (Definition 5.3.1) depends on the set of participants, which is static, and the two belief bases of the participants, which are also assumed to be static. Hence, for all s, $1 \leq s \leq t$, if D_r^t extends D_r^s, then* $\mathsf{PossAsserts}(D_r^t) = \mathsf{PossAsserts}(D_r^s)$. □

The next lemma states that the set of possible assert moves for any well-formed dialogue is always finite. This rests on the assumption that an agent's beliefs are finite.

Lemma 5.3.2 *If D_r^t is a well-formed dialogue, then the set* $\mathsf{PossAsserts}(D_r^t)$ *is finite.*

Proof: *The set of possible assert moves (Definition 5.3.1) depends on the set of participants, which is finite, and the arguments that can be constructed from the union of the two belief bases of the participants, which is finite (due to the fact that belief bases are assumed to be finite and from Lemma 5.1.3). Hence, the set* $\mathsf{PossAsserts}(D_r^t)$ *is finite.* □

We will see shortly that the set of possible assert moves is an upper bound on the set of assert moves that are made during a well-formed exhaustive dialogue. That is to say, if an assert move is made during a dialogue, then it must be part of the set of possible assert moves. I first need to define the set of all assert moves that are made during a dialogue. Note that this set does not consider the move made at $t = 1$. This is deliberate, as the first move in a top-level dialogue is chosen by some assumed higher-level planning process, distinct from this dialogue system.

Definition 5.3.2 *Let D_r^t be a well-formed exhaustive dialogue with participants x_1 and x_2. The set of* **assert moves made** *during D_r^t is denoted $\mathsf{AssertsMade}_{exh}(D_r^t)$ as follows:*

$$\begin{aligned}\mathsf{AssertsMade}_{exh}(D_r^t) = \{\langle X, assert, \langle\Phi,\phi\rangle\rangle \mid \quad & X \in \{x_1, x_2\} \textit{ and either} \\ & \textit{if } r \neq 1\textit{, then } \langle X, assert, \langle\Phi,\phi\rangle\rangle \textit{ appears in the sequence } D_r^t \\ & \textit{else, if } r = 1\textit{, then } \langle X, assert, \langle\Phi,\phi\rangle\rangle \textit{ appears in the sequence } D_2^t\}\end{aligned}$$

I now show that the set of possible assert moves is an upper bound on the set of assert moves that are made during a well-formed exhaustive dialogue. That is to say, if an assert move gets made in such a dialogue, then it is part of the set of possible asserts for that dialogue.

Lemma 5.3.3 *If D_r^t is a well-formed exhaustive dialogue with participants x_1 and x_2, then $\mathsf{AssertsMade}_{exh}(D_r^t) \subseteq \mathsf{PossAsserts}(D_r^t)$.*

Proof: *Let us assume that $\langle P, assert, \langle\Phi,\phi\rangle\rangle \in \mathsf{AssertsMade}_{exh}(D_r^t)$, hence, from the definition of assert moves made (Definition 5.3.2), $\langle P, assert, \langle\Phi,\phi\rangle\rangle$ appears either in the sequence D_r^t if $r \neq 1$ or in the sequence D_2^t if $r = 1$. Hence, according to the dialogue behaviour algorithm (Figure 4.2), if D_1^t is a top-dialogue of D_r^t, then $\langle P, assert, \langle\Phi,\phi\rangle\rangle = \Omega_{exh}(D_1^s, \overline{P})$, for some s, where $r - 1 \leq s < t$ and D_1^t extends D_1^s. From Lemma 5.2.1, we get that $\langle\Phi,\phi\rangle \in \mathcal{A}(\Sigma^{\overline{P}} \cup CS_P^s)$. From Lemma 5.2.2 and Lemma 5.1.1, we get that $\langle\Phi,\phi\rangle \in \mathcal{A}(\Sigma^{\overline{P}} \cup \Sigma^P)$. Hence, from the definition of the set of possible asserts (Definition 5.3.1), $\langle P, assert, \Phi, \phi\rangle\rangle \in \mathsf{PossAsserts}(D_r^t)$. Hence, $\mathsf{AssertsMade}_{exh}(D_r^t) \subseteq \mathsf{PossAsserts}(D_r^t)$.* □

I now show that the set of assert moves that are made during a well-formed exhaustive dialogue is finite. This is clear because the set of assert moves made is a subset of the possible assert moves (Lemma 5.3.3), which I have already shown to be finite (Lemma 5.3.2).

Lemma 5.3.4 *If D_r^t is a well-formed exhaustive dialogue, then the set $\mathsf{AssertsMade}_{exh}(D_r^t)$ is finite.*

Proof: *This follows from Lemma 5.3.2 and Lemma 5.3.3.* □

I define the set of all possible open moves for a well-formed dialogue in a similar manner. I define the set of possible open moves as the set of all open moves that have as their content a defeasible rule from one or other of the participating agent's beliefs. Again, this set is defined for all well-formed dialogues, and not just exhaustive ones.

Definition 5.3.3 *Let D_r^t be a well-formed dialogue with participants x_1 and x_2. The set of* **possible open moves** *for D_r^t is denoted* $\mathsf{PossOpens}(D_r^t)$ *such that*

$$\mathsf{PossOpens}(D_r^t) = \{\langle X, open, dialogue(ai, \alpha_1 \wedge \ldots \wedge \alpha_n \rightarrow \beta)\rangle \mid X \in \{x_1, x_2\} \text{ and there exists } L \in \mathbb{N} \text{ such that } (\alpha_1 \wedge \ldots \wedge \alpha_n \rightarrow \beta, L) \in \Sigma^{x_1} \cup \Sigma^{x_2}\}$$

I will now show that the set of possible open moves remains static for the duration of any well-formed dialogue. This is based on the assumption that an agent's beliefs do not change during a dialogue.

Lemma 5.3.5 *Let D_r^t be a well-formed dialogue. For all s, $1 \leq s \leq t$, if D_r^t extends D_r^s then* $\mathsf{PossOpens}(D_r^t) = \mathsf{PossOpens}(D_r^s)$.

Proof: *The set of possible open moves (Definition 5.3.3) depends on the set of participants, which is static, and the two belief bases of the participants, which are also assumed to be static. Hence, for all s, $1 \leq s \leq t$, if D_r^t extends D_r^s then* $\mathsf{PossOpens}(D_r^t) = \mathsf{PossOpens}(D_r^s)$. □

The next lemma states that the set of possible open moves is always finite. This is based on the assumption that an agent's beliefs are finite.

Lemma 5.3.6 *If D_r^t is a well-formed dialogue, then the set* $\mathsf{PossOpens}(D_r^t)$ *is finite.*

Proof: *The set of possible open moves (Definition 5.3.3) depends on the set of participants, which is finite, and the two belief bases of the participants, which are also assumed to be finite. Hence, the set* $\mathsf{PossOpens}(D_r^t)$ *is finite.* □

We will see shortly that the set of possible open moves is an upper bound on the set of open moves that are made during a well-formed exhaustive dialogue. That is to say, if an open move is made during such a dialogue, then it must be part of the set of possible open moves. I first define the set of all open moves that are made during a well-formed exhaustive dialogue. Note that, like with the set of assert moves made during such a dialogue, I do not include the move made at $t = 1$, as this is assumed to be selected by some higher-level planning process external to this system.

Definition 5.3.4 *Let D_r^t be a well-formed exhaustive dialogue with participants x_1 and x_2. The set of* **open moves made** *during D_r^t is denoted* $\mathsf{OpensMade}_{exh}(D_r^t)$ *such that*

$$\mathsf{OpensMade}_{exh}(D_r^t) = \{\langle X, open, dialogue(ai, \gamma)\rangle \mid X \in \{x_1, x_2\} \text{ and either if } r \neq 1, \text{ then } \langle X, open, dialogue(ai, \gamma)\rangle \text{ appears in the sequence } D_r^t \text{ else, if } r = 1, \text{ then } \langle X, open, dialogue(ai, \gamma)\rangle \text{ appears in the sequence } D_2^t\}$$

I now show that the set of possible open moves is an upper bound on the set of open moves that are made during a well-formed exhaustive dialogue. This follows from the definition of the exhaustive strategy (Definition 4.5.4).

Lemma 5.3.7 *If D_r^t is a well-formed exhaustive dialogue with participants x_1 and x_2, then* $\mathsf{OpensMade}_{exh}(D_r^t) \subseteq \mathsf{PossOpens}(D_r^t)$.

Proof: *Let us assume that $\langle P, open, dialogue(ai, \gamma)\rangle \in \mathsf{OpensMade}_{exh}(D_r^t)$, hence, from the definition of open moves made (Definition 5.3.4), $\langle P, open, dialogue(ai, \gamma)\rangle$ either appears in the sequence D_r^t if $r \neq 1$ else it appears in the sequence D_2^t if $r = 1$. Hence, according to the dialogue behaviour algorithm (Figure 4.2), if D_1^t is a top-dialogue of D_r^t, then $\langle P, open, dialogue(ai, \gamma)\rangle = \Omega_{exh}(D_1^s, \overline{P})$, for some s, where $r-1 \leq s < t$ and D_1^t extends D_1^s. From the definition of the exhaustive strategy (Definition 4.5.4), we get that there exists $L \in \mathbb{N}$ such that $(\gamma, L) \in \Sigma^{\overline{P}}$, hence $(\gamma, L) \in \Sigma^P \cup \Sigma^{\overline{P}}$. Hence, from the definition of the set of possible opens (Definition 5.3.3), $\langle P, open, dialogue(ai, \gamma)\rangle \in \mathsf{PossOpens}(D_r^t)$. Hence,* $\mathsf{OpensMade}_{exh}(D_r^t) \subseteq \mathsf{PossOpens}(D_r^t)$. $\Box$

I now show that the set of open moves that are made during a well-formed exhaustive dialogue is finite. This is clear, as I have shown that the set of open moves made is a subset of the set of possible open moves (Lemma 5.3.7), which I have shown to be finite (Lemma 5.3.6).

Lemma 5.3.8 *If D_r^t is a well-formed exhaustive dialogue, then the set* $\mathsf{OpensMade}_{exh}(D_r^t)$ *is finite.*

Proof: *This follows from Lemma 5.3.6 and Lemma 5.3.7.* $\Box$

I now define the set of all possible moves for a well-formed dialogue. This set consists of the possible assert moves, the possible open moves, and the relevant close moves. I will go on to show that this set is an upper bound on the moves that get made during a well-formed exhaustive dialogue.

Definition 5.3.5 *Let D_r^t be a well-formed dialogue with participants x_1 and x_2. The set of* **possible moves** *for D_r^t is denoted* $\mathsf{PossMoves}(D_r^t)$ *such that*

$$\begin{aligned}\mathsf{PossMoves}(D_r^t) = \quad & \{\langle X, close, dialogue(\theta, \gamma)\rangle \mid X \in \{x_1, x_2\}, \text{ and there exists a move} \\ & \quad \langle X', open, dialogue(\theta, \gamma)\rangle \text{ that appears in the sequence } D_r^t \\ & \quad \text{where } X' \in \{x_1, x_2\}\} \\ & \cup \mathsf{PossAsserts}(D_r^t) \cup \mathsf{PossOpens}(D_r^t)\end{aligned}$$

Note that it is not the case that the set of possible moves remains static throughout a dialogue. This is because the close moves that appear as part of this set depend on the open moves already made, hence, as embedded sub-dialogues are opened, this set will grow.

The next lemma states that the set of possible moves that can be made in any well-formed dialogue is always finite. This is based on the assumption that an agent's belief base is finite.

Lemma 5.3.9 *If D_r^t is a well-formed dialogue with participants x_1 and x_2, then the set* $\mathsf{PossMoves}(D_r^t)$ *is finite.*

Proof: *Consider the set taken from the definition of possible moves (Definition 5.3.5),*

$$\begin{aligned}\{\langle X, close, dialogue(\theta, \gamma)\rangle \mid \quad & X \in \{x_1, x_2\}, \text{ and there exists a move} \\ & \langle X', open, dialogue(\theta, \gamma)\rangle \text{ that appears in the sequence } D_r^t \\ & \text{where } X' \in \{x_1, x_2\}\}\end{aligned}$$

I have shown that the set of open moves made in a dialogue not including the move at $t = 1$ is finite (Lemma 5.3.8), hence there are a finite number of such moves $\langle X', open, dialogue(\theta, \gamma)\rangle$ (including the one move made at $t = 1$). Hence, the set above is finite. We know, from Lemma 5.3.2 and Lemma 5.3.6, that both the sets $\mathsf{PossAsserts}(D_r^t)$ *and* $\mathsf{PossOpens}(D_r^t)$ *are finite. Hence, from the definition of possible moves (Definition 5.3.5), the set* $\mathsf{PossMoves}(D_r^t)$ *is finite.* □

I now define the set of moves made in a well-formed exhaustive dialogue. Note again that I am not considering the move made at $t = 1$, as this is selected by some higher-level process that is beyond the scope of this work.

Definition 5.3.6 *Let D_r^t be a well-formed exhaustive dialogue with participants x_1 and x_2. The set of* **moves made** *during D_r^t is denoted* $\mathsf{MovesMade}_{exh}(D_r^t)$ *such that*

$$\mathsf{MovesMade}_{exh}(D_r^t) = \{m \mid \text{ if } r \neq 1, \text{ then } m \text{ appears in the sequence } D_r^t, \\ \text{else, if } r = 1, \text{ then } m \text{ appears in the sequence } D_2^t\}$$

I will now show that the set of possible moves is an upper bound on the set of moves made in a well-formed exhaustive dialogue. That is to say, if a move appears in the dialogue D_r^t, then it will be part of the set of possible moves for D_r^t.

Lemma 5.3.10 *If D_r^t is a well-formed exhaustive dialogue with participants x_1 and x_2, then* $\mathsf{MovesMade}_{exh}(D_r^t) \subseteq \mathsf{PossMoves}(D_r^t)$.

Proof: *Let us assume that $m \in$ $\mathsf{MovesMade}_{exh}(D_r^t)$, hence, from the definition of moves made (Definition 5.3.6), m appears either in the sequence D_r^t if $r \neq 1$ or in the sequence D_2^t if $r = 1$. Hence, according to the dialogue behaviour algorithm (Figure 4.2), if D_1^t is a top-dialogue of D_r^t, then $m = \Omega_{exh}(D_1^s, \overline{P})$ for some s, where $r - 1 \leq s < t$ and D_1^t extends D_1^s (so m is the move made at s). From the definition of the exhaustive strategy (Definition 4.5.4), we see there are three cases to consider.*
(Case 1) $m = \langle P, assert, \langle \Phi, \phi\rangle\rangle$ such that $\langle \Phi, \phi\rangle \in \mathcal{A}(\Sigma^P \cup CS_{\overline{P}}^{s-1})$, hence $m \in$ $\mathsf{PossAsserts}(D_r^t)$ *(from the definition of possible asserts, Definition 5.3.1), hence $m \in$* $\mathsf{PossMoves}(D_r^t)$ *(from the definition of possible moves, Definition 5.3.5).*
(Case 2) $m = \langle P, open, dialogue(ai, \gamma)\rangle$ such that there exists $L \in \mathbb{N}$ such that $(\gamma, L) \in \Sigma^P$, hence $m \in$ $\mathsf{PossOpens}(D_r^t)$ *(from the definition of possible opens, Definition 5.3.3), hence $m \in$* $\mathsf{PossMoves}(D_r^t)$ *(from the definition of possible moves, Definition 5.3.5).*
(Case 3) $m = \langle P, close, dialogue(\theta, \gamma)\rangle$ where $\theta =$ $\mathsf{cType}(D_1^{s-1})$ and $\gamma =$ $\mathsf{cTopic}(D_1^{s-1})$, hence (from the definition of the current dialogue, Definition 4.2.9), there must exist s' such that $r \leq s' < s$ and $m_{s'} = \langle P', open, dialogue(\theta, \gamma)\rangle$ where $P' \in \{x_1, x_2\}$. Hence (from the definition of possible moves, Definition 5.3.5), $m \in$ $\mathsf{PossMoves}(D_r^t)$.
Hence, it follows from cases 1, 2 and 3 that $\mathsf{MovesMade}_{exh}(D_r^t) \subseteq \mathsf{PossMoves}(D_r^t)$. □

I now show that the set of moves that are made during a well-formed exhaustive dialogue is finite. This is clear, as I have shown that the set of moves made is a subset of the set of possible moves (Lemma 5.3.10), which I have shown to be finite (Lemma 5.3.9).

Lemma 5.3.11 *If D_r^t is a well-formed exhaustive dialogue, then the set* $\mathsf{MovesMade}_{exh}(D_r^t)$ *is finite.*
Proof: *This follows from Lemma 5.3.9 and Lemma 5.3.10.* □

In this section, I have defined sets that act as upper bounds on the moves made during a well-formed exhaustive dialogue. I have shown that these sets are finite, and so the set of moves made in a well-formed exhaustive dialogue is also finite. This does not necessarily mean that a dialogue terminates though. Although I have shown that the set of moves made is finite, it may be the case that moves get repeated or that a dialogue does not end with a matched-close. In the next section, I will show that all dialogues do indeed terminate and that it is not the case that moves get repeated.

5.4 Results about termination of dialogues

I will shortly show that all well-formed exhaustive dialogues terminate, but in order to do so I must first introduce some further lemmas.

The following lemma shows that a maximum of one assert move with a certain content appears in any top-level well-formed exhaustive dialogue. That is to say, the move $m_s = \langle P, assert, \langle \Phi, \phi \rangle \rangle$ (where $P \in \{x_1, x_2\}$) and the move $m_{s'} = \langle P', assert, \langle \Phi, \phi \rangle \rangle$ (where $P' \in \{x_1, x_2\}$) both appear in dialogue D_r^t if and only if $s = s'$. This is clear from the definition of the warrant inquiry and argument inquiry protocols. The warrant inquiry protocol contains a constraint that something may only be asserted if it changes the dialogue tree, which would not occur if the argument had been previously asserted. The argument inquiry protocol contains a constraint that something may only be asserted if its support is not present in the union of the commitment stores, which it would be if the argument had previously been asserted.

Lemma 5.4.1 *Let D_r^t be a well-formed exhaustive dialogue with participants x_1 and x_2. If D_1^t is a top-dialogue of D_r^t and $m_s = \langle P, assert, \langle \Phi, \phi \rangle \rangle$ for some s where $1 < s \leq t$ and $P \in \{x_1, x_2\}$ and m_s appears in the sequence D_1^t, then there does not exist an s' such that $1 < s' \leq t$, $s \neq s'$, $m_{s'} = \langle P', assert, \langle \Phi, \phi \rangle \rangle$ where $P' \in \{x_1, x_2\}$ and $m_{s'}$ appears in D_1^t.*
Proof by contradiction: *Let us assume that D_1^t is a top-dialogue of D_r^t and there does exist an s' such that $1 < s' \leq t$, $s \neq s'$ and $m_{s'} = \langle P', assert, \langle \Phi, \phi \rangle \rangle$ where $P' \in \{x_1, x_2\}$. From the definition of the exhaustive strategy (Definition 4.5.4) we see that either (1) $\langle P, assert, \langle \Phi, \phi \rangle \rangle \in \Pi_{ai}^{assert}(D_1^{s-1}, \overline{P})$, or (2) $\langle P, assert, \langle \Phi, \phi \rangle \rangle \in \Pi_{wi}^{assert}(D_1^{s-1}, \overline{P})$.*
(Case 1) According to the argument inquiry protocol and the definition of Π_{ai}^{assert} (Definition 4.3.2), $\Phi \not\subseteq CS_P^{s'-1} \cup CS_{\overline{P}}^{s'-1}$. Hence, it cannot be the case that $s < s'$ as if that were so then $\Phi \subseteq CS_{\overline{P}}^{s'-1}$ would be true (from the definition of commitment store update, Definition 4.2.11, and the fact that commitment stores grow monotonically, Lemma 5.2.4).
It also cannot be the case that $s' < s$, as if that were so then $\Phi \subseteq CS_{P'}^{s'}$ (Definition 4.2.11), and so $\Phi \subseteq CS_{P'}^{s}$ (Lemma 5.2.4), and so $\Phi \subseteq CS_{\overline{P}}^{s} \cup CS_P^{s}$ (as $P \in \{x_1, x_2\}$ and $P' \in \{x_1, x_2\}$). This would mean that it is not the case that $\langle P, assert, \langle \Phi, \phi \rangle \rangle \in \Pi_{ai}^{assert}(D_1^{s-1}, \overline{P})$ (from the definition of the argument inquiry protocol, Definition 4.3.2). Hence, it must be the case that $s = s'$, which contradicts our assumption.

(Case 2) According to the warrant inquiry protocol and the definition of Π_{wi}^{assert} (Definition 4.4.3), $\mathsf{DialogueTree}(D_1^{s'-1} + \langle P', assert, \langle\Phi, \phi\rangle\rangle) \neq \mathsf{DialogueTree}(D_1^{s'-1})$*. Hence, it cannot be the case that* $s < s'$ *as if this were true then it would be true that* $\Phi \subseteq CS_P^{s'-1} \cup CS_{\overline{P}}^{s'-1}$ *(from Definition 4.2.11 and Lemma 5.2.4), which would mean that asserting* $\langle\Phi, \phi\rangle$ *would not have an effect on the dialogue tree and it would not be the case that* $\mathsf{DialogueTree}(D_1^{s'-1} + \langle P', assert, \langle\Phi, \phi\rangle\rangle) \neq \mathsf{DialogueTree}(D_1^{s'-1})$ *(as the dialogue tree depends only on the root argument, which does not change, and the content of the commitment stores, Definition 4.4.2). It also cannot be the case that* $s' < s$*, as then* $\Phi \subseteq CS_P^{s-1} \cup CS_{\overline{P}}^{s-1}$ *(from Definition 4.2.11 and Lemma 5.2.4), and so making the move* m_s *would not alter the commitment stores and so would not alter the dialogue tree. Hence, it must be the case that* $s = s'$*, which contradicts our assumption.* □

The next lemma is similar to the last lemma but concerns open moves. It states that a maximum of one open move with a certain content appears in any top-level well-formed exhaustive dialogue. That is to say, the move $m_s = \langle P, open, dialogue(ai, \gamma)\rangle$ (where $P \in \{x_1, x_2\}$) and the move $m_{s'} = \langle P', open, dialogue(ai, \gamma)\rangle$ (where $P' \in \{x_1, x_2\}$) both appear in dialogue D_r^t if and only if $s = s'$. Again, this is clear from the definition of the warrant inquiry and argument inquiry protocols. Both these protocols contain the constraint that a move opening an argument inquiry dialogue with the topic $\alpha_1 \wedge \ldots \wedge \alpha_n \rightarrow \beta$ can only be made if a question store with the content $\{\alpha_1, \ldots, \alpha_n, \beta\}$ has not previously been constructed, which it would have been if a move opening an argument inquiry dialogue with the topic $\alpha_1 \wedge \ldots \wedge \alpha_n \rightarrow \beta$ had previously been made.

Lemma 5.4.2 *Let* D_r^t *be a well-formed exhaustive dialogue with participants* x_1 *and* x_2*. If* D_1^t *is a top-dialogue of* D_r^t *and* $m_s = \langle P, open, dialogue(ai, \alpha_1 \wedge \ldots \wedge \alpha_n \rightarrow \beta)\rangle$ *for some* s *where* $1 < s \leq t$ *and* $P \in \{x_1, x_2\}$ *and* m_s *appears in the sequence* D_1^t*, then there does not exist an* s' *such that* $1 < s' \leq t$ *and* $s \neq s'$ *and* $m_{s'} = \langle P', open, dialogue(ai, \alpha_1 \wedge \ldots \wedge \alpha_n \rightarrow \beta)\rangle$ *where* $P' \in \{x_1, x_2\}$ *and* $m_{s'}$ *appears in* D_1^t*.*

Proof: *From the definition of the exhaustive strategy (Definition 4.5.4) we see that either (1)* $\langle P, open, dialogue(ai, \alpha_1 \wedge \ldots \wedge \alpha_n \rightarrow \beta)\rangle \in \Pi_{ai}^{open}(D_1^{s-1}, \overline{P})$ *or (2)* $\langle P, open, dialogue(ai, \alpha_1 \wedge \ldots \wedge \alpha_n \rightarrow \beta)\rangle \in \Pi_{wi}^{open}(D_1^{s-1}, \overline{P})$*.*

Combining these cases we get that $\langle P, open, dialogue(ai, \alpha_1 \wedge \ldots \wedge \alpha_n \rightarrow \beta)\rangle \in \Pi_{ai}^{open}(D_1^{s-1}, \overline{P}) \cup \Pi_{wi}^{open}(D_1^{s-1}, \overline{P})$*. According to the argument inquiry protocol and the warrant inquiry protocol and the definitions of* Π_{ai}^{open} *and* Π_{wi}^{open} *(Definition 4.3.2 and Definition 4.4.3), there does not exist* t'*,* $1 < t' \leq s-1$*, such that* $QS_{t'} = \{\alpha_1, \ldots, \alpha_n, \beta\}$*. Hence, from the definition of a question store (Definition 4.3.1), there does not exist* t'*,* $1 < t' < s$*, such that* $m_{t'} = \langle P', open, dialogue(ai, \alpha_1 \wedge \ldots \wedge \alpha_n \rightarrow \beta)\rangle$ *where* $P' \in \{x_1, x_2\}$*. It also cannot be the case that there exists* s' *such that* $s < s' \leq t$ *and* $m_{s'} = \langle P', open, dialogue(ai, \alpha_1 \wedge \ldots \wedge \alpha_n \rightarrow \beta)\rangle$ *where* $P' \in \{x_1, x_2\}$ *as* $QS_s = \{\alpha_1, \ldots, \alpha_n, \beta\}$ *and* $1 < s < s'$ *which violates a condition of both the warrant inquiry protocol and the argument inquiry dialogue (Definition 4.3.2 and Definition 4.4.3).* □

I am now able to prove the theorem that all well-formed exhaustive dialogues in my system terminate. This follows from the fact that there are always a finite number of assert and open moves from which the participating agents can choose, that the agents cannot repeat these moves, and that the exhaustive strategy dictates that if an agent cannot make an assert or open move then it must make a move to close the current dialogue.

Theorem 5.4.1 *If D_r^t is a well-formed exhaustive dialogue with participants x_1 and x_2 and D_1^t is a top-dialogue of D_r^t, then there exists t' such that $t \leq t'$, $D_r^{t'}$ extends D_r^t, $D_1^{t'}$ is a top-dialogue of $D_r^{t'}$, $D_1^{t'}$ extends D_1^t and $D_r^{t'}$ terminates at t'.*

Proof: *The set of assert moves made during a well-formed exhaustive dialogue, $\mathsf{AssertsMade}_{exh}(D_r^{t'})$, is finite (Lemma 5.3.4). The set of open moves made during a well-formed exhaustive dialogue, $\mathsf{OpensMade}_{exh}(D_r^{t'})$, is finite (Lemma 5.3.8). Neither assert nor open moves may be repeated in a dialogue (Lemma 5.4.1 and Lemma 5.4.2), hence there must come a point in every dialogue where no more open or assert moves may be made. The exhaustive strategy (Definition 4.5.4) dictates that when there are no more assert or open moves that can be made, the participating agents must each make a move to close the current dialogue (called a matched close, Definition 4.2.7). When this occurs the dialogue terminates (Definition 4.2.8).* □

Finally in this section, I show that if a dialogue terminates at t, then the subset of the set of legal moves from which an agent must select the move m_t does not include any open or assert moves. This is clear from the definition of the exhaustive strategy (Definition 4.5.4), which states that a close move may only be made if there are no assert or open moves to choose from.

Lemma 5.4.3 *If D_r^t is a well-formed exhaustive dialogue that terminates at t with participants x_1 and x_2, such that $\mathsf{Receiver}(m_{t-1}) = P$, D_r^t extends D_r^{t-1} and D_1^{t-1} is a top-dialogue of D_r^{t-1}, then the set $\mathsf{Asserts}_{exh}(D_1^{t-1}, P) = \emptyset$ and the set $\mathsf{Opens}_{exh}(D_1^{t-1}, P) = \emptyset$.*

Proof: *A dialogue is terminated with a matched-close (Definition 4.2.8). The exhaustive strategy (Definition 4.5.4) states that a close move will only be made by P at timepoint t if the sets $\mathsf{Asserts}_{exh}(D_1^{t-1}, P)$ and $\mathsf{Opens}_{exh}(D_1^{t-1}, P)$ are empty. Hence, $\mathsf{Asserts}_{exh}(D_1^{t-1}, P) = \emptyset$ and $\mathsf{Opens}_{exh}(D_1^{t-1}, P) = \emptyset$.* □

In this section, I have shown that all well-formed dialogues generated by the exhaustive strategy terminate. In the next section I will define soundness and completeness properties of the argument inquiry dialogue, and show that all argument inquiry dialogues produced by the exhaustive strategy are sound and complete.

5.5 Results about soundness and completeness of argument inquiry dialogues

In this section I consider a benchmark against which to compare my dialogues and use this to define soundness and completeness properties for argument inquiry dialogues. I go on to prove that all well-formed argument inquiry dialogues generated by the exhaustive strategy are sound and complete.

I believe that it is important to consider soundness and completeness properties if we are to understand the behaviour of dialogues. This is something that most other dialogue systems miss out on, as their lack of a mechanism for selecting exactly one move to make means that they can only model different legal dialogues and cannot guarantee a certain outcome in a given situation (which I believe is a crucial requirement for a dialogue system that is to be used in a medical system such as CREDO). One notable example of similar work that does consider soundness and completeness properties is [60], in which Sadri *et al.* define different agent programs for negotiation. If such an agent program is exhaustive and deterministic, then exactly one move is suggested by the program at a timepoint, making such a program generative and allowing Sadri *et al.* to consider soundness and completeness properties. Since no other formal inquiry dialogue systems provide a specific strategy, they miss the chance to better understand the dialogue behaviour by considering such properties.

In order to consider such properties, I must define a benchmark against which to compare the outcome of my dialogues. To my knowledge, the only other similar work that defines an equivalent benchmark is that of Sadri *et al.* [60]. As they are concerned with the specific problem of negotiating resource reallocation, their benchmark relates to the existence of a solution to this problem. As I am concerned with the safety-critical, cooperative medical domain, I want my agents to arrive at the best dialogue outcome that they can, regardless of how their beliefs are split between them. For this reason, I use the arguments that can be constructed from the union of the two agents' beliefs as a benchmark, and compare these arguments to the outcome of an argument inquiry dialogue in order to decide whether the dialogue is sound and complete.

The goal of an argument inquiry dialogue is for two agents to share appropriate parts of their knowledge in order to try to construct an argument for a specific claim or claims. The benchmark that I compare the outcome of the dialogue with is the set of arguments that can be constructed from the union of the two agents' beliefs. So this benchmark is, in a sense, the "ideal" situation where there is clearly no constraint on the sharing of beliefs.

I say that an argument inquiry dialogue is sound if and only if, when the outcome of the dialogue includes an argument, then that same argument can be constructed from the union of the two participating agents' beliefs. Note that this definition of soundness holds for all well-formed argument inquiry dialogues, and not just those generated by the exhaustive strategy.

Definition 5.5.1 *Let D_r^t be a well-formed argument inquiry dialogue with participants x_1 and x_2. D_r^t is* **sound** *iff, if $\langle \Phi, \phi \rangle \in \mathsf{Outcome}_{ai}(D_r^t)$, then $\langle \Phi, \phi \rangle \in \mathcal{A}(\Sigma^{x_1} \cup \Sigma^{x_2})$.*

I now show that all well-formed exhaustive argument inquiry dialogues are sound. This is clear from the definition of argument inquiry outcome (Definition 4.3.4).

Theorem 5.5.1 *If D_r^t is a well-formed exhaustive argument inquiry dialogue with participants x_1 and x_2, then D_r^t is sound.*

Proof: *Let us assume that $\langle \Phi, \phi \rangle \in \mathsf{Outcome}_{ai}(D_r^t)$. From the definition of argument inquiry outcome (Definition 4.3.4) $\langle \Phi, \phi \rangle \in \mathcal{A}(CS_{x_1}^t \cup CS_{x_2}^t)$. Hence, from Lemma 5.1.1 and Lemma 5.2.2, $\langle \Phi, \phi \rangle \in \mathcal{A}(\Sigma^{x_1} \cup \Sigma^{x_2})$. Hence, from the definition of argument inquiry soundness (Definition 5.5.1), D_r^t is sound.*
□

Similarly, an argument inquiry dialogue is complete if and only if, if the dialogue terminates at t and it is possible to construct an argument for the consequent of the topic of the dialogue from the union of the two participating agents' beliefs, then that argument will be in the outcome of the dialogue at t. Again this definition holds for all well-formed argument inquiry dialogues, and not just exhaustive ones.

Definition 5.5.2 *Let D_r^t be a well-formed argument inquiry dialogue with participants x_1 and x_2. D_r^t is* **complete** *iff, if D_r^t terminates at t and $\mathsf{Topic}(D_r^t) = \alpha_1 \wedge \ldots \wedge \alpha_n \rightarrow \phi$ and there exists Φ such that $\langle \Phi, \phi \rangle \in \mathcal{A}(\Sigma^{x_1} \cup \Sigma^{x_2})$, then $\langle \Phi, \phi \rangle \in \mathsf{Outcome}_{ai}(D_r^t)$.*

Before I show that argument inquiry dialogues are complete, I need to introduce some further lemmas. The first says that if two agents are participating in a well-formed exhaustive argument inquiry dialogue D_r^t that terminates at t such that $\phi \in QS_r$ and there is an argument for ϕ of the form $\langle \{(\phi, L)\}, \phi \rangle$ that can be constructed from the union of the two agents' beliefs, then (ϕ, L) will be in the union of the commitment stores at timepoint t.

Lemma 5.5.1 *If D_r^t is a well-formed exhaustive argument inquiry dialogue that terminates at t, with participants x_1 and x_2, such that D_1^t is a top-dialogue of D_r^t, $\phi \in QS_r$ and there exists $\langle \Phi, \phi \rangle \in \mathcal{A}(\Sigma^{x_1} \cup \Sigma^{x_2})$ such that $\Phi = \{(\phi, L)\}$, then $(\phi, L) \in CS_{x_1}^t \cup CS_{x_2}^t$.*

Proof: *Let us assume that D_1^t is a top-dialogue of D_r^t, $\phi \in QS_r$ and $\langle \{(\phi, L)\}, \phi \rangle \in \mathcal{A}(\Sigma^{x_1} \cup \Sigma^{x_2})$, hence either $(\phi, L) \in \Sigma^{x_1}$ or $(\phi, L) \in \Sigma^{x_2}$ (from the definition of an argument, Definition 3.2.1). Hence, for either $P = x_1$ or $P = x_2$, $\langle \{(\phi, L)\}, \phi \rangle \in \mathcal{A}(\Sigma^P)$. Hence, from Lemma 5.1.1, $\langle \{(\phi, L)\}, \phi \rangle \in \mathcal{A}(\Sigma^P \cup CS_{\overline{P}}^{t_2})$ for all values of t_2, $(1 \leq t_2 \leq t)$. From the definition of the argument inquiry protocol (Definition 4.3.2) and the definition of the exhaustive strategy (Definition 4.5.4) we get that for all t_3, $r < t_3 < t$, $\langle \overline{P}, assert, \langle \{(\phi, L)\}, \phi \rangle \rangle \in \mathsf{Asserts}_{exh}(D_1^{t_3}, P)$, where D_1^t extends $D_1^{t_3}$, unless $\{(\phi, L)\} \subseteq CS_{x_1}^{t_3} \cup CS_{x_2}^{t_3}$. As D_r^t terminates at t and from Lemma 5.4.3 we get that $\mathsf{Asserts}_{exh}(D_1^t, P) = \emptyset$, hence there must exist t_4, $(1 < t_4 < t)$, such that $\{(\phi, L)\} \subseteq CS_{x_1}^{t_4} \cup CS_{x_2}^{t_4}$. As commitment stores grow monotonically (Lemma 5.2.4), $(\phi, L) \in CS_{x_1}^t \cup CS_{x_2}^t$.* □

The next lemma states that if there is a defeasible rule whose consequent is present in the question store, then there will be a timepoint at which a question store will be created that contains all the literals of the antecedent of the defeasible rule.

Lemma 5.5.2 *If D_r^t is a well-formed exhaustive argument inquiry dialogue that terminates at t, with participants x_1 and x_2, such that D_1^t is a top-dialogue of D_r^t, $\phi \in QS_r$ and there exists a domain belief $(\alpha_1 \wedge \ldots \wedge \alpha_n \rightarrow \phi, L) \in \Sigma^{x_1} \cup \Sigma^{x_2}$, then there exists t_1, $1 < t_1 < t$, such that $QS_{t_1} = \{\alpha_1, \ldots, \alpha_n, \phi\}$.*

Proof: *Let us assume that D_1^t is a top-dialogue of D_r^t, $(\alpha_1 \wedge \ldots \wedge \alpha_n \rightarrow \phi, L) \in \Sigma^P$, where P is either x_1 or x_2, and $\phi \in QS_r$. If $QS_r = \{\alpha_1, \ldots, \alpha_n, \phi\}$ then this proof is trivially true, so let us also assume that $QS_r \neq \{\alpha_1, \ldots, \alpha_n, \phi\}$. From the definition of the argument inquiry protocol (Definition 4.3.2) and the definition of the exhaustive strategy (Definition 4.5.4), we get that for all t_2, $r < t_2 < t$, $\langle \overline{P}, open, dialogue(ai, \alpha_1 \wedge \ldots \wedge \alpha_n \rightarrow \phi) \rangle \in \mathsf{Opens}_{exh}(D_1^{t_2}, P)$ (where D_1^t extends $D_1^{t_2}$) unless there exists t_3 such that $1 < t_3 \leq t_2 < t$ and $QS_{t_3} = \{\alpha_1, \ldots, \alpha_n, \phi\}$. As D_r^t terminates* *at t and from Lemma 5.4.3,* $\mathsf{Opens}_{exh}(D_1^t, P) = \emptyset$*, hence there must exist* t_4*,* $1 < t_4 < t$*, such that* $QS_{t_4} = \{\alpha_1, \ldots, \alpha_n, \phi\}$*.* □

I am now able to use the two previous lemmas to show that all argument inquiry dialogues are complete.

Theorem 5.5.2 *If* D_r^t *is a well-formed exhaustive argument inquiry dialogue with participants* x_1 *and* x_2*, then* D_r^t *is complete.*

Proof: *If* D_r^t *does not terminate at t then* D_r^t *is complete. So let us assume* D_r^t *terminates at t,* $\phi \in QS_r$*, and* $\langle \Phi, \phi \rangle \in \mathcal{A}(\Sigma^{x_1} \cup \Sigma^{x_2})$*. From the definition of an argument (Definition 3.2.1),* $\Phi \subseteq \Sigma^{x_1} \cup \Sigma^{x_2}$*. There are two cases to consider.*

(Case 1) $\Phi = \{(\phi, L)\}$*. Hence from Lemma 5.5.1,* $(\phi, L) \in CS_{x_1}^t \cup CS_{x_2}^t$*. From Definition 4.3.4,* $\langle \Phi, \phi \rangle \in \mathsf{Outcome}_{ai}(D_r^t)$*.*

(Case 2) There exists a domain belief $(\alpha_1 \wedge \ldots \wedge \alpha_n \rightarrow \phi, L) \in \Phi$*. From the definition of an argument (Definition 3.2.1), for all* α_i *where* $1 \leq i \leq n$*, there exists* Φ_i *such that* $\langle \Phi_i, \alpha_i \rangle \in \mathcal{A}(\Sigma^{x_1} \cup \Sigma^{x_2})$*. From Lemma 5.5.2, there exists* t_1*,* $1 < t_1 \leq t$*, such that* $QS_{t_1} = \{\alpha_1, \ldots, \alpha_n, \phi\}$*. Each* Φ_i *is either an example of case 1 or case 2 and* Φ *is finite, so, by recursion,* $\exists r_2, t_2$*,* $r < r_2 < t_2 \leq t$*, such that* $\langle \Phi_i, \alpha_i \rangle \in \mathsf{Outcome}_{ai}(D_{r_2}^{t_2})$*. Hence, from the definition of an argument (Definition 3.2.1),* $\langle \Phi, \phi \rangle \in \mathsf{Outcome}_{ai}(D_r^t)$*.*

From case 1, case 2 and the definition of argument inquiry completeness (Defintion 5.5.2), D_r^t *is complete.* □

The theorem above becomes particularly interesting when I combine it with the theorem that all well-formed exhaustive dialogues terminate. This gives the desired result that if an argument can be constructed from the union of the two participating agents' beliefs whose claim is the consequent of the topic of the current dialogue, then there will come a timepoint at which that argument is in the outcome of that dialogue.

Theorem 5.5.3 *Let* D_r^t *be a well-formed exhaustive argument inquiry dialogue with participants* x_1 *and* x_2*. If* $\mathsf{Topic}(D_r^t) = \alpha_1 \wedge \ldots \wedge \alpha_n \rightarrow \phi$ *and there exists* Φ *such that* $\langle \Phi, \phi \rangle \in \mathcal{A}(\Sigma^{x_1} \cup \Sigma^{x_2})$ *then there exists* t_1*,* $1 < t \leq t_1$*, such that* $D_r^{t_1}$ *extends* D_r^t *and* $\langle \Phi, \phi \rangle \in \mathsf{Outcome}(D_r^{t_1})$*.*

Proof: *This follows from Theorem 5.4.1 and Theorem 5.5.2.* □

I have shown that all well-formed exhaustive argument inquiry dialogues generated by the exhaustive strategy are sound and complete. Before I can consider soundness and completeness of warrant inquiry dialogues I must give some results about dialogue trees, which I will do in the next section.

5.6 Results about dialogue trees

Before I can show that all warrant inquiry dialogues generated by the exhaustive strategy are sound and complete, I must give some results about the dialogue trees that they produce. I consider the relationship between a dialogue tree that is produced by a well-formed exhaustive warrant inquiry dialogue and the dialectical tree that is constructed from the union of the two participating agents' beliefs and has the same argument at the root. In particular, I will go on to show that these two trees are the same.

The following lemma states that if we have a warrant inquiry dialogue D_r^t that terminates at t, whose root argument is $\langle \Phi, \phi \rangle$, and there is a path from the root node $[\langle \Phi, \phi \rangle, \langle \Phi_1, \phi_1 \rangle, \ldots, \langle \Phi_n, \phi_n \rangle]$ that appears in the dialogue tree $\mathsf{DialogueTree}(D_r^t)$, then the same path from the root node appears in the dialectical tree that is constructed from the union of the two participating agents' beliefs and has $\langle \Phi, \phi \rangle$ at its root. This is due to the relationship between the commitment stores and the agents' beliefs.

Lemma 5.6.1 *Let D_r^t be a well-formed exhaustive warrant inquiry dialogue that terminates at t with participants x_1 and x_2 such that $\mathsf{RootArg}(D_r^t) = \langle \Phi, \phi \rangle$. If there exists a path $[\langle \Phi, \phi \rangle, \langle \Phi_1, \phi_1 \rangle, \ldots, \langle \Phi_n, \phi_n \rangle]$ in $\mathsf{DialogueTree}(D_r^t)$, then the same path exists in the dialectical tree T_A^Δ where $A = \langle \Phi, \phi \rangle$ and $\Delta = \Sigma^{x_1} \cup \Sigma^{x_2}$.*

Proof: *Let us assume that the path $[\langle \Phi, \phi \rangle, \langle \Phi_1, \phi_1 \rangle, \ldots, \langle \Phi_n, \phi_n \rangle]$ appears in $\mathsf{DialogueTree}(D_r^t)$. From the definition of a dialogue tree (Definition 4.4.2) we see that this means that the path $[\langle \Phi, \phi \rangle, \langle \Phi_1, \phi_1 \rangle, \ldots, \langle \Phi_n, \phi_n \rangle]$ appears in the dialectical tree $\mathsf{T}_A^{\Delta'}$ where $\Delta' = CS_{x_1}^t \cup CS_{x_2}^t$. Hence, for $1 \leq i \leq n$, $\langle \Phi_i, \phi_i \rangle \in \mathcal{A}(CS_{x_1}^t \cup CS_{x_2}^t)$ (from the definition of a dialectical tree, Definition 3.6.1), hence $\langle \Phi_i, \phi_i \rangle \in \mathcal{A}(\Sigma^{x_1} \cup \Sigma^{x_2})$ (from Lemma 5.2.2 and Lemma 5.1.1), hence the path $[\langle \Phi, \phi \rangle, \langle \Phi_1, \phi_1 \rangle, \ldots, \langle \Phi_n, \phi_n \rangle]$ can also be constructed from Δ. As $[\langle \Phi, \phi \rangle, \langle \Phi_1, \phi_1 \rangle, \ldots, \langle \Phi_n, \phi_n \rangle]$ is an acceptable argumentation line in $\mathsf{DialogueTree}(D_r^t)$, it must also be an acceptable argumentation line in T_A^Δ (from the definition of an acceptable argumentation line, Definition 3.5.5). Hence, if there exists a path $[\langle \Phi, \phi \rangle, \langle \Phi_1, \phi_1 \rangle, \ldots, \langle \Phi_n, \phi_n \rangle]$ in $\mathsf{DialogueTree}(D_r^t)$, then there exists a path $[\langle \Phi, \phi \rangle, \langle \Phi_1, \phi_1 \rangle, \ldots, \langle \Phi_n, \phi_n \rangle]$ in T_A^Δ.* $\square$

The next lemma is the reverse of the previous one. It states that if we have a warrant inquiry dialogue D_r^t that terminates at t, whose root argument is $\langle \Phi, \phi \rangle$, and there is a path $[\langle \Phi, \phi \rangle, \langle \Phi_1, \phi_1 \rangle, \ldots, \langle \Phi_n, \phi_n \rangle]$ that appears in the dialectical tree that is constructed from the union of the two participating agents' beliefs and has $\langle \Phi, \phi \rangle$ at its root, then the same path $[\langle \Phi, \phi \rangle, \langle \Phi_1, \phi_1 \rangle, \ldots, \langle \Phi_n, \phi_n \rangle]$ appears in $\mathsf{DialogueTree}(D_r^t)$. This is due to the fact that the warrant inquiry protocol along with the exhaustive strategy ensures that all arguments that change the dialogue tree (i.e. cause a new node to be added to the tree) get asserted during the dialogue (Definition 4.4.3: $\mathsf{DialogueTree}(D_1^t + \langle \overline{P}, assert, \langle \Phi, \phi \rangle \rangle) \neq \mathsf{DialogueTree}(D_1^t)$).

Lemma 5.6.2 *Let D_r^t be a well-formed exhaustive warrant inquiry dialogue that terminates at t with participants x_1 and x_2 such that $\mathsf{RootArg}(D_r^t) = \langle \Phi, \phi \rangle$. If D_1^t is a top-dialogue of D_r^t and there exists a path $[\langle \Phi, \phi \rangle, \langle \Phi_1, \phi_1 \rangle, \ldots, \langle \Phi_n, \phi_n \rangle]$ in the dialectical tree T_A^Δ, where $A = \langle \Phi, \phi \rangle$ and $\Delta = \Sigma^{x_1} \cup \Sigma^{x_2}$, then there exists a path $[\langle \Phi, \phi \rangle, \langle \Phi_1, \phi_1 \rangle, \ldots, \langle \Phi_n, \phi_n \rangle]$ in $\mathsf{DialogueTree}(D_r^t)$.*

Proof: *Let us assume that the path $[\langle \Phi, \phi \rangle, \langle \Phi_1, \phi_1 \rangle, \ldots, \langle \Phi_n, \phi_n \rangle]$ appears in the dialectical tree T_A^Δ. Let us also assume that there exists t_1 such that $\Phi \subseteq CS_{x_1}^{t_1} \cup CS_{x_2}^{t_1}$ and there does not exist t' such that $1 < t' < t_1$ and $\Phi \subseteq CS_{x_1}^{t'} \cup CS_{x_2}^{t'}$ (i.e. t_1 is the timepoint at which the root argument is asserted). According to the definition of a dialectical tree (Definition 3.6.1) this means that for all i, $1 \leq i \leq n$, $\langle \Phi_i, \phi_i \rangle \in \mathcal{A}(\Sigma^{x_1} \cup \Sigma^{x_2})$. There are two cases.*

(Case 1) $\langle \Phi_i, \phi_i \rangle \in \mathcal{A}(\Sigma^P)$ *where* $P = x_1$ *or* $P = x_2$.

(Case 2) $\langle \Phi_i, \phi_i \rangle \notin \mathcal{A}(\Sigma^{x_1})$ *and* $\langle \Phi_i, \phi_i \rangle \notin \mathcal{A}(\Sigma^{x_2})$, *in which case there exists a defeasible rule* $(\alpha_1 \wedge \ldots \wedge \alpha_m \rightarrow \phi_i, L) \in \Phi_i$ *such that* $(\alpha_1 \wedge \ldots \wedge \alpha_m \rightarrow \phi_i, L) \in \Sigma^P$, *where* $P = x_1$ *or* $P = x_2$.

Let us now consider $\langle \Phi_1, \phi_1 \rangle$. *It is either an instance of case 1, or of case 2.*

If it is case 1: From the definition of the warrant inquiry protocol and the definition of the exhaustive strategy (Definitions 4.4.3 and 4.5.4), for all t_2 *such that* $t_1 < t_2 \leq t$ *and* D_1^t *extends* $D_1^{t_2}$, $\langle \overline{P}, assert, \langle \Phi_1, \phi_1 \rangle \rangle \in \mathsf{Asserts}_{exh}(D_1^{t_2}, P)$ *unless* $\mathsf{DialogueTree}(D_1^{t_2} + \langle \overline{P}, assert, \langle \Phi_1, \phi_1 \rangle \rangle) \neq \mathsf{DialogueTree}(D_1^{t_2})$ *(i.e making the move to assert* $\langle \Phi_1, \phi_1 \rangle$ *does not change the dialogue tree).*

As the path $[\langle \Phi, \phi \rangle, \langle \Phi_1, \phi_1 \rangle]$ *appears in* T_A^Δ *it must be the case that* $\langle \Phi_1, \phi_1 \rangle$ *is a defeater for* $\langle \Phi, \phi \rangle$ *and the argumentation line* $[\langle \Phi, \phi \rangle, \langle \Phi_1, \phi_1 \rangle]$ *is acceptable, from the definition of a dialectical tree (Definition 3.6.1). Hence if* $\langle \overline{P}, assert, \langle \Phi_1, \phi_1 \rangle \rangle \notin \mathsf{Asserts}_{exh}(D_1^{t_2}, P)$ *then the argumentation line* $[\langle \Phi, \phi \rangle, \langle \Phi_1, \phi_1 \rangle]$ *must already appear in* $\mathsf{DialogueTree}(D_r^{t_2})$, *otherwise asserting* $\langle \Phi, \phi \rangle$ *would certainly change the dialogue tree. As* $\mathsf{Asserts}_{exh}(D_1^{t-1}, P) = \emptyset$ *(Lemma 5.4.3) it is the case that* $\langle \overline{P}, assert, \langle \Phi_1, \phi_1 \rangle \rangle \notin \mathsf{Asserts}_{exh}(D_1^{t-1}, P)$ *and so the argumentation line* $[\langle \Phi, \phi \rangle, \langle \Phi_1, \phi_1 \rangle]$ *must already appear in* $\mathsf{DialogueTree}(D_r^{t-1})$. *Hence, there must exist* t_3 *such that* $t_1 < t_3 < t$ *and* $[\langle \Phi, \phi \rangle, \langle \Phi_1, \phi_1 \rangle]$ *appears in* $\mathsf{DialogueTree}(D_r^{t_3})$.

If it is case 2: As $\langle \Phi_1, \phi_1 \rangle$ *is a defeater for* $\langle \Phi, \phi \rangle$, $\neg\phi_1 \in \mathsf{DefDerivations}(CS_{x_1}^{t_1} \cup CS_{x_2}^{t_1})$. *Hence, from the definition of the warrant inquiry protocol and the definition of the exhaustive strategy (Definitions 4.4.3 and 4.5.4), for all* t_4 *such that* $t_1 < t_4 \leq t$, $\langle \overline{P}, open, dialogue(ai, \alpha_1 \wedge \ldots \wedge \alpha_m \rightarrow \phi_1) \rangle \in \mathsf{Opens}_{exh}(D_1^{t_4}, P)$ *unless there exists a* t_5 *such that* $1 < t_5 \leq t_4$ *and* $QS_{t_5} = \{\alpha_1, \ldots, \alpha_m, \phi_1\}$, *in which case, from Theorem 5.5.3, there exists* t_6 *such that* $t_5 \leq t_6 < t$ *and* $\langle \Phi_1, \phi_1 \rangle \in \mathsf{Outcome}_{ai}(D_r^{t_6})$. *As* $\mathsf{Opens}_{exh}(D_1^{t-1}, P) = \emptyset$ *(Lemma 5.4.3), it must be the case that there exists* t_6 *such that* $t_5 \leq t_6 < t$ *and* $\langle \Phi_1, \phi_1 \rangle \in \mathsf{Outcome}_{ai}(D_r^{t_6})$. *From the definition of argument inquiry outcome (Definition 4.3.4),* $\langle \Phi_1, \phi_1 \rangle \in CS_{x_1}^{t_6} \cup CS_{x_2}^{t_6}$, *hence, from the definition of a dialectical tree (Definition 3.6.1),* $[\langle \Phi, \phi \rangle, \langle \Phi_1, \phi_1 \rangle]$ *appears in* $\mathsf{DialogueTree}(D_r^{t_6})$.

Now let us consider $\langle \Phi_2, \phi_2 \rangle$. *We can apply the same reasoning again to show that there will exist a timepoint at which* $[\langle \Phi, \phi \rangle, \langle \Phi_1, \phi_1 \rangle, \langle \Phi_2, \phi_2 \rangle]$ *appears in the dialogue tree. If we now consider* $\langle \Phi_3, \phi_3 \rangle$ *we can apply the same reasoning to show us that there will exist a timepoint at which* $[\langle \Phi, \phi \rangle, \langle \Phi_1, \phi_1 \rangle, \langle \Phi_2, \phi_2 \rangle, \langle \Phi_3, \phi_3 \rangle]$ *appears in the dialogue tree. Hence, if we continued in this way, by recursion, there exists a path* $[\langle \Phi, \phi \rangle, \langle \Phi_1, \phi_1 \rangle, \ldots, \langle \Phi_n, \phi_n \rangle]$ *in the dialogue tree* $\mathsf{DialogueTree}(D_r^t)$.
□

The following theorem states that if we have a well-formed exhaustive warrant inquiry dialogue D_r^t that terminates at t, whose root argument is $\langle \Phi, \phi \rangle$, then the dialogue tree $\mathsf{DialogueTree}(D_r^t)$ equals the dialectical tree that is constructed from the union of the two participating agents' beliefs and has $\langle \Phi, \phi \rangle$ as its root.

Theorem 5.6.1 *If* D_r^t *is a well-formed exhaustive warrant inquiry dialogue that terminates at* t *with participants* x_1 *and* x_2 *such that* $\mathsf{RootArg}(D_r^t) = \langle \Phi, \phi \rangle$, *then* $\mathsf{DialogueTree}(D_r^t) = \mathsf{T}_A^\Delta$ *where* $A = \langle \Phi, \phi \rangle$ *and* $\Delta = \Sigma^{x_1} \cup \Sigma^{x_2}$.

Proof: *This follows from Definition 3.6.2 (the definition of what it means for two trees to equal each other), Lemma 5.6.1 and Lemma 5.6.2.* □

I have shown that the dialogue tree constructed during a well-formed exhaustive warrant inquiry dialogue is the same as the dialectical tree constructed from the union of the agents' beliefs that has the same argument at its root as the dialogue tree. In the next section I will use this result to show that well-formed exhaustive warrant inquiry dialogues are sound and complete.

5.7 Results about soundness and completeness of warrant inquiry dialogues

The goal of a warrant inquiry dialogue is for two agents to share relevant parts of their knowledge in order to construct a dialogue tree that has an argument for the topic of the dialogue at the root. This dialogue tree then acts as a warrant for the root argument if the status of the root node is U (i.e. it is undefeated). The benchmark that I compare this to is the dialectical tree that has the same root argument as the dialogue tree but is constructed from the union of the two agents' beliefs. Again, this benchmark is, in a sense, the "ideal" situation, in which there are no constraints on the sharing of beliefs.

However, accepting this benchmark as the ideal means assuming that one agent's beliefs are as equally acceptable as the other's. An agent's knowledge is stratified internally, according to the preference levels of its beliefs, but perhaps we should also consider a stratification between agents, perhaps where one agent's beliefs on a certain topic are preferred to another's.

Let us consider the case where a recently qualified junior house officer is having a conversation with an experienced oncologist about the diagnosis of a patient with breast symptoms. The junior house officer is very confident that the patient has a cyst (so his internal preference level for the belief is high), whilst the oncologist believes that the patient has breast cancer but is only fairly confident of this belief (so her internal preference level for the belief is not high). We might expect the junior house officer to give more credence to the oncologist's opinions, as the oncologist is more expert in the domain of breast cancer, despite the oncologist not having a high confidence in her belief.

Perhaps, then, I should not only be internally stratifying the agents' beliefs but should also be weighting one agent's beliefs over another's. If one agent is considered to be more expert than another than its beliefs could be given a higher weighting. Or it may be the case that an agent is an expert in certain topics but not in others. It would be interesting to consider a way of classifying an agent's knowledge into different topics and giving each of these topics a rating depending on how expert the agent is in that topic. When an agent then asserts a belief that falls into a certain topic classification, that belief could be weighted according to whether the agent was more or less expert in that topic than the other participating agent.

My referral agent scenario contains two agents (Section 1.2.1, page 1.2.1). One of these agents contains only domain beliefs, that is knowledge relating to the guideline, the other agent contains only state beliefs, that is knowledge relating to the state of the world. As each agent contains only one type of knowledge, and each contains knowledge distinct from the other, it is reasonable to claim that each agent is the expert in its own knowledge. As each agent has only knowledge of a distinct type to that held by the other agent, there is no opportunity within the scenario for both agents to put forward things that are of the same type of knowledge. I therefore feel that it is reasonable, for this scenario at least, to compare a warrant inquiry dialogue tree to the equivalent dialectical tree constructed from the union of the participating agents' beliefs. When considering different scenarios at a later point, it would be interesting to consider different ideal inference systems with which to compare this system.

I say that a warrant inquiry dialogue is sound if and only if, when the outcome of the terminated dialogue is an argument $\langle \Phi, \phi \rangle$, the status of the root node of a dialectical tree that is constructed from the union of the participating agents' beliefs and has $\langle \Phi, \phi \rangle$ at its root is U. Note that this definition holds for all well-formed warrant inquiry dialogues, regardless of what strategy was followed.

Definition 5.7.1 *Let D_r^t be a well-formed warrant inquiry dialogue with participants x_1 and x_2. D_r^t is* **sound** *iff, if D_r^t terminates at t and $\mathsf{Outcome}_{wi}(D_r^t) = \{\langle \Phi, \phi \rangle\}$, then $\mathsf{Status}(\mathsf{Root}(\mathsf{T}_A^\Delta), \mathsf{T}_A^\Delta) = \mathsf{U}$ where $A = \langle \Phi, \phi \rangle$ and $\Delta = \Sigma^{x_1} \cup \Sigma^{x_2}$.*

I now define what it means for a warrant inquiry dialogue to be complete. I say that a warrant inquiry dialogue is complete if and only if, if the root argument of the dialogue is $\langle \Phi, \phi \rangle$ and the status of the root node of a dialectical tree that has $\langle \Phi, \phi \rangle$ at its root and is constructed from the union of the participating agents' beliefs is U, then the outcome of the dialogue when it is terminated is $\langle \Phi, \phi \rangle$. Again, this definition holds for all well-formed warrant inquiry dialogues.

Definition 5.7.2 *Let D_r^t be a well-formed warrant inquiry dialogue with participants x_1 and x_2. D_r^t is* **complete** *iff, if D_r^t terminates at t, $\mathsf{RootArg}(D_r^t) = \langle \Phi, \phi \rangle$ and $\mathsf{Status}(\mathsf{Root}(\mathsf{T}_A^\Delta), \mathsf{T}_A^\Delta) = \mathsf{U}$ where $A = \langle \Phi, \phi \rangle$ and $\Delta = \Sigma^{x_1} \cup \Sigma^{x_2}$, then $\mathsf{Outcome}_{wi}(D_r^t) = \{\langle \Phi, \phi \rangle\}$.*

I have shown that the dialogue tree produced during a warrant inquiry dialogue is the same as the dialectical tree constructed from the union of the participating agents' beliefs that has the same argument at its root. Hence, it is clear that warrant inquiry dialogues produced by the exhaustive strategy are sound.

Theorem 5.7.1 *If D_r^t is a well-formed exhaustive warrant inquiry dialogue, then D_r^t is sound.*

Proof: *If D_r^t does not terminate at t, then D_r^t is sound. So, let us assume that D_r^t terminates at t. Theorem 5.6.1 states that $\mathsf{DialogueTree}(D_r^t) = \mathsf{T}_A^\Delta$ where $A = \mathsf{RootArg}(D_r^t)$ and $\Delta = \Sigma^{x_1} \cup \Sigma^{x_2}$. If $\mathsf{Outcome}_{wi}(D_r^t) = \{\langle \Phi, \phi \rangle\}$, then $\mathsf{Status}(\mathsf{Root}(\mathsf{DialogueTree}(D_r^t)), \mathsf{DialogueTree}(D_r^t)) = \mathsf{U}$ (from the definition of warrant inquiry outcome, Definition 4.4.6), hence if $\mathsf{Outcome}_{wi}(D_r^t) = \{\langle \Phi, \phi \rangle\}$, then $\mathsf{Status}(\mathsf{Root}(\mathsf{T}_A^\Delta), \mathsf{T}_A^\Delta) = \mathsf{U}$. Hence, from the definition of warrant inquiry soundness (Definition 5.7.1), D_r^t is sound.* □

I now show that all well-formed exhaustive warrant inquiry dialogues are complete, again, based on the fact that the dialogue tree is the same as the dialectical tree constructed from the agents' beliefs with the same argument at the root.

Theorem 5.7.2 *If D_r^t is a well-formed exhaustive warrant inquiry dialogue, then D_r^t is complete.*

Proof: *If D_r^t does not terminate at t, then D_r^t is complete. So, let us assume that D_r^t terminates at* *t. Theorem 5.6.1 states that* $\mathsf{DialogueTree}(D_r^t) = \mathsf{T}_A^\Delta$ *where* $A = \mathsf{RootArg}(D_r^t)$ *and* $\Delta = \Sigma^{x_1} \cup \Sigma^{x_2}$*. Hence if* $\mathsf{Status}(\mathsf{Root}(\mathsf{T}_A^\Delta), \mathsf{T}_A^\Delta) = \mathsf{U}$*, then* $\mathsf{Outcome}_{wi}(D_r^t) = \{\langle\Phi, \phi\rangle\}$ *(from the definition of warrant inquiry outcome, Definition 4.4.6). Hence, from the definition of warrant inquiry completeness (Definition 5.7.2),* D_r^t *is complete.* □

I now combine the two previous theorems with the result that all dialogues terminate, to give us the following desired results. The first states that all warrant inquiry dialogues have a continuation such that if the outcome of this continuation is $\langle\Phi, \phi\rangle$, then the status of the root node of the dialectical tree that is constructed from the union of the agents' beliefs with $\langle\Phi, \phi\rangle$ at its root is U.

Theorem 5.7.3 *Let* D_r^t *be a well-formed exhaustive warrant inquiry dialogue with participants* x_1 *and* x_2*. There exists* t' *such that* $r < t'$*,* $D_r^{t'}$ *extends* D_r^t*, and if* $\mathsf{Outcome}_{wi}(D_r^{t'}) = \{\langle\Phi, \phi\rangle\}$*, then* $\mathsf{Status}(\mathsf{Root}(\mathsf{T}_A^\Delta), \mathsf{T}_A^\Delta) = \mathsf{U}$ *where* $A = \langle\Phi, \phi\rangle$ *and* $\Delta = \Sigma^{x_1} \cup \Sigma^{x_2}$*.*
Proof: *This follows from Theorems 5.7.1 and 5.4.1.* □

The next theorem states that all warrant inquiry dialogues have a continuation such that if the root argument of the dialogue is $\langle\Phi, \phi\rangle$ and the status of the root node of the dialectical tree that is constructed from the union of the participating agents' beliefs and has $\langle\Phi, \phi\rangle$ at its root is U, then the outcome of the continuation of the dialogue is $\langle\Phi, \phi\rangle$.

Theorem 5.7.4 *Let* D_r^t *be a well-formed exhaustive warrant inquiry dialogue with participants* x_1 *and* x_2*. There exists* t' *such that* $r < t'$*,* $D_r^{t'}$ *extends* D_r^t*, and if* $\mathsf{RootArg}(D_r^t) = \langle\Phi, \phi\rangle$ *and* $\mathsf{Status}(\mathsf{Root}(\mathsf{T}_A^\Delta), \mathsf{T}_A^\Delta) = \mathsf{U}$ *where* $A = \langle\Phi, \phi\rangle$ *and* $\Delta = \Sigma^{x_1} \cup \Sigma^{x_2}$*, then* $\mathsf{Outcome}_{wi}(D_r^t) = \{\langle\Phi, \phi\rangle\}$*.*
Proof: *This follows from Theorems 5.7.2 and 5.4.1.* □

5.8 Summary

I have shown in this chapter that dialogues generated by two agents in my system who are both following the exhaustive strategy (i.e. well-formed exhaustive dialogues) are sound and complete. In the next chapter I will introduce a new strategy called the *pruned tree strategy*. My intention when designing the pruned tree strategy was that it should be more efficient in some way than the exhaustive strategy, but that the dialogues it generates must also be sound and complete.

Chapter 6

Pruned tree strategy

In this chapter I introduce another strategy for use during a warrant inquiry dialogue, that I call the *pruned tree strategy* (unlike the exhaustive strategy, agents cannot use the pruned tree strategy in argument inquiry dialogues as well as in warrant inquiry dialogues). The motivation behind this strategy is to produce a pruned version of the dialogue tree, but to still produce sound and complete warrant inquiry dialogues. I give examples of warrant inquiry dialogues produced by this strategy later in this chapter. These examples are easily comparable with the examples of warrant inquiry dialogues produced by the exhaustive strategy given in Section 4.7, as the examples in this chapter are based on the same situations as the examples in Section 4.7 (i.e. the belief bases of the agents are the same and the topic of the dialogue is the same, only the strategy being followed changes).

Dialogue trees have the potential to be quite large. Even if the participating agents only have relatively small belief bases, it is possible that these beliefs can combine to create many interacting arguments. Some branches of a dialogue tree may be considered redundant and I will now discuss such redundancy in dialogue trees, that I intend to avoid with the use of the pruned tree strategy.

Consider the dialogue tree shown in the left of Figure 6.1. The node labelled with A_3 is defeated, as A_7 defeats this argument and there is no argument that defeats A_7. A_6 also defeats A_3 but this node has no bearing on the status of A_3 as it is itself defeated by A_9. This is an example of a type of redundancy in dialogue trees. If a node that is defeated has any children that are also defeated, then the sub-tree that has the defeated child at its root is redundant.

Now consider the node labelled with A_2. This argument is defeated by both A_4 and A_5, and both of these arguments are themselves undefeated. However, either one of A_4 or A_5 on its own is sufficient to defeat node A_2, and this is is another type of redundancy that we see in argument trees. This type of redundancy is seen if a node that is defeated has more than one child that is not defeated.

The tree on the right hand side of Figure 6.1 shows a dialogue tree that we may generate from the same set of arguments with the redundancies detailed here removed. Note that the status of the root node is the same in both trees.

Adding a child to a node that is already defeated will have no effect on the status of that node, nor on the status of any other nodes in the dialogue tree. This is something that García and Simari also considered. They observe that a marked dialectical tree resembles the *minimax* tree [22] and propose a

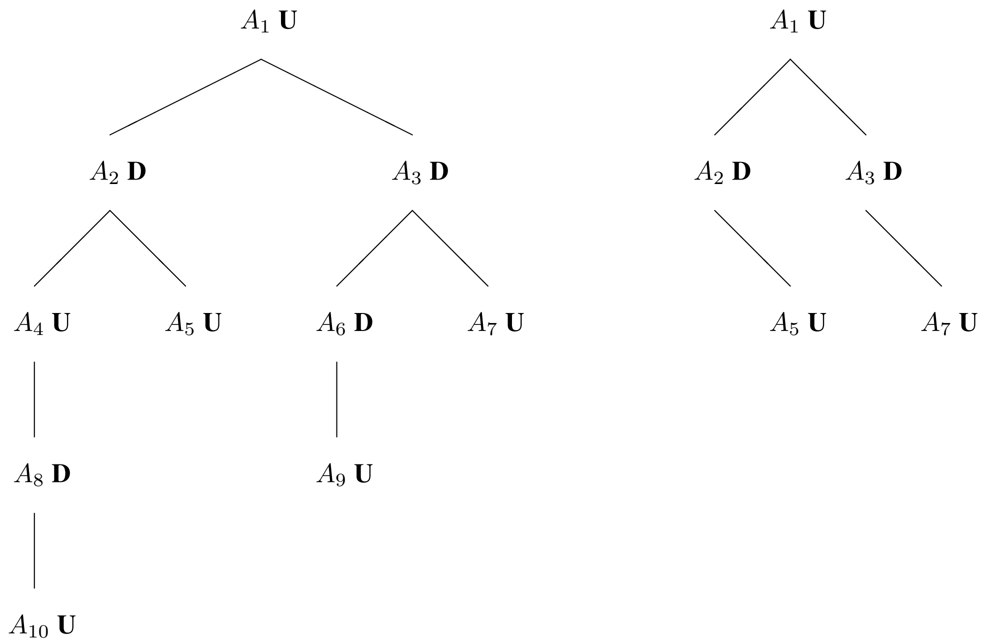


Figure 6.1: Redundancy in dialogue trees. The tree on the right is a pruned version of the tree on the left, with the redundancies discussed here removed.

PROLOG-like specification of the top-level of the warrant procedure with pruning, where the pruning process is similar to the $\alpha - \beta$ pruning of a minimax tree.

As a dialectical tree is constructed, the status of the nodes in the tree change. This means that, although adding a certain argument at some point in the construction of the tree might not affect the status of any nodes (because all the nodes that it defeats are already defeated) at another point this may no longer be the case (as the status of the nodes that it defeats may change). The reverse is also true. It may be the case that adding an argument at a certain point in the construction of a dialectical tree causes the status of a node to change. However, if that argument does not get added at that point, then it may be the case that there will never be another point at which adding it will change the status of a node, as the node whose status is changed is already defeated by some other argument.

An example of this is shown in Figure 6.2. Each of the trees in this figure are meant to represent a dialectical tree at three specific points in its construction, at which the agent has the opportunity to add the argument A_4 which defeats the argument A_2. I have shown where A_4 would go in the tree if it were added with a dotted line. In the first tree (on the left of the figure) A_2 is already defeated by A_3, which is itself undefeated, and so adding A_4 would have no effect on the status of any nodes. In the second tree (in the middle of the figure) A_2 is undefeated, as a new argument has been added, A_5, which defeats A_3. So, at this point in the construction of the tree, adding A_4 would change the status of both A_2 and A_1. In the final tree (on the right of the figure) A_2 is now back to being defeated again, as another new argument has been added, A_6, that defeats A_5. So adding A_4 to the tree at this point would again have no effect on the status of any nodes in the dialogue tree.

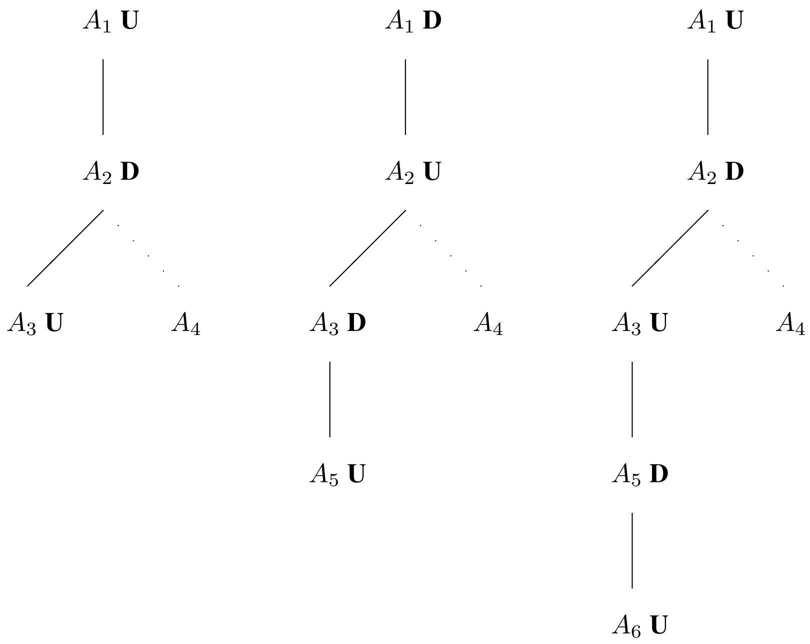


Figure 6.2: Changing node status in dialogue trees. Adding A_4 to the tree on the left or to the tree on the right does not affect the status of any node in the tree it is being added to. Adding A_4 to the tree in the middle does affect the status of existing nodes in the tree.

García and Simari's system is intended for use by a single agent. This means that an agent can construct a dialectical tree in a depth-first manner, and once it has constructed a path from the root to a leaf (for which it knows it has no acceptable defeaters) it can be sure that there will not be any more defeaters that will get added as a child of the leaf (as there is no acquisition of new information). It is then able to know which of the nodes in the path it is worth defeating, being the ones that are marked U.

As my system deals with inter-agent reasoning between two agents, the beliefs are split across the two agents, and new knowledge is acquired via the commitment stores, it is not always possible to construct dialogue trees in a depth-first manner. This is shown in the following example.

Example 6.0.1 *Let x_1 and x_2 be two agents that are participating in a well-formed exhaustive warrant inquiry dialogue.*

$$\Sigma^{x_1} = \{(a, 4), (e, 1), (\neg d, 1), (a \rightarrow b, 4), (e \rightarrow \neg b, 1)\}$$

$$\Sigma^{x_2} = \{(c, 3), (d, 2), (c \rightarrow \neg a, 3), (d \rightarrow \neg c, 2)$$

Agent x_2 opens a warrant inquiry dialogue with topic b. Informally, the dialogue progresses as follows.

- *agent x_1 asserts* $\langle\{(a, 4), (a \rightarrow b, 4)\}, b\rangle$
- *agent x_2 asserts* $\langle\{(c, 3), (c \rightarrow \neg a, 3)\}, \neg a\rangle$
- *agent x_1 asserts* $\langle\{(e, 1), (e \rightarrow \neg b, 1)\}, \neg b\rangle$
- *agent x_2 asserts* $\langle\{(d, 2), (d \rightarrow \neg c, 2)\}, \neg c\rangle$

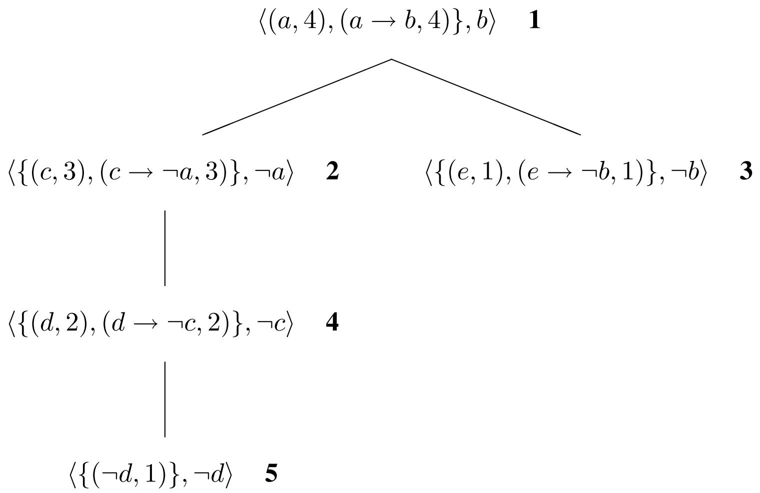


Figure 6.3: Dialogue tree from Example 6.0.1 labelled with a number representing the order in which the arguments were asserted.

- *agent* x_1 *asserts* $\langle\{(\neg d, 1)\}, \neg d\rangle$

This produces the dialogue tree shown in Figure 6.3. I have numbered the nodes to show the order the tree was constructed, which is not in a depth-first manner.

There are, then, two principles that my pruned tree strategy adopts. Informally, these are: only assert an argument if it will alter the status of a node in the tree; if there is a choice of such arguments to assert then pick the one that changes the status of a node furthest away from the root. The pruned tree strategy, like the exhaustive strategy, dictates that the agent must pick the most preferred of a constrained set of assert moves if there are any, else it must pick the most preferred of a constrained set of open moves if there are any, else it must make a close move.

The set of assert moves from which an agent must pick the most preferred are those that assert an argument which the agent can construct and which would change the status of a node in the dialogue tree. In addition to this, the level of the node whose status it would change must be at least as great as the level of any other nodes whose status may be altered by asserting an argument that the agent can construct.

The set of open moves from which an agent must pick the most preferred are those that open an argument inquiry dialogue which has as its topic a domain belief which is present in the agent's beliefs and, if an argument for the consequent of the topic were successfully found, then there exists a node in the tree, N, that is status U and that is labelled with an argument that would be in conflict with the argument for the consequent of the topic of the dialogue. In addition to this, if there are any other open moves that open an argument inquiry dialogue which has as its topic a domain belief that is present in the agent's beliefs, then the level of any node in the tree that is labelled with an argument that would be in conflict with an argument for the consequent of the topic of the dialogue must be less than or equal to the level of node N.

I will now formally define this strategy.

Definition 6.0.1 *The* **pruned tree strategy** *is a function* $\Omega_{prn} : \mathcal{D}_{top} \times \mathcal{I} \mapsto \mathcal{M}$*, where* $\mathsf{cTopic}(D_1^t) = \gamma$*,* $\mathsf{cType}(D_1^t) = wi$*,* $\mathsf{Current}(D_1^t) = D_r^t$ *and*

$$\Omega_{prn}(D_1^t, P) = \begin{cases} \mathsf{Pref_a}(\mathsf{Asserts}_{prn}(D_1^t, P)) & \textit{iff } \mathsf{Asserts}_{prn}(D_1^t, P) \neq \emptyset \\ \mathsf{Pref_o}(\mathsf{Opens}_{prn}(D_1^t, P)) & \textit{iff } \mathsf{Asserts}_{prn}(D_1^t, P) = \emptyset \textit{ and } \mathsf{Opens}_{prn}(D_1^t, P) \neq \emptyset \\ \langle P, close, dialogue(wi, \gamma)\rangle & \textit{iff } \mathsf{Asserts}_{prn}(D_1^t, P) = \emptyset \textit{ and } \mathsf{Opens}_{prn}(D_1^t, P) = \emptyset \end{cases}$$

where

$\mathsf{Asserts}_{prn}(D_1^t, P) = \{\langle \overline{P}, assert, \langle \Phi, \phi\rangle\rangle \in \Pi_{wi}^{assert}(D_1^t, P) \mid \langle \Phi, \phi\rangle \in \mathcal{A}(\Sigma^P \cup CS_{\overline{P}}^t)$

and either (1) $\phi = \gamma$ *or*

(2) there exists $N \in \mathsf{Nodes}(\mathsf{DialogueTree}(D_r^t))$ *such that*

$[\mathsf{Status}(N, \mathsf{DialogueTree}(D_r^t)) = \mathsf{U}$ *and*

$\mathsf{Status}(N, \mathsf{DialogueTree}(D_r^t + \langle \overline{P}, assert, \langle \Phi, \phi\rangle\rangle)) = \mathsf{D}$ *and*

for all $\langle \overline{P}, assert, \langle \Phi', \phi'\rangle\rangle \in \Pi_{wi}^{assert}(D_1^t, P)$ *such that*

$\langle \Phi', \phi'\rangle \in \mathcal{A}(\Sigma^P \cup CS_{\overline{P}}^t)$

[*if* $N' \in \mathsf{Nodes}(\mathsf{DialogueTree}(D_r^t))$ *such that* $\mathsf{Status}(N', \mathsf{DialogueTree}(D_r^t)) = \mathsf{U}$

and $\mathsf{Status}(N', \mathsf{DialogueTree}(D_r^t + \langle \overline{P}, assert, \langle \Phi', \phi'\rangle\rangle)) = \mathsf{D}$,

then $\mathsf{Level}(N) \geq \mathsf{Level}(N')]]\}$

$\mathsf{Opens}_{prn}(D_1^t, P) = \{\langle \overline{P}, open, dialogue(ai, \alpha_1 \wedge \ldots \wedge \alpha_n \rightarrow \beta)\rangle \in \Pi_{wi}^{open}(D_1^t, P) \mid$

there exists $L \in \mathbb{N}$ *such that* $(\alpha_1 \wedge \ldots \wedge \alpha_n \rightarrow \beta, L) \in \Sigma^P$ *and*

either (1) $\mathsf{RootArg}(D_r^t) = null$ *and* $\beta = \gamma$ *or*

(2) there exists $\langle \Phi, \phi\rangle \in \mathcal{A}(CS_P^t \cup CS_{\overline{P}}^t)$ *such that*

$[\Phi \mid\!\sim \neg\beta$ *and there exists* $N \in \mathsf{Nodes}(\mathsf{DialogueTree}(D_r^t))$ *such that*

$\mathsf{Label}(N) = \langle \Phi, \phi\rangle$ *and* $\mathsf{Status}(N, \mathsf{DialogueTree}(D_r^t)) = \mathsf{U}$

and for all $\langle \overline{P}, open, dialogue(ai, \alpha_1' \wedge \ldots \wedge \alpha_m' \rightarrow \beta')\rangle \in \Pi_{wi}^{open}(D_1^t, P)$

such that there exists $L \in \mathbb{N}$ *such that* $(\alpha_1' \wedge \ldots \wedge \alpha_m' \rightarrow \beta', L) \in \Sigma^P$

[*if there exists* $\langle \Phi', \phi'\rangle \in \mathcal{A}(CS_P^t \cup CS_{\overline{P}}^t)$ *such that* $\Phi' \mid\!\sim \neg\beta'$

and there exists $N' \in \mathsf{Nodes}(\mathsf{DialogueTree}(D_r^t))$ *such that*

$\mathsf{Label}(N') = \langle \Phi', \phi'\rangle$ *and* $\mathsf{Status}(N', \mathsf{DialogueTree}(D_r^t)) = \mathsf{U}$,

then $\mathsf{Level}(N) \geq \mathsf{Level}(N')]]\}$

The pruned tree strategy states that an agent must chose the most preferred assert move from $\mathsf{Asserts}_{prn}(D_1^t, P)$ if this set is not empty. This set only consists of legal assert moves from the set $\Pi_{wi}^{assert}(D_1^t, P)$, and so they must change the dialogue tree in some way. It also only consists of moves that assert an argument that can be constructed from agent making the move's beliefs and the other agent's commitment store. If there is not already a root argument and there is such a move that asserts an argument whose claim is the topic of the dialogue, then the set $\mathsf{Asserts}_{prn}(D_1^t, P)$ will consist of only this move. Otherwise, the set $\mathsf{Asserts}_{prn}(D_1^t, P)$ will consist of the set of such moves that assert an argument $\langle \Phi, \phi \rangle$ such that there is a node N in the dialogue tree that has status U ($\mathsf{Status}(N, \mathsf{DialogueTree}(D_r^t)) = \mathsf{U}$) and that asserting $\langle \Phi, \phi \rangle$ would cause the status of the node to change to D ($\mathsf{Status}(N, \mathsf{DialogueTree}(D_r^t + \langle P, assert, \langle \Phi, \phi \rangle \rangle)) = \mathsf{D}$). In addition to this, out of all the possible assert moves that would cause the status of a node in the dialogue tree to change from U to D, only those that would alter the status of a node at least as far away from the root of the tree as the other nodes whose status would be changed by other such assert moves will appear in the set $\mathsf{Asserts}_{prn}(D_1^t, P)$. This is because for all $\langle \overline{P}, assert, \langle \Phi', \phi' \rangle \rangle \in \Pi_{wi}^{assert}(D_1^t, P)$ such that $\langle \Phi', \phi' \rangle \in \mathcal{A}(\Sigma^P \cup CS_{\overline{P}}^t)$ [if $N' \in \mathsf{Nodes}(\mathsf{DialogueTree}(D_r^t))$ such that $\mathsf{Status}(N', \mathsf{DialogueTree}(D_r^t)) = \mathsf{U}$ and $\mathsf{Status}(N', \mathsf{DialogueTree}(D_r^t + \langle \overline{P}, assert, \langle \Phi', \phi' \rangle \rangle)) = \mathsf{D}$, then $\mathsf{Level}(N) \geq \mathsf{Level}(N')$].

If the set $\mathsf{Asserts}_{prn}(D_1^t, P)$ is empty, then the agent must chose the most preferred move from $\mathsf{Opens}_{prn}(D_1^t, P)$, as long as this set is not empty. This set only consists of legal open moves from the set $\Pi_{wi}^{open}(D_1^t, P)$, and so they cannot cause a question store to be created with the same content as one that has previously been created during the dialogue. It also only consists of moves that open an argument inquiry dialogue whose topic is present in P's beliefs. If there is not yet a root argument for the dialogue, then $\mathsf{Opens}_{prn}(D_1^t, P)$ will consist of any such open moves that open an argument inquiry dialogue whose topic is a defeasible rule, the consequent of which is the topic of the top-level warrant inquiry dialogue. Otherwise, the set $\mathsf{Opens}_{prn}(D_1^t, P)$ consists of moves that open an argument inquiry dialogue with topic $\alpha_1 \wedge \ldots \wedge \alpha_n \rightarrow \beta$ as its topic such that there is a node in the dialogue tree whose status is U and that is labelled with an argument from which you can defeasibly derive $\neg\beta$, (there exists $\langle \Phi, \phi \rangle \in \mathcal{A}(CS_P^t \cup CS_{\overline{P}}^t)$ such that [$\Phi \mid\!\sim \neg\beta$ and there exists $N \in \mathsf{Nodes}(\mathsf{DialogueTree}(D_r^t))$ such that $\mathsf{Label}(N) = \langle \Phi, \phi \rangle$ and $\mathsf{Status}(N, \mathsf{DialogueTree}(D_r^t)) = \mathsf{U}$]). In addition to this, out of all the open moves that fit this description, only the ones that conflict with a node that is at least as far away from the root node of the tree as other nodes that conflict with other such open moves appear in the set $\mathsf{Opens}_{prn}(D_1^t, P)$. This is because for all $\langle \overline{P}, open, dialogue(ai, \alpha'_1 \wedge \ldots \wedge \alpha'_m \rightarrow \beta') \rangle \in \Pi_{wi}^{open}(D_1^t, P)$ such that there exists $L \in \mathbb{N}$ such that $(\alpha'_1 \wedge \ldots \wedge \alpha'_m \rightarrow \beta', L) \in \Sigma^P$ [if there exists $\langle \Phi', \phi' \rangle \in \mathcal{A}(CS_P^t \cup CS_{\overline{P}}^t)$ such that $\Phi' \mid\!\sim \neg\beta'$ and there exists $N' \in \mathsf{Nodes}(\mathsf{DialogueTree}(D_r^t))$ such that $\mathsf{Label}(N') = \langle \Phi', \phi' \rangle$ and $\mathsf{Status}(N', \mathsf{DialogueTree}(D_r^t)) = \mathsf{U}$, then $\mathsf{Level}(N) \geq \mathsf{Level}(N')$].

If both the sets $\mathsf{Asserts}_{prn}(D_1^t, P)$ and $\mathsf{Opens}_{prn}(D_1^t, P)$ are empty, then the agent must make a close move.

I now define a well-formed pruned tree dialogue. This is a well-formed warrant inquiry dialogue that is generated by two agents who both follow the pruned tree strategy if the current dialogue is a warrant inquiry dialogue but still follow the exhaustive strategy if the current dialogue is an argument inquiry dialogue. Note that I have specified that a well-formed pruned tree dialogue must be a well-formed warrant inquiry dialogue as the pruned tree strategy can only be used for generating warrant inquiry dialogues.

Definition 6.0.2 *A* **well-formed pruned tree dialogue** *is a well-formed warrant inquiry dialogue* D_r^t *with participants* x_1 *and* x_2 *such that*

for all s *such that* $r \leq s < t$ *and* D_r^t *extends* D_r^s*,*

if D_1^t *is a top-dialogue of* D_r^t *and* D_1^s *is a top-dialogue of* D_r^s

and D_1^t *extends* D_1^s

and $\mathsf{Receiver}(m_s) = P$ *(*$P \in \{x_1, x_2\}$*) and* $\mathsf{cType}(D_r^s) = \theta$*,*

then if $\theta = wi$*, then* $\Omega_{prn}(D_1^s, P) = m_{s+1}$*,*

else if $\theta = ai$*, then* $\Omega_{exh}(D_1^s, P) = m_{s+1}$*,*

In the next section I give examples of well-formed pruned tree dialogues.

6.1 Dialogue examples

In this section I give examples of warrant inquiry dialogues that take place between two agents, x_1 and x_2, both of which follow the exhaustive strategy whilst the current dialogue is a nested argument inquiry dialogue, but follow the pruned tree strategy whilst the current dialogue is the top-level warrant inquiry dialogue. Throughout all the examples in this section, I will assume that $\mu(a) = 1$, $\mu(\neg a) = 2$, $\mu(b) = 3$, $\mu(\neg b) = 4$, $\mu(c) = 5$, $\mu(\neg c) = 6$, $\mu(d) = 7$ etc. Each example here has an equivalent example in Section 4.7, where the participating agents' beliefs and the topic of the dialogue are the same as here. This is so I can directly compare warrant inquiry dialogues generated with the exhaustive strategy with warrant inquiry dialogues generated with the pruned tree strategy.

6.1.1 Warrant inquiry dialogue example 1

This example uses the same situation as the one from Section 4.7.9, except the agents apply the pruned tree strategy whilst taking part in a warrant inquiry dialogue. In the following example we have an agent x_1 who wishes to enter into a dialogue with agent x_2 in order to try to find a warrant for an argument for b. We have

$$\Sigma^{x_1} = \{(c, 1), (c \rightarrow \neg a, 1), (\neg d, 1)\}$$

$$\Sigma^{x_2} = \{(a, 2), (a \rightarrow b, 2), (d, 2), (d \rightarrow \neg b, 1), (\neg c, 2)\}$$

Agent x_1 opens an warrant inquiry dialogue with b as its topic. The dialogue proceeds as in Table 4.10.

The outcome of the top-level warrant inquiry dialogue D_1^5 depends on the dialogue tree $\mathsf{DialogueTree}(D_1^5)$. The corresponding marked dialogue tree constructed at the end of the dialogue is shown in Figure 6.4.

As the root argument of the dialogue tree is defeated, the outcome of the dialogue is the empty set.

$$\mathsf{Outcome}_{wi}(D_1^5) = \emptyset$$

t	$CS^t_{x_1}$	m_t	$CS^t_{x_2}$	QS_t
1		$\langle x_2, open, dialogue(wi, b)\rangle$		
2		$\langle x_1, assert, \langle\{(a,2),(a \to b,2)\}, b\rangle\rangle$	$(a,2)$ $(a \to b, 2)$	
3	$(c,1)$ $(c \to \neg a, 1)$	$\langle x_2, assert, \langle\{(c,1),(c \to \neg a,1)\}, \neg a\rangle\rangle$		
4		$\langle x_1, close, dialogue(wi, b)\rangle$		
5		$\langle x_2, close, dialogue(wi, b)\rangle$		

Table 6.1: Warrant inquiry dialogue example 1.

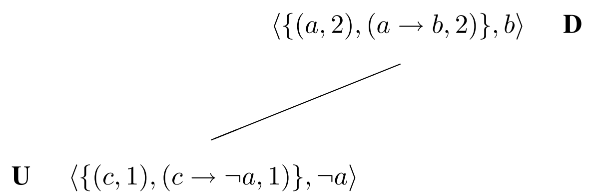


Figure 6.4: The marked dialogue tree for warrant inquiry dialogue example 1.

Note that agent x_2 cannot assert the argument $\langle\{(\neg c, 2)\}, \neg c\rangle$ at $t = 4$, even though it conflicts with the argument $\langle\{(c,1),(c \to \neg a,1)\}, \neg a\rangle$. This is because doing so would not alter the dialogue tree, as $\langle\{(\neg c, 2)\}, \neg c\rangle$ only has a preference level of 2 and so is not a defeater for $\langle\{(c,1),(c \to \neg a,1)\}, \neg a\rangle$.

Also note that the dialogue tree produced by this dialogue (Figure 6.4) has two less nodes than that produced in the equivalent dialogue using the exhaustive strategy (Figure 4.3). However, the outcomes of both dialogues are the same.

6.1.2 Warrant inquiry dialogue example 2

This example uses the same situation as the one from Section 4.7.10, except the agents apply the pruned tree strategy whilst taking part in a warrant inquiry dialogue. In the following example we have an agent x_1 who wishes to enter into a dialogue with agent x_2 in order to try to find a warrant for an argument for b. We have

$$\Sigma^{x_1} = \{(a,2), (\neg c, 1)\} \qquad \Sigma^{x_2} = \{(a \to b, 1), (c \to \neg a, 1), (c, 1)\}$$

Agent x_1 opens an warrant inquiry dialogue with b as its topic. The dialogue proceeds as in Table 6.1.2.

The outcome of the top-level warrant inquiry dialogue D_1^{14} depends on the dialogue tree $\mathsf{DialogueTree}(D_1^{14})$. The corresponding marked dialogue tree constructed at the end of the dialogue is shown in Figure 6.5.

As the root node of the dialogue tree is marked as undefeated, the outcome of the dialogue D_1^{11} is

t	$CS^t_{x_1}$	m_t	$CS^t_{x_2}$	QS_t
1		$\langle x_2, open, dialogue(wi, b)\rangle$		
2		$\langle x_1, open, dialogue(ai, a \rightarrow b)\rangle$		$QS_2 = \{a, b\}$
3	$(a, 2)$	$\langle x_2, assert, \langle\{(a, 2)\}, a\rangle\rangle$		
4		$\langle x_1, assert, \langle\{(a, 2), (a \rightarrow b, 1)\}, b\rangle\rangle$	$(a, 2)$ $(a \rightarrow b, 1)$	
5		$\langle x_2, close, dialogue(ai, a \rightarrow b)\rangle$		
6		$\langle x_1, close, dialogue(ai, a \rightarrow b)\rangle$		
7		$\langle x_2, close, dialogue(wi, b)\rangle$		
8		$\langle x_1, assert, \langle\{(c, 1), (c \rightarrow \neg a, 1)\}, \neg a\rangle\rangle$	$(c, 1)$ $(c \rightarrow \neg a, 1)$	
9	$(\neg c, 1)$	$\langle x_2, assert, \langle\{(\neg c, 1)\}\neg c\rangle\rangle$		
10		$\langle x_1, open, dialogue(ai, c \rightarrow \neg a)\rangle$		$QS_{10} = \{c, \neg a\}$
11		$\langle x_2, close, dialogue(ai, c \rightarrow \neg a)\rangle$		
12		$\langle x_1, close, dialogue(ai, c \rightarrow \neg a)\rangle$		
13		$\langle x_1, close, dialogue(wi, b)\rangle$		
14		$\langle x_2, close, dialogue(wi, b)\rangle$		

Table 6.2: Warrant inquiry dialogue example 2.

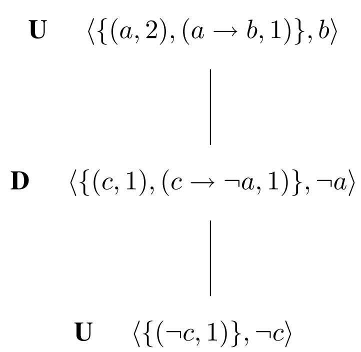


Figure 6.5: The marked dialogue tree for warrant inquiry dialogue example 2.

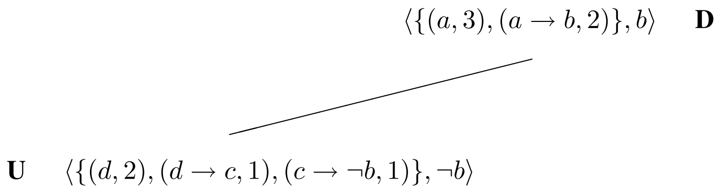


Figure 6.6: The marked dialogue tree for the warrant inquiry dialogue example 3.

the argument at the root of the tree.

$$\mathsf{Outcome}_{wi}(D_1^{14}) = \{\langle\{(a, 2), (a \rightarrow b, 1)\}, b\rangle\}$$

There are two nested argument inquiry dialogues that are sub-dialogues of D_1^{14}: D_2^6 and D_{10}^{12}.

$$\mathsf{Outcome}_{ai}(D_2^6) = \{\langle\{(a, 2), (a \rightarrow b, 1)\}, b\rangle\}$$

$$\mathsf{Outcome}_{ai}(D_2^6) = \emptyset$$

Note that the dialogue tree produced by this dialogue (Figure 6.5) is the same as that produced in the equivalent dialogue using the exhaustive strategy (Figure 4.4). In fact, both dialogues proceed in exactly the same way.

6.1.3 Warrant inquiry dialogue example 3

This example uses the same situation as the one from Section 4.7.11, except the agents apply the pruned tree strategy whilst taking part in a warrant inquiry dialogue. In the following example we have an agent x_1 who wishes to enter into a dialogue with agent x_2 in order to try to find a warrant for an argument for b. We have

$$\Sigma^{x_1} = \{(a, 3), (d, 2)\} \qquad \Sigma^{x_2} = \{(a \rightarrow b, 2), (d \rightarrow c, 1), (c \rightarrow \neg b, 1), (d \rightarrow \neg a, 3)\}$$

Agent x_1 opens an warrant inquiry dialogue with b as its topic. The dialogue proceeds as in Table 6.3.

The outcome of the top-level warrant inquiry dialogue D_1^{20} depends on the dialogue tree $\mathsf{DialogueTree}(D_1^{20})$. The corresponding marked dialogue tree constructed at the end of the dialogue is shown in Figure 6.6.

As the root node of the dialogue tree is marked as defeated, the outcome of the dialogue D_1^{26} is the empty set.

$$\mathsf{Outcome}_{wi}(D_1^{20}) = \emptyset$$

There are three nested argument inquiry dialogues that are sub-dialogues of D_1^{26}: D_2^6, D_8^{18} and D_{10}^{14}. D_{10}^{14} is also a sub-dialogue of D_8^{18}.

t	$CS^t_{x_1}$	m_t	$CS^t_{x_2}$	QS_t
1		$\langle x_2, open, dialogue(wi, b)\rangle$		
2		$\langle x_1, open, dialogue(ai, a \rightarrow b)\rangle$		$QS_2 = \{a, b\}$
3	$(a, 3)$	$\langle x_2, assert, \langle\{(a, 3)\}, a\rangle\rangle$		
4		$\langle x_1, assert, \langle\{(a, 3), (a \rightarrow b, 2)\}, b\rangle\rangle$	$(a, 3)$ $(a \rightarrow b, 2)$	
5		$\langle x_2, close, dialogue(ai, a \rightarrow b)\rangle$		
6		$\langle x_1, close, dialogue(ai, a \rightarrow b)\rangle$		
7		$\langle x_2, close, dialogue(wi, b)\rangle$		
8		$\langle x_1, open, dialogue(ai, c \rightarrow \neg b)\rangle$		$QS_8 = \{c, \neg b\}$
9		$\langle x_2, close, dialogue(ai, c \rightarrow \neg b)\rangle$		
10		$\langle x_1, open, dialogue(ai, d \rightarrow c)\rangle$		$QS_{10} = \{d, c\}$
11	$(d, 2)$	$\langle x_2, assert, \langle\{(d, 2)\}, d\rangle\rangle$		
12		$\langle x_1, assert, \langle\{(d, 2), (d \rightarrow c, 1)\}, c\rangle\rangle$	$(d, 2)$ $(d \rightarrow c, 1)$	
13		$\langle x_2, close, dialogue(ai, d \rightarrow c)\rangle$		
14		$\langle x_1, close, dialogue(ai, d \rightarrow c)\rangle$		
15		$\langle x_2, close, dialogue(ai, c \rightarrow \neg b)\rangle$		
16		$\langle x_1, assert, \langle\{(d, 2), (d \rightarrow c, 1),$ $(c \rightarrow b, 1)\}\rangle$	$(c \rightarrow \neg b, 1)$	
17		$\langle x_2, close, dialogue(ai, c \rightarrow \neg b)\rangle$		
18		$\langle x_1, close, dialogue(ai, c \rightarrow \neg b)\rangle$		
19		$\langle x_2, close, dialogue(wi, b)\rangle$		
20		$\langle x_1, close, dialogue(wi, b)\rangle$		

Table 6.3: Warrant inquiry dialogue example 3.

t	$CS^t_{x_1}$	m_t	$CS^t_{x_2}$	QS_t
1		$\langle x_2, open, dialogue(wi, b)\rangle$		
2		$\langle x_1, close, dialogue(wi, b)\rangle$		
3		$\langle x_2, assert, \langle\{(a, 2), (a \rightarrow b, 1)\}, b\rangle\rangle$	$(a, 2)$	
			$(a \rightarrow b, 1)$	
4		$\langle x_1, open, dialogue(ai, c \rightarrow \neg a)\rangle$		$QS_4 = \{c, \neg a\}$
5		$\langle x_2, assert, \langle\{(c, 1)\}, c\rangle\rangle$	$(c, 1)$	
6		$\langle x_1, assert, \langle\{(c, 1), (c \rightarrow \neg a, 1)\}, \neg a\rangle\rangle$	$(c \rightarrow \neg a, 1)$	
7		$\langle x_2, close, dialogue(ai, c \rightarrow \neg a)\rangle$		
8		$\langle x_1, close, dialogue(ai, c \rightarrow \neg a)\rangle$		
9	$(\neg c, 1)$	$\langle x_2, assert, \langle\{(\neg c, 1)\}, \neg c\rangle\rangle$		
10		$\langle x_1, open, dialogue(wi, b)\rangle$		
11		$\langle x_2, close, dialogue(wi, b)\rangle$		

Table 6.4: Warrant inquiry dialogue example 4.

$$\mathsf{Outcome}_{ai}(D^{6}_{2}) = \{\langle\{(a, 3), (a \rightarrow b, 2)\}, b\rangle\}$$

$$\mathsf{Outcome}_{ai}(D^{18}_{8}) = \{\langle\{(d, 2), (d \rightarrow c, 1), (c \rightarrow \neg b, 1)\}, \neg b\rangle\}$$

$$\mathsf{Outcome}_{ai}(D^{14}_{10}) = \{\langle\{(d, 2), (d \rightarrow c, 1)\}, c\rangle\}$$

Note that the dialogue tree generated here (Figure 6.6) is one node smaller than that generated in Section 4.7.11, although the outcomes of the two dialogues are the same.

6.1.4 Warrant inquiry dialogue example 4

This example uses the same situation as the one from Section 4.7.12, except the agents apply the pruned tree strategy whilst taking part in a warrant inquiry dialogue. In the following example we have an agent x_1 who wishes to enter into a dialogue with agent x_2 in order to try to find a warrant for an argument for b. We have

$$\Sigma^{x_1} = \{(a, 2), (a \rightarrow b, 1), (c, 1), (\neg c, 1)\} \qquad \Sigma^{x_2} = \{(c \rightarrow \neg a, 1)\}$$

Agent x_1 opens an warrant inquiry dialogue with b as its topic. The dialogue proceeds as in Table 6.4.

U $\langle\{(a,2),(a\rightarrow b,1)\},b\rangle$

D $\langle\{(c,1),(c\rightarrow\neg a,1)\},\neg a\rangle$

U $\langle\{(\neg c,1)\},\neg c\rangle$

Figure 6.7: The marked dialogue tree for the warrant inquiry dialogue example 4.

The outcome of the top-level warrant inquiry dialogue D_1^{11} depends on the dialogue tree $\mathsf{DialogueTree}(D_1^{11})$. The corresponding marked dialogue tree constructed at the end of the dialogue is shown in Figure 6.7.

As the root argument of the dialogue tree is undefeated, the outcome of the dialogue is the argument at the root of the tree.

$$\mathsf{Outcome}_{wi}(D_1^{11}) = \langle\{(a,2),(a\rightarrow b,1)\},b\rangle$$

Note that there is a nested argument inquiry dialogue that appears as a sub-dialogue of D_1^{11}:D_4^8.

$$\mathsf{Outcome}_{ai}(D_4^8) = \{\langle\{(c,1),(c\rightarrow\neg a,1)\},\neg a\rangle\}$$

Also note that the dialogue tree produced by this dialogue (Figure 6.7) is the same as that produced in the equivalent dialogue using the exhaustive strategy (Figure 4.6). In fact, both dialogues proceed in exactly the same way.

6.1.5 Warrant inquiry dialogue example 5

This example uses the same situation as the one from Section 4.7.13, except the agents apply the pruned tree strategy whilst taking part in a warrant inquiry dialogue. In the following example we have an agent x_1 who wishes to enter into a dialogue with agent x_2 in order to try to find a warrant for an argument for b. We have

$$\Sigma^{x_1} = \{(a,4),(a\rightarrow b,4),(c,3),(c\rightarrow\neg b,3),(e,2)\}$$
$$\Sigma^{x_2} = \{(d,3),(d\rightarrow\neg a,3),(\neg d,1),(e\rightarrow\neg d,2),(\neg e,1)\}$$

Agent x_1 opens an warrant inquiry dialogue with b as its topic. The dialogue proceeds as in Table 6.5.

The outcome of the top-level warrant inquiry dialogue D_1^{13} depends on the dialogue tree $\mathsf{DialogueTree}(D_1^{13})$. The corresponding marked dialogue tree constructed at the end of the dialogue is shown in Figure 6.8.

t	$CS^t_{x_1}$	m_t	$CS^t_{x_2}$	QS_t
1		$\langle x_2, open, dialogue(wi, b)\rangle$		
2		$\langle x_1, close, dialogue(wi, b)\rangle$		
3	$(a, 4)$	$\langle x_2, assert, \langle\{(a, 4), (a \rightarrow b, 4)\}, b\rangle\rangle$		
	$(a \rightarrow b, 4)$			
4		$\langle x_1, assert, \langle\{(d, 3), (d \rightarrow \neg a, 3)\}, \neg a\rangle\rangle$	$(d, 3)$	
			$(d \rightarrow \neg a, 3)$	
5		$\langle x_2, close, dialogue(wi, b)\rangle$		
6		$\langle x_1, assert, \langle\{(\neg d, 1)\}, \neg d\rangle\rangle$	$(\neg d, 1)$	
7	$(c, 3)$	$\langle x_2, assert, \langle\{(c, 3), (c \rightarrow \neg b, 3)\}, \neg a\rangle\rangle$		
	$(c \rightarrow \neg b, 3)$			
8		$\langle x_1, close, dialogue(wi, b)\rangle$		
9		$\langle x_2, open, dialogue(ai, a \rightarrow b)\rangle$		$QS_9 = \{a, b\}$
10		$\langle x_1, close, dialogue(ai, a \rightarrow b)\rangle$		
11		$\langle x_2, close, dialogue(ai, a \rightarrow b)\rangle$		
12		$\langle x_1, close, dialogue(wi, b)\rangle$		
13		$\langle x_2, close, dialogue(wi, b)\rangle$		

Table 6.5: Warrant inquiry dialogue example 5.

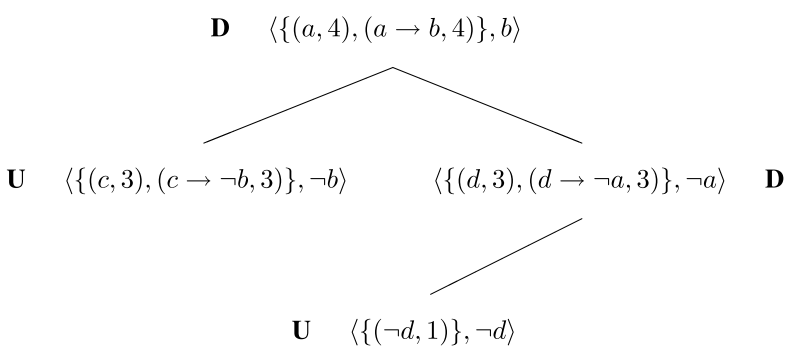


Figure 6.8: The marked dialogue tree for warrant inquiry dialogue example 5.

As the root argument of the dialogue tree is defeated, the outcome of the dialogue is the empty set.

$$\mathsf{Outcome}_{wi}(D_1^{13}) = \emptyset$$

Note that the dialogue tree produced by this dialogue (Figure 6.8) is two nodes smaller than that produced in the equivalent dialogue using the exhaustive strategy (Figure 4.7), although the outcomes of both dialogues are the same.

6.2 Summary

In this chapter I have specified a new strategy for use during warrant inquiry dialogues. I have also given several examples of dialogues generated by this strategy. These examples are easily comparable with the examples of warrant inquiry dialogues produced by the exhaustive strategy given in Section 4.7. I have observed that each dialogue example given here has the same outcome as that of the equivalent example using the exhaustive strategy (in Section 4.7). I have also observed that, in some of the examples where the pruned tree strategy is used, the dialogue tree produced has less nodes than the dialogue tree produced by the exhaustive strategy. In the next chapter I will give an analysis of the dialogues produced by the pruned tree strategy.

Chapter 7

Analysis of dialogue system with pruned tree strategy

In this chapter I give results about warrant inquiry dialogues produced by my system in which both participating agents follow the pruned tree strategy whilst participating in a warrant inquiry dialogue but still follow the exhaustive strategy whilst participating in an embedded argument inquiry dialogue (i.e. about well-formed pruned tree dialogues, Definition 6.0.2). I will go on to show that all well-formed pruned tree dialogues terminate and that they are sound and complete. I will then show that the dialogue tree produced by the pruned tree strategy is never bigger than, and sometimes smaller than, the dialogue tree produced in the same situation but by agents that are following the exhaustive strategy.

7.1 Results about commitment stores

This section gives results about the contents of commitment stores that are constructed during a well-formed pruned tree dialogue. These results are similar to those for well-formed exhaustive dialogues given in Section 5.2, but they follow from different definitions. These lemmas are particularly simple but are included as they are useful building blocks which are later reused to give more interesting results.

The first lemma states that if the pruned tree strategy selects a move that asserts an argument, then it will be possible to construct that argument from the union of the agent making the move's beliefs and the other participating agent's commitment store. This is clear from the definition of the pruned tree strategy.

Lemma 7.1.1 *Let D_r^t be a well-formed pruned tree dialogue with participants x_1 and x_2. If D_1^t is a top-dialogue of D_r^t and $\Omega_{prn}(D_1^t, \overline{P}) = \langle P, assert, \langle \Phi, \phi \rangle \rangle$, then $\langle \Phi, \phi \rangle \in \mathcal{A}(\Sigma^{\overline{P}} \cup CS_P^t)$.*

Proof: *The definition of the pruned tree strategy (Definition 6.0.1) ensures that if the assert move $\langle P, assert, \langle \Phi, \phi \rangle \rangle$ is selected then it will be the case that $\langle \Phi, \phi \rangle \in \mathcal{A}(\Sigma^{\overline{P}} \cup CS_P^t)$.* $\square$

The next lemma states that an agent's commitment store is always a subset of the union of both of the participating agents' beliefs. This is clear from the previous lemma and the fact that commitment stores are only updated when an assert move is made.

Lemma 7.1.2 *If D_r^t is a well-formed pruned tree dialogue with participants x_1 and x_2, then $CS_{x_1}^t \cup CS_{x_2}^t \subseteq \Sigma^{x_1} \cup \Sigma^{x_2}$.*

Proof: *Commitment stores are updated as follows (Definition 4.2.11).*

$$CS_P^t = \begin{cases} \emptyset & \text{iff } t = 0, \\ CS_P^{t-1} \cup \Phi & \text{iff } m_t = \langle \overline{P}, assert, \langle \Phi, \phi \rangle \rangle, \\ CS_P^{t-1} & \text{otherwise.} \end{cases}$$

Hence, the only time that a commitment store is changed is when an agent P makes the move $\langle \overline{P}, assert, \langle \Phi, \phi \rangle \rangle$. From Lemma 7.1.1, we see that for $\langle \overline{P}, assert, \langle \Phi, \phi \rangle \rangle$ to be a move made at point $t+1$ in a dialogue, the condition $\langle \Phi, \phi \rangle \in \mathcal{A}(\Sigma^P \cup CS_{\overline{P}}^t)$ must hold, hence $\Phi \subseteq \Sigma^P \cup CS_{\overline{P}}^t$ (from the definition of an argument, Definition 3.2.1). As a commitment store is empty when $t = 0$, all elements of the commitment stores must be an element of the agents' beliefs, hence $CS_{x_1}^t \cup CS_{x_2}^t \subseteq \Sigma^{x_1} \cup \Sigma^{x_2}$. □

The next lemma states that commitment stores are always finite. This is because they are subsets of the agents' beliefs, which I have assumed to be finite.

Lemma 7.1.3 *Let D_r^t be a well-formed pruned tree dialogue with participants x_1 and x_2. The sets $CS_{x_1}^t$ and $CS_{x_2}^t$ are both finite.*

Proof: *From Lemma 7.1.2 we know that $CS_{x_1}^t \cup CS_{x_2}^t \subseteq \Sigma^{x_1} \cup \Sigma^{x_2}$. As the belief bases Σ^{x_1} and Σ^{x_2} are each assumed to be finite, we know that the sets $CS_{x_1}^t$ and $CS_{x_2}^t$ are both finite.* □

In the next section I discuss moves made in a well-formed pruned tree dialogue.

7.2 Results about moves

In this section I define the sets of different types of moves made during a well-formed pruned tree dialogue I go on to show that these sets are subsets of the upper bounds that I defined in Section 5.3. Although the results given here are very similar to those given for well-formed exhaustive dialogues in Section 5.3, they follow from different definitions and lemmas.

We will see shortly that the set of possible assert moves that I defined in Chapter 5 (Definition 5.3.1) is an upper bound on the set of assert moves made during a well-formed pruned tree dialogue. I first define the set of all assert moves that are made during such a dialogue. Note that this set does not include the move made at $t = 1$. This is deliberate, as the first move in a dialogue is chosen by some higher-level planning process, assumed to be separate from this system.

Definition 7.2.1 *Let D_r^t be a well-formed pruned tree dialogue with participants x_1 and x_2. The set of* **assert moves made** *during D_r^t is denoted $\mathsf{AssertsMade}_{prn}(D_r^t)$ as follows:*

$$\begin{aligned} \mathsf{AssertsMade}_{prn}(D_r^t) = \{\langle X, assert, \langle \Phi, \phi \rangle \rangle \mid \quad & X \in \{x_1, x_2\} \text{ and either} \\ & \text{if } r \neq 1, \text{ then } \langle X, assert, \langle \Phi, \phi \rangle \rangle \text{ appears in the sequence } D_r^t \\ & \text{else, if } r = 1, \text{ then } \langle X, assert, \langle \Phi, \phi \rangle \rangle \text{ appears in the sequence } D_2^t \} \end{aligned}$$

I now show that the set of possible assert moves (Definition 5.3.1) is an upper bound on the set of assert moves that are made during a well-formed pruned tree dialogue. That is to say, if an assert move appears in such a dialogue, then it is part of the set of possible assert moves for that dialogue. This is clear from the definitions of the exhaustive and the pruned tree strategies (Definitions 4.5.4 and 6.0.1).

Lemma 7.2.1 *If D_r^t is a well-formed pruned tree dialogue with participants x_1 and x_2, then* $\mathsf{AssertsMade}_{prn}(D_r^t) \subseteq \mathsf{PossAsserts}(D_r^t)$.

Proof: *Let us assume that $\langle \overline{P}, assert, \langle \Phi, \phi \rangle \rangle \in \mathsf{AssertsMade}_{prn}(D_r^t)$ and D_1^t is a top-dialogue of D_r^t. From the definition of assert moves made (Definition 7.2.1), we see that either (1) $\langle \overline{P}, assert, \langle \Phi, \phi \rangle \rangle = \Omega_{exh}(D_1^s, P)$, for some s, where $r - 1 \leq s < t$ and D_1^t extends D_1^s, or (2) $\langle \overline{P}, assert, \langle \Phi, \phi \rangle \rangle = \Omega_{prn}(D_1^s, P)$, for some s, where $r - 1 \leq s < t$ and D_1^t extends D_1^s.*

(Case 1) From the definition of the exhaustive strategy (Definition 4.5.4), we get that $\langle \Phi, \phi \rangle \in \mathcal{A}(\Sigma^P \cup CS^s_{\overline{P}})$. From Lemma 7.1.2 and Lemma 5.1.1, we get that $\langle \Phi, \phi \rangle \in \mathcal{A}(\Sigma^{\overline{P}} \cup \Sigma^P)$. Hence (from the definition of the set of possible asserts, Definition 5.3.1), $\langle \overline{P}, assert, \langle \Phi, \phi \rangle \rangle \in \mathsf{PossAsserts}(D_r^t)$.

(Case 2) From the definition of the pruned tree strategy (Definition 6.0.1), we get that $\langle \Phi, \phi \rangle \in \mathcal{A}(\Sigma^P \cup CS^s_{\overline{P}})$. From Lemma 7.1.2 and Lemma 5.1.1, we get that $\langle \Phi, \phi \rangle \in \mathcal{A}(\Sigma^{\overline{P}} \cup \Sigma^P)$. Hence (from the definition of the set of possible asserts, Definition 5.3.1), $\langle \overline{P}, assert, \langle \Phi, \phi \rangle \rangle \in \mathsf{PossAsserts}(D_r^t)$.

Hence, from case 1 and case 2, $\mathsf{AssertsMade}_{prn}(D_r^t) \subseteq \mathsf{PossAsserts}(D_r^t)$. $\square$

I now show that the set of assert moves that are made during a dialogue is finite, as I have shown that it is a subset of the set of possible assert moves (Lemma 7.2.1), which I have already shown to be finite (Lemma 5.3.2).

Lemma 7.2.2 *If D_r^t is a well-formed pruned tree dialogue, then the set* $\mathsf{AssertsMade}_{prn}(D_r^t)$ *is finite.*

Proof: *This follows from Lemma 5.3.2 and Lemma 7.2.1.* $\square$

I now consider open moves. I will show shortly that the set of possible open moves (Definition 5.3.3) is an upper bound on the set of open moves made during a well-formed pruned tree dialogue. That is to say, if an open move is made during such a dialogue, then it must be part of the set of possible open moves. I now define the set of all open moves made during such a dialogue, which does not include the open move made at $t = 1$, as this move is assumed to be selected by some higher-level planning process external to this dialogue system.

Definition 7.2.2 *Let D_r^t be a well-formed pruned tree dialogue with participants x_1 and x_2. The set of* **open moves made** *during D_r^t is denoted* $\mathsf{OpensMade}_{prn}(D_r^t)$ *such that*

$$\begin{aligned}\mathsf{OpensMade}_{prn}(D_r^t) = \{\langle X, open, dialogue(ai, \gamma)\rangle \mid \quad & X \in \{x_1, x_2\} \text{ and either} \\ & \text{if } r \neq 1, \text{ then } \langle X, open, dialogue(ai, \gamma)\rangle \text{ appears in the sequence } D_r^t \\ & \text{else, if } r = 1, \text{ then } \langle X, open, dialogue(ai, \gamma)\rangle \text{ appears in the sequence } D_2^t\}\end{aligned}$$

I now show that the set of possible open moves is an upper bound on the set of open moves that are made during a well-formed pruned tree dialogue. This is clear from the definitions of the exhaustive and the pruned tree strategies (Definitions 4.5.4 and 6.0.1).

Lemma 7.2.3 *If D_r^t is a well-formed pruned tree dialogue with participants x_1 and x_2, then* $\mathsf{OpensMade}_{prn}(D_r^t) \subseteq \mathsf{PossOpens}(D_r^t)$.

Proof: *Let us assume that* $\langle P, open, dialogue(ai,\gamma)\rangle \in \mathsf{OpensMade}_{prn}(D_r^t)$ *and* D_1^t *is a top-dialogue of* D_r^t. *From the definition of open moves made (Definition 7.2.2), we see that either (1)* $\langle P, open, dialogue(ai,\gamma)\rangle = \Omega_{exh}(D_1^s, \overline{P})$*, for some* s, $r-1 \leq s < t$*, where* D_1^t *extends* D_1^s*, or (2)* $\langle P, open, dialogue(ai,\gamma)\rangle = \Omega_{prn}(D_1^s, \overline{P})$*, for some* s, $r-1 \leq s < t$*, where* D_1^t *extends* D_1^s.

(Case 1) From the definition of the exhaustive strategy (Definition 4.5.4), we get that there exists $L \in \mathbb{N}$ *such that* $(\gamma, L) \in \Sigma^{\overline{P}}$*, hence,* $(\gamma, L) \in \Sigma^P \cup \Sigma^{\overline{P}}$*. Hence, from the definition of the set of possible opens (Definition 5.3.3),* $\langle P, open, dialogue(ai,\gamma)\rangle \in \mathsf{PossOpens}(D_r^t)$.

Case 2: From the definition of the pruned tree strategy (Definition 6.0.1), we get that there exists $L \in \mathbb{N}$ *such that* $(\gamma, L) \in \Sigma^{\overline{P}}$*, hence,* $(\gamma, L) \in \Sigma^P \cup \Sigma^{\overline{P}}$*. Hence, from the definition of the set of possible opens (Definition 5.3.3),* $\langle P, open, dialogue(ai,\gamma)\rangle \in \mathsf{PossOpens}(D_r^t)$.

Hence, from case 1 and case 2, $\mathsf{OpensMade}_{prn}(D_r^t) \subseteq \mathsf{PossOpens}(D_r^t)$.□

I now show that the set of open moves that are made during a well-formed pruned tree dialogue is finite. This is clear as I have shown that the set of possible moves is an upper bound on the set of open moves made (Lemma 7.2.3), and I have shown that this set is finite (Lemma 5.3.6).

Lemma 7.2.4 *If* D_r^t *is a well-formed pruned tree dialogue, then the set* $\mathsf{OpensMade}_{prn}(D_r^t)$ *is finite.*
Proof: *This follows from Lemma 5.3.6 and Lemma 7.2.3.* □

I now define the set of moves used in a well-formed pruned tree dialogue. Note again that I am not considering the move made at $t = 1$, as this is selected by some higher-level process that is beyond the scope of this work.

Definition 7.2.3 *Let* D_r^t *be a well-formed pruned tree dialogue with participants* x_1 *and* x_2*. The set of* **moves made** *during* D_r^t *is denoted* $\mathsf{MovesMade}_{prn}(D_r^t)$ *such that*

$$\mathsf{MovesMade}_{prn}(D_r^t) = \{m \mid \text{ if } r \neq 1, \text{ then } m \text{ appears in the sequence } D_r^t, \text{ else, if } r = 1, \text{ then } m \text{ appears in the sequence } D_2^t\}$$

I will now show that the set of possible moves (Definition 5.3.5) is an upper bound on the set of moves made during a dialogue. Again, this is clear from the definitions of the exhaustive and the pruned tree strategies (Definitions 4.5.4 and 6.0.1).

Lemma 7.2.5 *If* D_r^t *is a well-formed pruned tree dialogue with participants* x_1 *and* x_2*, then* $\mathsf{MovesMade}_{prn}(D_r^t) \subseteq \mathsf{PossMoves}(D_r^t)$.

Proof: *Let us assume that* $m_s \in \mathsf{MovesMade}_{prn}(D_r^t)$, $r \leq s \leq t$*. From the definition of moves made (Definition 7.2.3), we see that either* $\mathsf{cType}(D_1^{s-1}) = ai$ *and* $\Omega_{exh}(D_1^s, \overline{P}) = m_s$*, or* $\mathsf{cType}(D_1^{s-1}) = wi$ *and* $\Omega_{prn}(D_1^s, \overline{P}) = m_s$*. Let us first assume* $\mathsf{cType}(D_1^{s-1}) = ai$*. If this is the case, then it follows from Lemma 5.3.10 that* $m_s \in \mathsf{PossMoves}(D_r^t)$.

Let us now assume $\mathsf{cType}(D_1^{s-1}) = wi$*. From the definition of the pruned tree strategy (Definition 6.0.1), we see there are three cases to consider.*

(Case 1) $m_s = \langle P, assert, \langle \Phi, \phi\rangle\rangle$ *such that* $\langle \Phi, \phi\rangle \in \mathcal{A}(\Sigma^P \cup CS_{\overline{P}}^{s-1})$*. From Lemma 5.1.1 and*

Lemma 7.1.2, we get that $\langle \Phi, \phi \rangle \in \mathcal{A}(\Sigma^P \cup \Sigma^{\overline{P}})$. *Hence (from the definition of possible asserts, Definition 5.3.1),* $m_s \in \mathsf{PossAsserts}(D_r^t)$. *Hence* $m_s \in \mathsf{PossMoves}(D_r^t)$ *(from the definition of possible moves, Definition 5.3.5).*
(Case 2) $m_s = \langle P, open, dialogue(ai, \gamma) \rangle$ *such that there exists* $L \in \mathbb{N}$ *such that* $(\gamma, L) \in \Sigma^P$, *hence* $m_s \in \mathsf{PossOpens}(D_r^t)$ *(from the definition of possible opens, Definition 5.3.3), hence* $m_s \in \mathsf{PossMoves}(D_r^t)$ *(from the definition of possible moves, Definition 5.3.5).*
(Case 3) $m_s = \langle P, close, dialogue(\theta, \gamma) \rangle$ *where* $\theta = \mathsf{cType}(D_1^{s-1})$ *and* $\gamma = \mathsf{cTopic}(D_1^{s-1})$, *hence (from the definition of the current dialogue, Definition 4.2.9), there must exist* s' *such that* $r \leq s' < s$ *and* $m_{s'} = \langle P', open, dialogue(\theta, \gamma) \rangle$ *(where* $P' \in \{x_1, x_2\}$*), hence (from the definition of possible moves, Definition 5.3.5),* $m_s \in \mathsf{PossMoves}(D_r^t)$.
Hence, from case 1, case 2 and case 3, $\mathsf{MovesMade}_{prn}(D_r^t) \subseteq \mathsf{PossMoves}(D_r^t)$. □

The following lemma shows that the set of moves made is finite. This is because this set is bounded above by the set of possible moves (Lemma 7.2.5), which I have already shown to be finite (Lemma 5.3.9).

Lemma 7.2.6 *If* D_r^t *is a well-formed pruned tree dialogue, then the set* $\mathsf{MovesMade}_{prn}(D_r^t)$ *is finite.*
Proof: *This follows from Lemma 5.3.9 and Lemma 7.2.5.* □

Although I have shown that the set of moves made during a well-formed pruned tree dialogue is finite, this does not necessarily mean that a dialogue terminates. It may be the case that moves get repeated or that a dialogue does not end with a matched-close. In the next section, I will show that all well-formed pruned tree dialogues do indeed terminate and that it is not the case that moves get repeated.

7.3 Results about termination of dialogues

I will shortly show that all dialogues produced by the pruned tree strategy terminate, but in order to do so I must first introduce some lemmas. The first two lemmas are concerned with the repetition of moves during a dialogue.

The following lemma shows that a maximum of one assert move with a certain content appears in any top-level well-formed pruned tree dialogue. That is to say, both moves $m_s = \langle P, assert, \langle \Phi, \phi \rangle \rangle$ and $m_{s'} = \langle P', assert, \langle \Phi, \phi \rangle \rangle$ appear in well-formed pruned tree dialogue D_1^t (where $P, P' \in \{x_1, x_2\}$), if and only if $s = s'$. This is clear from the definition of the warrant inquiry and argument inquiry protocols. The warrant inquiry protocol contains a constraint that something may only be asserted if it changes the dialogue tree, which would not occur if the argument had been previously asserted. The argument inquiry protocol contains a constraint that something may only be asserted if its support is not present in the union of the commitment stores, which it would be if the argument had previously been asserted.

Lemma 7.3.1 *Let* D_r^t *be a well-formed pruned tree dialogue with participants* x_1 *and* x_2, *such that* D_1^t *is a top-dialogue of* D_r^t *and* $m_s = \langle P, assert, \langle \Phi, \phi \rangle \rangle$ *appears in the sequence* D_1^t *for some* s, $1 < s \leq t$, *where* $P \in \{x_1, x_2\}$. *There does not exist an* s' *such that* $1 < s' \leq t$, $s \neq s'$ *and*

$m_{s'} = \langle P', assert, \langle \Phi, \phi \rangle \rangle$ appears in the sequence D_1^t where $P' \in \{x_1, x_2\}$.

Proof by contradiction: *Let us assume that $m_s = \langle P, assert, \langle \Phi, \phi \rangle \rangle$. Either $\mathsf{Current}(D_1^{s-1}) = ai$ and $\Omega_{exh}(D_1^{s-1}, \overline{P}) = m_s$, in which case the proof proceeds as in Lemma 5.4.1, or $\mathsf{Current}(D_1^{s-1}) = wi$ and $\Omega_{prn}(D_1^{s-1}, \overline{P}) = m_s$, in which case the proof proceeds as follows.*

From the definition of the pruned tree strategy (Definition 6.0.1) we see that $\langle P, assert, \langle \Phi, \phi \rangle \rangle \in \Pi_{wi}^{assert}(D_1^{s-1}, \overline{P})$. Let us assume that there does exist an s' such that $1 < s' \leq t$, $s \neq s'$ and $m_{s'} = \langle P', assert, \langle \Phi, \phi \rangle \rangle$ where $P' \in \{x_1, x_2\}$. According to the warrant inquiry protocol (Definition 4.4.3) this means that $\mathsf{DialogueTree}(D_1^{s'-1} + \langle P', assert, \langle \Phi, \phi \rangle \rangle) \neq \mathsf{DialogueTree}(D_1^{s'-1})$. Hence, it cannot be the case that $s < s'$ as if this were true then it would be true that $\Phi \subseteq CS_P^{s'-1} \cup CS_{\overline{P}}^{s'-1}$ (from the definition of commitment store update and the fact that commitment stores grow monotonically, Definition 4.2.11 and Lemma 5.2.4), which would mean that asserting $\langle \Phi, \phi \rangle$ would not have an effect on the dialogue tree and it would not be the case that $\mathsf{DialogueTree}(D_1^{s'-1} + \langle P', assert, \langle \Phi, \phi \rangle \rangle) \neq \mathsf{DialogueTree}(D_1^{s'-1})$ (as the dialogue tree depends on the root argument, which does not change, and the content of the commitment stores, Definition 4.4.2). It also cannot be the case that $s' < s$, as then it would be the case that $\Phi \subseteq CS_P^{s-1} \cup CS_{\overline{P}}^{s-1}$ (from Definition 4.2.11 and Lemma 5.2.4), and so making the move m_s would not alter the commitment stores and so would not alter the dialogue tree. Hence, it must be the case that $s = s'$, which contradicts our assumption.□

The next lemma is similar to the previous one but concerns open moves. It states that a maximum of one open move with a certain content appears in any top-level dialogue. That is to say, both moves $m_s = \langle P, open, dialogue(ai, \gamma) \rangle$ and $m_{s'} = \langle P', open, dialogue(ai, \gamma) \rangle$ appear in dialogue D_r^t (where $P, P' \in \{x_1, x_2\}$), if and only if $s = s'$. Again, this is clear from the definitions of the warrant inquiry and argument inquiry protocols. Both these protocols contain the constraint that a move opening an argument inquiry dialogue with the topic $\alpha_1 \wedge \ldots \wedge \alpha_n \rightarrow \beta$ can only be made if a question store with the content $\{\alpha_1, \ldots, \alpha_n, \beta\}$ has not previously been constructed, which it would have been if a move opening an argument inquiry dialogue with the topic $\alpha_1 \wedge \ldots \wedge \alpha_n \rightarrow \beta$ had previously been made.

Lemma 7.3.2 *Let D_r^t be a well-formed pruned tree dialogue with participants x_1 and x_2, such that D_1^t is a top-dialogue of D_r^t and $m_s = \langle P, open, dialogue(ai, \alpha_1 \wedge \ldots \wedge \alpha_n \rightarrow \beta) \rangle$ appears in the sequence D_1^t for some s, $1 < s \leq t$, where $P \in \{x_1, x_2\}$. There does not exist an s' such that $1 < s' \leq t$, $s \neq s'$ and $m_{s'} = \langle P', open, dialogue(ai, \alpha_1 \wedge \ldots \wedge \alpha_n \rightarrow \beta) \rangle$ appears in the sequence D_1^t where $P' \in \{x_1, x_2\}$.*

Proof: *Let us assume that $m_s = \langle P, open, dialogue(ai, \alpha_1 \wedge \ldots \wedge \alpha_n \rightarrow \beta) \rangle$. Either $\mathsf{Current}(D_1^{s-1}) = ai$ and $\Omega_{exh}(D_1^{s-1}, \overline{P}) = m_s$, in which case the proof proceeds as in Lemma 5.4.2, or $\mathsf{Current}(D_1^{s-1}) = wi$ and $\Omega_{prn}(D_1^{s-1}, \overline{P}) = m_s$, in which case the proof proceeds as follows.*

From the definition of the pruned tree strategy (Definition 6.0.1) we see that $\langle P, open, dialogue(ai, \alpha_1 \wedge \ldots \wedge \alpha_n \rightarrow \beta) \rangle \in \Pi_{wi}^{open}(D_1^{s-1}, \overline{P})$. According to the warrant inquiry protocol (Definition 4.4.3), this means that there does not exist t', $1 < t' \leq s-1$, such that $QS_{t'} = \{\alpha_1, \ldots, \alpha_n, \beta\}$. Hence (from the definition of a question store, Definition 4.3.1), there does not exist s', $1 < s' < s$, such that $m_{s'} = \langle P', open, dialogue(ai, \alpha_1 \wedge \ldots \wedge \alpha_n \rightarrow \beta) \rangle$ where $P' \in \{x_1, x_2\}$. It also cannot be the case *that there exists s' such that $s < s' \leq t$ and $m_{s'} = \langle P', open, dialogue(ai, \alpha_1 \wedge \ldots \wedge \alpha_n \rightarrow \beta)\rangle$ where $P' \in \{x_1, x_2\}$, as $QS_s = \{\alpha_1, \ldots, \alpha_n, \beta\}$ and $1 < s < s'$ which violates a condition of the warrant inquiry protocol (Definition 4.4.3).* □

I am now able to prove the theorem that all well-formed pruned tree dialogues terminate. This follows from the fact that there are always a finite number of assert and open moves from which the participating agents can choose, that the agents cannot repeat these moves, and that the pruned tree strategy dictates that if an agent cannot make an assert or open move then it must make a move to close the current dialogue.

Theorem 7.3.1 *If D_r^t is a well-formed pruned tree dialogue with participants x_1 and x_2 and D_1^t is a top-dialogue of D_r^t, then there exists t' such that $t \leq t'$, $D_r^{t'}$ extends D_r^t, $D_1^{t'}$ is a top-dialogue of $D_r^{t'}$, $D_1^{t'}$ extends D_1^t and $D_r^{t'}$ terminates at t'.*

Proof: *The set of assert moves made during a well-formed pruned tree dialogue, $\mathsf{AssertsMade}_{exh}(D_r^{t'})$, is finite (Lemma 7.2.2). The set of open moves made during a well-formed pruned tree dialogue, $\mathsf{OpensMade}_{exh}(D_r^{t'})$, is finite (Lemma 7.2.4). Neither assert nor open moves may be repeated in a well-formed pruned tree dialogue (Lemma 7.3.1 and Lemma 7.3.2), hence there must come a point in every dialogue where no more open or assert moves may be made. The pruned tree strategy (Definition 6.0.1) dictates that when there are no more assert or open moves that can be made, the participating agents must each make a move to close the current dialogue (called a matched close, Definition 4.2.7). When this occurs the dialogue terminates (Definition 4.2.8).* □

I now give two lemmas that relate to all well-formed warrant inquiry dialogues, regardless of what strategy is being followed. The next lemma states that if a well-formed warrant inquiry dialogue terminates at t, then the commitment stores of the participating agents at t are the same as the commitment stores at $t-1$ and at $t-2$. This is because both agents must make a close move to terminate the dialogue and close moves have no effect on commitment stores.

Lemma 7.3.3 *If D_r^t is a well-formed warrant inquiry dialogue that terminates at t with participants x_1 and x_2, then $CS_{x_1}^t = CS_{x_1}^{t-1}$, $CS_{x_1}^t = CS_{x_1}^{t-2}$, $CS_{x_2}^t = CS_{x_2}^{t-1}$, and $CS_{x_2}^t = CS_{x_2}^{t-2}$.*

Proof: *From the definition of terminates (Definition 4.2.8), we see that m_t is a close move and m_{t-1} is a close move. From the definition of commitment store update (Definition 4.2.11), we see that it must be the case that $CS_{x_1}^t = CS_{x_1}^{t-1}$, $CS_{x_1}^t = CS_{x_1}^{t-2}$, $CS_{x_2}^t = CS_{x_2}^{t-1}$, and $CS_{x_2}^t = CS_{x_2}^{t-2}$.* □

The next lemma uses the previous result to show that if a well-formed warrant inquiry dialogue terminates at t then the dialogue tree at t is the same as the dialogue tree at $t-1$ and at $t-2$. This is clear as the dialogue tree depends on only two things. Firstly, the root argument, which does not change throughout a dialogue as it is the first argument for the topic that is asserted. Secondly, the tree depends on the commitment stores, which I have shown do not change from $t-2$ to t (Lemma 7.3.3).

Lemma 7.3.4 *If D_r^t is a well-formed warrant inquiry dialogue that terminates at t, then $\mathsf{DialogueTree}(D_r^t) = \mathsf{DialogueTree}(D_r^{t-1})$ and $\mathsf{DialogueTree}(D_r^{t-1}) = \mathsf{DialogueTree}(D_r^{t-2})$.*

Proof: *From the definition of terminates (Definition 4.2.8), we see that* m_t *is a close move and* m_{t-1} *is a close move. From the definition of a dialogue tree (Definition 4.4.2), and Lemma 7.3.3 we see that it must be the case that* $\mathsf{DialogueTree}(D_r^t) = \mathsf{DialogueTree}(D_r^{t-1})$ *and* $\mathsf{DialogueTree}(D_r^{t-1}) = \mathsf{DialogueTree}(D_r^{t-2})$. $\square$

Finally in this section, I show that if a well-formed pruned tree warrant inquiry dialogue terminates at t, then the subset of the set of legal moves from which an agent must select the move m_t does not include any open or assert moves. This is clear from the definition of the pruned tree strategy (Definition 6.0.1), which states that a close move may only be made if there are no assert or open moves to choose from.

Lemma 7.3.5 *If* D_r^t *is a well-formed pruned tree dialogue that terminates at* t *with participants* x_1 *and* x_2*, such that* $\mathsf{Receiver}(m_{t-1}) = P$, D_r^t *extends* D_r^{t-1} *and* D_1^{t-1} *is a top-dialogue of* D_r^{t-1}*, then the set* $\mathsf{Asserts}_{prn}(D_1^{t-1}, P) = \emptyset$ *and the set* $\mathsf{Opens}_{prn}(D_1^{t-1}, P) = \emptyset$.

Proof: *A dialogue is terminated with a matched-close (Definition 4.2.8). The pruned tree strategy (Definition 6.0.1) states that a close move will only be made by* P *at timepoint* t *if the sets* $\mathsf{Asserts}_{prn}(D_1^{t-1}, P)$ *and* $\mathsf{Opens}_{prn}(D_1^{t-1}, P)$ *are empty. Hence,* $\mathsf{Asserts}_{prn}(D_1^{t-1}, P) = \emptyset$ *and* $\mathsf{Opens}_{prn}(D_1^{t-1}, P) = \emptyset$. $\square$

In this section, I have shown that all well-formed pruned tree warrant inquiry dialogues terminate. In the next section I discuss some results about the root argument of such a dialogue.

7.4 Results about root arguments

In this section I give two results relating to the root argument of a well-formed pruned tree warrant inquiry dialogue. I show that if there exists an argument for the topic of such a dialogue that can be constructed from the union of the agents' beliefs, then there will exist a root argument for the dialogue when it terminates. I then combine this with the result that all such dialogues terminate.

The first lemma shows that if there is some argument for the topic of a well-formed pruned tree dialogue that can be constructed from the union of the agents' beliefs, then, if the dialogue terminates at t, the root argument of the dialogue at t will not be $null$. This is clear from the definition of the pruned tree strategy.

Lemma 7.4.1 *If* D_r^t *is a well-formed pruned tree dialogue that terminates at* t *with participants* x_1 *and* x_2 *such that* $\mathsf{Topic}(D_r^t) = \phi$ *and there exists* $\langle \Phi, \phi \rangle \in \mathcal{A}(\Sigma^{x_1} \cup \Sigma^{x_2})$*, then* $\mathsf{RootArg}(D_r^t) \neq null$.

Proof: *There are two cases to consider.*

(Case 1) $\langle \Phi, \phi \rangle \in \mathcal{A}(\Sigma^P)$ *for some* $P \in \{x_1, x_2\}$*. We see from the definition of the warrant inquiry protocol (Definition 4.4.3) and the definition of the pruned tree strategy (Definition 6.0.1) that if* $\overline{P}$ *is the agent who opens the dialogue, then agent* P *will make the move* $m_2 = \langle \overline{P}, assert, \langle \Phi, \phi \rangle \rangle$.

Else, if P *is the agent who opens the dialogue, then agent* P *will make the move* $m_3 = \langle \overline{P}, assert, \langle \Phi, \phi \rangle \rangle$ *unless doing so does not change the dialogue tree, which would mean that there was already a root argument.*

(Case 2) $\langle \Phi, \phi \rangle \notin \mathcal{A}(\Sigma^P)$ *for some* $P \in \{x_1, x_2\}$*. Hence, it must be the case that there exists* $(\alpha_1 \wedge \ldots \wedge \alpha_n \rightarrow \beta, L) \in \Sigma^P$*, for some* $P \in \{x_1, x_2\}$ *and some* $L \in \mathbb{N}$*. We see from the definition of the warrant inquiry protocol (Definition 4.4.3) and the definition of the pruned tree strategy (Definition 6.0.1) that if* $\overline{P}$ *is the agent who opens the dialogue, then agent* P *will make the move* $m_2 = \langle \overline{P}, open, dialogue(ai, \alpha_1 \wedge \ldots \wedge \alpha_n \rightarrow \beta) \rangle$*. As argument inquiry dialogues are complete (Theorem 5.5.3), it must be the case that* $\langle \Phi, \phi \rangle \in \mathcal{A}(CS^t_{x_2} \cup CS^t_{x_1})$*, hence there must be some root argument.*
Else, if P *is the agent who opens the dialogue, then agent* P *will make the move* $m_3 = \langle \overline{P}, open, dialogue(ai, \alpha_1 \wedge \ldots \wedge \alpha_n \rightarrow \beta) \rangle$ *if* $\mathsf{RootArg}(D^2_r) = null$*, from the definition of the pruned tree strategy (Definition 6.0.1). As argument inquiry dialogues are complete (Theorem 5.5.3), it must be the case that* $\langle \Phi, \phi \rangle \in \mathcal{A}(CS^t_{x_2} \cup CS^t_{x_1})$*, hence there must be some root argument.* □

The next lemma states that for all well-formed pruned tree dialogues, if there is some argument for the topic of a well-formed pruned tree dialogue that can be constructed from the union of the agents' beliefs, then the root argument of the dialogue at t will not be $null$. This is clear from the previous result and the theorem that all well-formed pruned tree dialogues in which the participants are following the pruned tree strategy terminate.

Lemma 7.4.2 *If* D^t_r *is a well-formed pruned tree dialogue with participants* x_1 *and* x_2 *such that* $\mathsf{Topic}(D^t_r) = \phi$ *and there exists* $\langle \Phi, \phi \rangle \in \mathcal{A}(\Sigma^{x_1} \cup \Sigma^{x_2})$*, then there exists* t' *such that* $D^{t'}_r$ *extends* D^t_r *and* $\mathsf{RootArg}(D^t_r) \neq null$*.*
Proof: *This follows from Theorem 7.3.1 and Lemma 7.4.1.*□

In the next section I give results about the dialogue trees constructed during well-formed pruned tree dialogues.

7.5 Results about dialogue trees

In this section I give some results concerning the dialogue tree constructed during a well-formed pruned tree dialogue. The following lemma states that if we have a well-formed pruned tree dialogue D^t_r that terminates at t, whose root argument is $\langle \Phi, \phi \rangle$, and there is a path from the root node $[\langle \Phi, \phi \rangle, \langle \Phi_1, \phi_1 \rangle, \ldots, \langle \Phi_n, \phi_n \rangle]$ that appears in $\mathsf{DialogueTree}(D^t_r)$, then the same path appears in the dialectical tree that is constructed from the union of the two participating agents' beliefs and has $\langle \Phi, \phi \rangle$ at its root. This is due to the relationship between the commitment stores and the agents' beliefs. An argument that can be constructed from the union of the agents' commitment stores can also be constructed from the union of their beliefs (Lemmas 5.1.1 and 7.1.2). As the dialogue tree is constructed from the commitment stores, it follows that a dialectical tree with the same argument at its root that is constructed from the agents' beliefs will include any paths from the root that appear in the dialogue tree.

Lemma 7.5.1 *Let* D^t_r *be a well-formed pruned tree dialogue that terminates at* t *with participants* x_1 *and* x_2 *such that* D^t_1 *is a top-dialogue of* D^t_r *and* $\mathsf{RootArg}(D^t_r) = \langle \Phi, \phi \rangle$*. If there exists a path* $[\langle \Phi, \phi \rangle, \langle \Phi_1, \phi_1 \rangle, \ldots, \langle \Phi_n, \phi_n \rangle]$ *in* $\mathsf{DialogueTree}(D^t_r)$*, then there exists a path*

$[\langle\Phi,\phi\rangle,\langle\Phi_1,\phi_1\rangle,\ldots,\langle\Phi_n,\phi_n\rangle]$ in the dialectical tree T^{Δ}_{A} where $A=\langle\Phi,\phi\rangle$ and $\Delta=\Sigma^{x_1}\cup\Sigma^{x_2}$.

Proof: *Let us assume that the path $[\langle\Phi,\phi\rangle,\langle\Phi_1,\phi_1\rangle,\ldots,\langle\Phi_n,\phi_n\rangle]$ appears in $\mathsf{DialogueTree}(D^t_r)$. From the definition of a dialogue tree (Definition 4.4.2), we see that this means that the path $[\langle\Phi,\phi\rangle,\langle\Phi_1,\phi_1\rangle,\ldots,\langle\Phi_n,\phi_n\rangle]$ appears in the dialectical tree $\mathsf{T}^{\Delta'}_{A}$ where $\Delta'=CS^t_{x_1}\cup CS^t_{x_2}$. Hence, for $1\le i\le n$, $\langle\Phi_i,\phi_i\rangle\in\mathcal{A}(CS^t_{x_1}\cup CS^t_{x_2})$, hence $\langle\Phi_i,\phi_i\rangle\in\mathcal{A}(\Sigma^{x_1}\cup\Sigma^{x_2})$ (from Lemma 7.1.2 and Lemma 5.1.1). As $[\langle\Phi,\phi\rangle,\langle\Phi_1,\phi_1\rangle,\ldots,\langle\Phi_n,\phi_n\rangle]$ is an acceptable argumentation line in $\mathsf{DialogueTree}(D^t_r)$, it must also be an acceptable argumentation line from the root node of $\mathsf{T}^{\Delta'}_{A}$ (from the definition of an acceptable argumentation line, Definition 3.5.5). Hence (from the definition of a dialectical tree, Definition 3.6.1), if there exists a path $[\langle\Phi,\phi\rangle,\langle\Phi_1,\phi_1\rangle,\ldots,\langle\Phi_n,\phi_n\rangle]$ in $\mathsf{DialogueTree}(D^t_r)$, then there exists a path $[\langle\Phi,\phi\rangle,\langle\Phi_1,\phi_1\rangle,\ldots,\langle\Phi_n,\phi_n\rangle]$ in T^{Δ}_{A}.* □

The previous result tells us that the dialogue tree constructed during a well-formed pruned tree dialogue in which the agents are following the pruned tree strategy, is a sub-tree of the dialectical tree constructed from the union of the two agents' beliefs that has the same argument at the root. I will shortly show that if a well-formed pruned tree dialogue terminates at t, then there are no arguments that can be constructed from the union of the agents' beliefs and which, if asserted, would change the status of a node in the dialogue tree. In order to show this, I show that if if there are any such arguments that could be asserted, then the sets of legal assert and open moves from which an agent selects the most preferred move to make cannot both be empty, in which case a close move would not be made and the dialogue would not terminate.

The next lemma states that if an agent participating in a well-formed pruned tree dialogue can construct an argument that, if asserted, would alter the status of a node in the dialogue tree, then the set of assert moves from which it may select the most preferred move is not empty. This is clear from the definition of the pruned tree strategy.

Lemma 7.5.2 *Let D^t_r be a well-formed pruned tree dialogue with participants x_1 and x_2 such that $\mathsf{RootArg}(D^t_r)\neq null$ and D^t_1 is a top-dialogue of D^t_r. If there exists $\langle\Phi,\phi\rangle\in\mathcal{A}(\Sigma^P\cup CS^t_{\overline{P}})$ such that there exists $N\in\mathsf{Nodes}(\mathsf{DialogueTree}(D^t_r))$ and $\mathsf{Status}(N,\mathsf{DialogueTree}(D^t_r))=\mathsf{U}$ and $\mathsf{Status}(N,\mathsf{DialogueTree}(D^t_r+\langle P,assert,\langle\Phi,\phi\rangle\rangle))=\mathsf{D}$, then $\mathsf{Asserts}_{prn}(D^t_1,P)\neq\emptyset$.*

Proof: *The pruned tree strategy (Definition 6.0.1), states that, as $\mathsf{RootArg}(D^t_r)\neq null$, the set $\mathsf{Asserts}_{prn}(D^t_1,P)$ consists of the moves that assert arguments that can be constructed by P and that when asserted change the status of a node in the dialogue tree from U to D.*

(This appears in Definition 6.0.1 as the moves that assert $\langle\Phi,\phi\rangle$ such that $\langle\Phi,\phi\rangle\in\mathcal{A}(\Sigma^P\cup CS^t_{\overline{P}})$ and there exists $N\in\mathsf{Nodes}(\mathsf{DialogueTree}(D^t_r))$ such that $\mathsf{Status}(N,\mathsf{DialogueTree}(D^t_r))=\mathsf{U}$ and $\mathsf{Status}(N,\mathsf{DialogueTree}(D^t_r+\langle\overline{P},assert,\langle\Phi,\phi\rangle\rangle))=\mathsf{D}$.)

As an extra constraint, the moves in $\mathsf{Asserts}_{prn}(D^t_1,P)$ must also change the status of a node at least as far away from the root of the dialogue tree than any other nodes whose status would be changed by some other legal assert move whose content can be constructed by P.

(This appears in Definition 6.0.1 as for all $\langle\overline{P},assert,\langle\Phi',\phi'\rangle\rangle\in\Pi^{assert}_{wi}(D^t_1,P)$ such that $\langle\Phi',\phi'\rangle\in\mathcal{A}(\Sigma^P\cup CS^t_{\overline{P}})$, if $N'\in\mathsf{Nodes}(\mathsf{DialogueTree}(D^t_r))$ such that $\mathsf{Status}(N',\mathsf{DialogueTree}(D^t_r))=\mathsf{U}$ and

$\mathsf{Status}(N', \mathsf{DialogueTree}(D_r^t + \langle \overline{P}, assert, \langle \Phi', \phi' \rangle \rangle)) = \mathsf{D}$, *then* $\mathsf{Level}(N) \geq \mathsf{Level}(N')$*)*
Hence, if there exists $\langle \Phi, \phi \rangle \in \mathcal{A}(\Sigma^P \cup CS^t_{\overline{P}})$ *such that there exists* $N \in \mathsf{Nodes}(\mathsf{DialogueTree}(D_r^t))$ *and* $\mathsf{Status}(N, \mathsf{DialogueTree}(D_r^t)) = \mathsf{U}$ *and* $\mathsf{Status}(N, \mathsf{DialogueTree}(D_r^t + \langle P, assert, \langle \Phi, \phi \rangle \rangle)) = \mathsf{D}$, *then* $\mathsf{Asserts}_{prn}(D_1^t, P)$ *must contain at least one assert move.* □

The next lemma shows that if an agent has a belief in a defeasible rule whose consequent is β and there is a node in the dialogue tree whose status is U and which is labelled with an argument $\langle \Phi, \phi \rangle$, such that one can defeasibly derive $\neg\beta$ from Φ, then the set of open moves from which it may select the most preferred open move is not equal to the empty set. This is clear from the definition of the pruned tree strategy.

Lemma 7.5.3 *Let* D_r^t *be a well-formed pruned tree dialogue with participants* x_1 *and* x_2 *such that* $\mathsf{RootArg}(D_r^t) \neq null$ *and* D_1^t *is a top-dialogue of* D_r^t*. If there exists* $(\alpha_1 \wedge \ldots \wedge \alpha_n \rightarrow \beta, L) \in \Sigma^P$ *such that there exists* $\langle \Phi, \phi \rangle \in \mathcal{A}(CS_P^{t-1} \cup CS_{\overline{P}}^{t-1})$ *such that* $\Phi \mid\!\sim \neg\beta$ *and there exists* $N \in \mathsf{Nodes}(\mathsf{DialogueTree}(D_r^{t-1}))$ *such that* $\mathsf{Label}(N) = \langle \Phi, \phi \rangle$ *and* $\mathsf{Status}(N, \mathsf{DialogueTree}(D_r^{t-1})) = \mathsf{U}$, *then* $\mathsf{Opens}_{prn}(D_1^t, P) \neq \emptyset$.
Proof: *The pruned tree strategy (Definition 6.0.1) states that, as* $\mathsf{RootArg}(D_r^t) \neq null$*, the set* $\mathsf{Opens}_{prn}(D_1^t, P)$ *consists of the moves that open an argument inquiry dialogue with a rule from* P*'s beliefs* $\alpha_1 \wedge \ldots \wedge \alpha_n \rightarrow \beta$ *as its topic such that there is a node in the dialogue tree* N *whose status is* U *and that is labelled with an argument, from the support of which* $\neg\beta$ *can be defeasibly derived (and so an argument for* β *would conflict with it).*
(This appears in Definition 6.0.1 as there exists $L \in \mathbb{N}$ *such that* $(\alpha_1 \wedge \ldots \wedge \alpha_n \rightarrow \beta, L) \in \Sigma^P$ *and there exists* $\langle \Phi, \phi \rangle \in \mathcal{A}(CS_P^t \cup CS_{\overline{P}}^t)$ *such that* $\Phi \mid\!\sim \neg\beta$ *and there exists* $N \in \mathsf{Nodes}(\mathsf{DialogueTree}(D_r^t))$ *such that* $\mathsf{Label}(N) = \langle \Phi, \phi \rangle$ *and* $\mathsf{Status}(N, \mathsf{DialogueTree}(D_r^t)) = \mathsf{U}$.*)*
As an extra constraint, for all other legal open moves that open an argument inquiry dialogue with a rule from P*'s beliefs* $\alpha'_1 \wedge \ldots \wedge \alpha'_n \rightarrow \beta'$ *as its topic such that there is a node in the dialogue tree* N *whose status is* U *and that is labelled with an argument, from the support of which* $\neg\beta'$ *can be defeasibly derived, the distance from the root to* N *must be at least as far as the distance from the root to* N'*.*
(This appears in Definition 6.0.1 as for all $\langle \overline{P}, assert, \langle \Phi', \phi' \rangle \rangle \in \Pi_{wi}^{assert}(D_1^t, P)$ *such that* $\langle \Phi', \phi' \rangle \in \mathcal{A}(\Sigma^P \cup CS^t_{\overline{P}})$ *if* $N' \in \mathsf{Nodes}(\mathsf{DialogueTree}(D_r^t))$ *such that* $\mathsf{Status}(N', \mathsf{DialogueTree}(D_r^t)) = \mathsf{U}$ *and* $\mathsf{Status}(N', \mathsf{DialogueTree}(D_r^t + \langle \overline{P}, assert, \langle \Phi', \phi' \rangle \rangle)) = \mathsf{D}$, *then* $\mathsf{Level}(N) \geq \mathsf{Level}(N')$.*)*
Hence, if there exists $\langle \Phi, \phi \rangle \in \mathcal{A}(CS_P^{t-1} \cup CS_{\overline{P}}^{t-1})$ *such that* $\Phi \mid\!\sim \neg\beta$ *and there exists* $N \in \mathsf{Nodes}(\mathsf{DialogueTree}(D_r^{t-1}))$ *such that* $\mathsf{Label}(N) = \langle \Phi, \phi \rangle$ *and* $\mathsf{Status}(N, \mathsf{DialogueTree}(D_r^{t-1})) = \mathsf{U}$, *then* $\mathsf{Opens}_{prn}(D_1^t, P)$ *must contain at least one open move.* □

The next lemma states that if a well-formed pruned tree dialogue terminates at t, then neither agent can construct an argument which, if asserted, would change the status of a node in the dialogue tree. This is because if an agent could construct such an argument, then the set of assert moves available to it would not be empty and so it would not make a close move and the dialogue would not terminate at t.

Lemma 7.5.4 *Let* D_r^t *be a well-formed pruned tree dialogue that terminates at* t *with participants* x_1 *and* x_2 *such that* D_1^t *is a top-dialogue of* D_r^t *and* $\mathsf{Receiver}(m_t) = x_2$.

Part 1: *There does not exist* $\langle\Phi,\phi\rangle \in \mathcal{A}(\Sigma^{x_1} \cup CS^t_{x_2})$ *such that there exists* $N \in \mathsf{Nodes}(\mathsf{DialogueTree}(D^t_r))$ *and* $\mathsf{Status}(N, \mathsf{DialogueTree}(D^t_r)) = \mathsf{U}$ *and* $\mathsf{Status}(N, \mathsf{DialogueTree}(D^t_r + \langle x_2, assert, \langle\Phi,\phi\rangle\rangle)) = \mathsf{D}$.

Part 2: *There does not exist* $\langle\Phi,\phi\rangle \in \mathcal{A}(\Sigma^{x_2} \cup CS^t_{x_1})$ *such that there exists* $N \in \mathsf{Nodes}(\mathsf{DialogueTree}(D^t_r))$ *and* $\mathsf{Status}(N, \mathsf{DialogueTree}(D^t_r)) = \mathsf{U}$ *and* $\mathsf{Status}(N, \mathsf{DialogueTree}(D^t_r + \langle x_1, assert, \langle\Phi,\phi\rangle\rangle)) = \mathsf{D}$.

Proof by contradiction:

(Part 1) As D^t_r *terminates at* t *and* $\mathsf{Receiver}(m_t) = x_2$*, it must be the case that agent* x_1 *made the close move* m_t *(from the definition of terminates, Definition 4.2.8), hence* $\mathsf{Asserts}_{prn}(D^{t-1}_1, x_1) = \emptyset$ *(from Lemma 7.3.5).*

Let us assume that there does exist $\langle\Phi,\phi\rangle \in \mathcal{A}(\Sigma^{x_1} \cup CS^t_{x_2})$ *such that there exists* $N \in \mathsf{Nodes}(\mathsf{DialogueTree}(D^t_r))$ *and* $\mathsf{Status}(N, \mathsf{DialogueTree}(D^t_r)) = \mathsf{U}$ *and* $\mathsf{Status}(N, \mathsf{DialogueTree}(D^t_r + \langle x_1, assert, \langle\Phi,\phi\rangle\rangle)) = \mathsf{D}$*. From Lemmas 7.3.3 and 7.3.4, there exists* $\langle\Phi,\phi\rangle \in \mathcal{A}(\Sigma^{x_1} \cup CS^{t-1}_{x_2})$ *such that there exists* $N \in \mathsf{Nodes}(\mathsf{DialogueTree}(D^{t-1}_r))$ *and* $\mathsf{Status}(N, \mathsf{DialogueTree}(D^{t-1}_r)) = \mathsf{U}$ *and* $\mathsf{Status}(N, \mathsf{DialogueTree}(D^{t-1}_r + \langle x_2, assert, \langle\Phi,\phi\rangle\rangle)) = \mathsf{D}$*. From Lemma 7.5.2, we see that this means that there must be at least one element in the set* $\mathsf{Asserts}_{prn}(D^{t-1}_1, x_1)$*, contradicting the assumption.*

(Part 2) As D^t_r *terminates at* t *and* $\mathsf{Receiver}(m_t) = x_2$*, it must be the case that agent* x_2 *made the close move* m_{t-1} *(from the definition of terminates, Definition 4.2.8), hence* $\mathsf{Asserts}_{prn}(D^{t-2}_1, x_2) = \emptyset$ *(from Lemma 7.3.5).*

Let us assume that there does exist $\langle\Phi,\phi\rangle \in \mathcal{A}(\Sigma^{x_2} \cup CS^t_{x_1})$ *such that there exists* $N \in \mathsf{Nodes}(\mathsf{DialogueTree}(D^t_r))$ *and* $\mathsf{Status}(N, \mathsf{DialogueTree}(D^t_r)) = \mathsf{U}$ *and* $\mathsf{Status}(N, \mathsf{DialogueTree}(D^t_r + \langle x_1, assert, \langle\Phi,\phi\rangle\rangle)) = \mathsf{D}$*. From Lemmas 7.3.3 and 7.3.4, there exists* $\langle\Phi,\phi\rangle \in \mathcal{A}(\Sigma^{x_2} \cup CS^{t-2}_{x_1})$ *such that there exists* $N \in \mathsf{Nodes}(\mathsf{DialogueTree}(D^{t-2}_r))$ *and* $\mathsf{Status}(N, \mathsf{DialogueTree}(D^{t-2}_r)) = \mathsf{U}$ *and* $\mathsf{Status}(N, \mathsf{DialogueTree}(D^{t-2}_r + \langle x_1, assert, \langle\Phi,\phi\rangle\rangle)) = \mathsf{D}$*. From Lemma 7.5.2, we see that this means that there must be at least one element in the set* $\mathsf{Asserts}_{prn}(D^{t-2}_1, x_2)$*, contradicting the assumption.*□

The next lemma states that if a well-formed pruned tree dialogue terminates at t, then neither agent has a belief in a defeasible rule whose consequent is β such that there is an argument that appears at a node in the dialogue tree whose status is U and from the support of which one can defeasibly derive $\neg\beta$. This is because if an agent did have such a belief, then the set of open moves available to it would not be empty and so it would not make a close move and the dialogue would not terminate at t.

Lemma 7.5.5 *Let* D^t_r *be a well-formed pruned tree dialogue that terminates at* t *with participants* x_1 *and* x_2 *such that* D^t_1 *is a top-dialogue of* D^t_r *and* $\mathsf{Receiver}(m_t) = x_2$*.*

Part 1: *There does not exist* $(\alpha_1 \wedge \ldots \wedge \alpha_n \rightarrow \beta, L) \in \Sigma^{x_1}$ *such that there exists* $\langle\Phi,\phi\rangle \in \mathcal{A}(CS^t_{x_1} \cup CS^t_{x_2})$ *such that* $\Phi \mid\!\sim \neg\beta$ *and there exists* $N \in \mathsf{Nodes}(\mathsf{DialogueTree}(D^t_r))$ *such that* $\mathsf{Label}(N) = \langle\Phi,\phi\rangle$ *and* $\mathsf{Status}(N, \mathsf{DialogueTree}(D^t_r)) = \mathsf{U}$.

Part 2: *There does not exist* $(\alpha_1 \wedge \ldots \wedge \alpha_n \rightarrow \beta, L) \in \Sigma^{x_2}$*, (*$L \in \mathbb{N}$*), such that there exists* $\langle\Phi,\phi\rangle \in \mathcal{A}(CS^t_{x_1} \cup CS^t_{x_2})$ *such that* $\Phi \mid\!\sim \neg\beta$ *and there exists* $N \in \mathsf{Nodes}(\mathsf{DialogueTree}(D^t_r))$ *such that* $\mathsf{Label}(N) = \langle\Phi,\phi\rangle$ *and* $\mathsf{Status}(N, \mathsf{DialogueTree}(D^t_r)) = \mathsf{U}$.

Proof by contradiction:

(Part 1) As D^t_r *terminates at* t *and* $\mathsf{Receiver}(m_t) = x_2$*, it must be the case that agent* x_1 *made the close*

move m_t *(from the definition of terminates, Definition 4.2.8), hence* $\mathsf{Opens}_{prn}(D_1^{t-1}, x_1) = \emptyset$ *(from Lemma 7.3.5).*

Let us assume that there does exist $(\alpha_1 \wedge \ldots \wedge \alpha_n \rightarrow \beta, L) \in \Sigma^{x_1}$ *such that there exists* $\langle \Phi, \phi \rangle \in \mathcal{A}(CS_{x_1}^t \cup CS_{x_2}^t)$ *such that* $\Phi \mid\sim \neg\beta$ *and there exists* $N \in \mathsf{Nodes}(\mathsf{DialogueTree}(D_r^t))$ *such that* $\mathsf{Label}(N) = \langle \Phi, \phi \rangle$ *and* $\mathsf{Status}(N, \mathsf{DialogueTree}(D_r^t)) = \mathsf{U}$. *From Lemmas 7.3.3 and 7.3.4, there exists* $\langle \Phi, \phi \rangle \in \mathcal{A}(CS_{x_1}^{t-1} \cup CS_{x_2}^{t-1})$ *such that* $\Phi \mid\sim \neg\beta$ *and there exists* $N \in \mathsf{Nodes}(\mathsf{DialogueTree}(D_r^{t-1}))$ *such that* $\mathsf{Label}(N) = \langle \Phi, \phi \rangle$ *and* $\mathsf{Status}(N, \mathsf{DialogueTree}(D_r^{t-1})) = \mathsf{U}$. *From Lemma 7.5.3, we see that this means that there must be at least one element in the set* $\mathsf{Opens}_{prn}(D_1^{t-1}, x_1)$*, contradicting the assumption.*

(Part 2) As D_r^t *terminates at* t *and* $\mathsf{Receiver}(m_t) = x_2$*, it must be the case that agent* x_2 *made the close move* m_{t-1} *(from the definition of terminates, Definition 4.2.8), hence* $\mathsf{Opens}_{prn}(D_1^{t-2}, x_2) = \emptyset$ *(from Lemma 7.3.5).*

Let us assume that there does exist $(\alpha_1 \wedge \ldots \wedge \alpha_n \rightarrow \beta, L) \in \Sigma^{x_1}$ *such that there exists* $\langle \Phi, \phi \rangle \in \mathcal{A}(CS_{x_1}^t \cup CS_{x_2}^t)$ *such that* $\Phi \mid\sim \neg\beta$ *and there exists* $N \in \mathsf{Nodes}(\mathsf{DialogueTree}(D_r^t))$ *such that* $\mathsf{Label}(N) = \langle \Phi, \phi \rangle$ *and* $\mathsf{Status}(N, \mathsf{DialogueTree}(D_r^t)) = \mathsf{U}$. *From Lemmas 7.3.3 and 7.3.4, there exists* $\langle \Phi, \phi \rangle \in \mathcal{A}(CS_{x_1}^{t-2} \cup CS_{x_2}^{t-2})$ *such that* $\Phi \mid\sim \neg\beta$ *and there exists* $N \in \mathsf{Nodes}(\mathsf{DialogueTree}(D_r^{t-2}))$ *such that* $\mathsf{Label}(N) = \langle \Phi, \phi \rangle$ *and* $\mathsf{Status}(N, \mathsf{DialogueTree}(D_r^{t-2})) = \mathsf{U}$. *From Lemma 7.5.3, we see that this means that there must be at least one element in the set* $\mathsf{Opens}_{prn}(D_1^{t-2}, x_2)$*, contradicting the assumption.*□

The next lemma states that if a well-formed pruned tree dialogue terminates at t, then there are no arguments that can be constructed from the union of the agents' beliefs that, if asserted, would change the status of any node in the dialogue tree. This follows from the previous two lemmas.

Lemma 7.5.6 *Let* D_r^t *be a well-formed pruned tree dialogue that terminates at* t *with participants* x_1 *and* x_2*. There does not exist an argument* $\langle \Phi, \phi \rangle \in \mathcal{A}(\Sigma^{x_1} \cup \Sigma^{x_2})$ *such that there exists a node* $N \in \mathsf{Nodes}(\mathsf{DialogueTree}(D_r^t))$ *such that* $\mathsf{Status}(N, \mathsf{DialogueTree}(D_r^t)) = \mathsf{U}$ *and* $\mathsf{Status}(N, \mathsf{DialogueTree}(D_r^t + \langle P, assert, \langle \Phi, \phi \rangle \rangle)) = \mathsf{D}$.

Proof by contradiction: *Let us assume that there does exist* $\langle \Phi, \phi \rangle \in \mathcal{A}(\Sigma^P \cup \Sigma^{\overline{P}})$ *such that there exists* $N \in \mathsf{Nodes}(\mathsf{DialogueTree}(D_r^t))$ *such that* $\mathsf{Status}(N, \mathsf{DialogueTree}(D_r^t)) = \mathsf{U}$ *and* $\mathsf{Status}(N, \mathsf{DialogueTree}(D_r^t + \langle P, assert, \langle \Phi, \phi \rangle \rangle)) = \mathsf{D}$. *There are two cases to consider.*

(Case 1) $\Phi = \{(\phi, L)\}$ *in which case* $\langle \Phi, \phi \rangle \in \mathcal{A}(\Sigma^P)$*, for some* $P \in \{x_1, x_2\}$*. From Lemma 5.1.1, we see that* $\langle \Phi, \phi \rangle \in \mathcal{A}(\Sigma^P \cup CS_{\overline{P}}^t)$ *such that there exists* $N \in \mathsf{Nodes}(\mathsf{DialogueTree}(D_r^t))$ *such that* $\mathsf{Status}(N, \mathsf{DialogueTree}(D_r^t)) = \mathsf{U}$ *and* $\mathsf{Status}(N, \mathsf{DialogueTree}(D_r^t + \langle P, assert, \langle \Phi, \phi \rangle \rangle)) = \mathsf{D}$. *However, it has been shown that this cannot be the case (Lemma 7.5.4), contradicting the assumption.*

(Case 2) There must be a rule $(\alpha_1 \wedge \ldots \wedge \alpha_n \rightarrow \beta, L) \in \Phi$*, hence* $(\alpha_1 \wedge \ldots \wedge \alpha_n \rightarrow \beta, L) \in \Sigma^P$*, for some* $P \in \{x_1, x_2\}$*, such that there is a node in the tree that is labelled with an argument, from the support set of which we can defeasibly derive* $\neg\beta$ *and that has status* U*, i.e. there exists* $\langle \Phi', \phi' \rangle \in \mathcal{A}(CS_P^t \cup CS_{\overline{P}}^t)$ *such that there exists* $N \in \mathsf{Nodes}(\mathsf{DialogueTree}(D_r^t))$ *such that* $\mathsf{Status}(N, \mathsf{DialogueTree}(D_r^t)) = \mathsf{U}$*. However, it has been shown that this cannot be the case (Lemma 7.5.5), and so we have contradicted the*

initial assumption. □

I have shown that when a well-formed pruned tree dialogue terminates, there are no more arguments from the union of the participating agents' beliefs that, if asserted, would alter the status of any node. In the following section I will use this result to show that warrant inquiry dialogues produced by the pruned tree strategy are sound and complete.

7.6 Results about soundness and completeness of warrant inquiry dialogues

In this section I show that a warrant inquiry dialogue produced by two agents who are following the pruned tree strategy is both sound and complete. As a reminder, a well-formed pruned tree dialogue is sound if and only if, if it terminates with an argument $\langle \Phi, \phi \rangle$ as its outcome, then the status of the root node of the dialectical tree constructed from the union of the agents' beliefs that has $\langle \Phi, \phi \rangle$ at its root is U (Definition 5.7.1). I now show that all well-formed pruned tree dialogues are sound. This is because the dialectical tree in question is constructed from the union of the agents' beliefs and, when a dialogue terminates, there are no arguments that can be constructed from the union of the agents' beliefs that, if asserted, would alter the status of any node in the dialogue tree.

Theorem 7.6.1 *If D_r^t is a well-formed pruned tree dialogue with participants x_1 and x_2, then D_r^t is sound.*

Proof: *Let us assume that D_r^t terminates at t and $\mathsf{Outcome}_{wi}(D_r^t) = \{\langle \Phi, \phi \rangle\}$ (recall that if this is not the case then D_r^t is trivially sound, Definition 5.7.1). From the definition of warrant inquiry outcome (Definition 4.4.6) we see that this means that $\mathsf{RootArg}(D_r^t) = \langle \Phi, \phi \rangle$. From the definition of a dialogue tree (Definition 4.4.2) we see that $\mathsf{DialogueTree}(D_r^t) = \mathsf{T}_A^\Delta$ where $A = \langle \Phi, \phi \rangle$ and $\Delta = CS_{x_1}^t \cup CS_{x_2}^t$ (i.e. the dialogue tree is equal to the dialectical tree constructed from the union of the commitment stores with the same argument at its root). In order to prove soundness we must compare the dialectical tree T_A^Δ with the dialectical tree $\mathsf{T}_A^{\Delta'}$ where $\Delta' = \Sigma^{x_1} \cup \Sigma^{x_2}$ (i.e. with the dialectical tree which has the same argument at its root but which is constructed from the union of the agents' beliefs). It has been shown that there are no arguments that can be constructed from the union of the agents' beliefs that, if asserted, would alter the status of any node in $\mathsf{DialogueTree}(D_r^t)$ (Lemma 7.5.6). Hence, there are no arguments that can be constructed from the union of the agents' beliefs that, if asserted, would alter the status of any node in the dialectical tree T_A^Δ. This means that there are no arguments that appear in the dialectical tree $\mathsf{T}_A^{\Delta'}$ which, if asserted, would change the status of a node in T_A^Δ, and hence in $\mathsf{DialogueTree}(D_r^t)$. Hence the status of the root node in $\mathsf{DialogueTree}(D_r^t)$ must be the same as the status of the root node in T_A^Δ. As $\mathsf{Outcome}_{wi}(D_r^t) = \{\langle \Phi, \phi \rangle\}$, $\mathsf{Status}(\mathsf{Root}(\mathsf{DialogueTree}(D_r^t)), \mathsf{DialogueTree}(D_r^t)) = \mathsf{U}$ (from the definition of warrant inquiry outcome, Definition 4.4.6), hence, $\mathsf{Status}(\mathsf{Root}(\mathsf{T}_A^\Delta), \mathsf{T}_A^\Delta) = \mathsf{U}$, hence D_r^t is sound (from the definition of warrant inquiry soundness, Definition 5.7.1).* □

A well-formed pruned tree dialogue that terminates at t and has root argument $\langle \Phi, \phi \rangle$ is complete if and only if, if the status of the root node of the dialectical tree that is constructed from the union of the participating agents' beliefs and has $\langle \Phi, \phi \rangle$ at its root is U, then the outcome of the dialogue at t is $\{\langle \Phi, \phi \rangle\}$ (Definition 5.7.2). I now similarly show that all well-formed pruned tree dialogues are complete.

Theorem 7.6.2 *If D_r^t is a well-formed pruned tree dialogue with participants x_1 and x_2, then D_r^t is complete.*

Proof: *Let us assume that D_r^t terminates at t, $\mathsf{RootArg}(D_r^t) = \langle \Phi, \phi \rangle$ and $\mathsf{Status}(\mathsf{Root}(\mathsf{T}_A^\Delta), \mathsf{T}_A^\Delta) = \mathsf{U}$ where $A = \langle \Phi, \phi \rangle$ and $\Delta = \Sigma^{x_1} \cup \Sigma^{x_2}$ (recall that if this is not the case then D_r^t is trivially complete, Definition 5.7.2). As in the previous proof, from the definition of a dialogue tree (Definition 4.4.2) we see that $\mathsf{DialogueTree}(D_r^t) = \mathsf{T}_A^\Delta$ where $A = \langle \Phi, \phi \rangle$ and $\Delta = CS_{x_1}^t \cup CS_{x_2}^t$ (i.e. the dialogue tree is equal to the dialectical tree constructed from the union of the commitment stores with the same argument at its root). In order to prove completeness we must compare the dialectical tree T_A^Δ with the dialectical tree $\mathsf{T}_A^{\Delta'}$ where $\Delta' = \Sigma^{x_1} \cup \Sigma^{x_2}$ (i.e. with the dialectical tree which has the same argument at its root but which is constructed from the union of the agents' beliefs). It has been shown that there are no arguments that can be constructed from the union of the agents' beliefs that, if asserted, would alter the status of any node in $\mathsf{DialogueTree}(D_r^t)$ (Lemma 7.5.6). Hence, there are no arguments that can be constructed from the union of the agents' beliefs that, if asserted, would alter the status of any node in the dialectical tree T_A^Δ. This means that there are no arguments that appear in the dialectical tree $\mathsf{T}_A^{\Delta'}$ which, if asserted, would change the status of a node in T_A^Δ, and hence in $\mathsf{DialogueTree}(D_r^t)$. Hence the status of the root node in $\mathsf{DialogueTree}(D_r^t)$ must be the same as the status of the root node in T_A^Δ. Hence, as $\mathsf{Status}(\mathsf{Root}(\mathsf{T}_A^\Delta), \mathsf{T}_A^\Delta) = U$ it must be the case that $\mathsf{Status}(\mathsf{Root}(\mathsf{DialogueTree}(D_r^t)), \mathsf{DialogueTree}(D_r^t)) = U$. From the definition of warrant inquiry outcome (Definition 4.4.6) we get that $\mathsf{Outcome}_{wi}(D_r^t) = \{\langle \Phi, \phi \rangle\}$, hence D_r^t is complete (from the definition of warrant inquiry completeness, Definition 5.7.2).* $\square$

I have shown here the desired result that warrant inquiry dialogues produced by the pruned tree strategy are sound and complete. In the next section I will show that the number of nodes in such a dialogue tree is never more than, and is sometimes less than, the number of nodes in the dialectical tree constructed from the union of the participating agents' beliefs that has the same argument at its root.

7.7 Results about the number of nodes in the dialogue tree

In this section I will discuss the size of the dialogue tree produced during a warrant inquiry dialogue in which the participants are following the pruned tree strategy. In particular, I first show that such a dialogue tree never has more nodes than there are in the dialectical tree constructed from the union of the participating agents' beliefs that has the same argument at its root. I then show that there exist some cases in which there are fewer nodes in the dialogue tree produced by the pruned tree strategy than there are in the relevant dialectical tree.

The following lemma shows that the number of nodes in a dialogue tree produced by two agents following the pruned tree strategy is never greater than the number of nodes that appear in a dialectical tree that is constructed from the union of the agents' beliefs and that has the same argument at its root as the dialogue tree. This is clear as I have already shown that if a path appears in the dialogue tree then it also appears in the dialectical tree constructed from the agents' beliefs.

Lemma 7.7.1 *If D_r^t is a well-formed pruned tree dialogue with participants x_1 and x_2, then* $|\ \mathsf{Nodes}(\mathsf{DialogueTree}(D_r^t))\ | \leq |\ \mathsf{Nodes}(\mathsf{T}_A^{\Delta})\ |$ *where* $A = \mathsf{RootArg}(\mathsf{DialogueTree}(D_r^t))$ *and* $\Delta = \Sigma^{x_1} \cup \Sigma^{x_2}$.

Proof: *It has previously been shown that if a path appears in* $\mathsf{DialogueTree}(D_r^t)$*, then it appears in the dialectical tree* T_A^{Δ} *(Lemma 7.5.1), hence* $|\ \mathsf{Nodes}(\mathsf{DialogueTree}(D_r^t))\ | \leq |\ \mathsf{Nodes}(\mathsf{T}_A^{\Delta})\ |$. □

In the next lemma I show, by example, that sometimes the dialogue tree produced by two agents following the pruned tree strategy contains less nodes than appear in a dialectical tree that is constructed from the union of the agents' beliefs and that has the same argument at its root as the dialogue tree

Lemma 7.7.2 *There exists a well-formed pruned tree dialogue D_r^t with participants x_1 and x_2, such that* $|\ \mathsf{Nodes}(\mathsf{DialogueTree}(D_r^t))\ | < |\ \mathsf{Nodes}(\mathsf{T}_A^{\Delta})\ |$, *where* $A = \mathsf{RootArg}(\mathsf{DialogueTree}(D_r^t))$ *and* $\Delta = \Sigma^{x_1} \cup \Sigma^{x_2}$.

Proof: *Consider the examples of warrant inquiry dialogues given in Sections 6.1.1, 6.1.3, and 6.1.5. In all of these examples the number of nodes in the dialogue tree produced is less than the number of nodes in the equivalent dialectical tree constructed from the union of the agents' beliefs.* □

In the next section I will use these results to show that warrant inquiry dialogues produced by the pruned tree strategy are, in a sense, more efficient than those produced by the exhaustive strategy. If we consider two agents following the pruned tree strategy, and they conduct a warrant inquiry dialogue with topic ϕ, I will show that they will produce a dialogue tree that is not bigger than the dialogue tree produced by the same two agents (i.e. their beliefs do not change) when conducting a warrant inquiry dialogue with topic ϕ but following the exhaustive strategy. I will then show that the dialogue tree produced by the pruned tree strategy is sometimes smaller (has less nodes) than the dialogue tree produced by the exhaustive strategy in the same situation.

7.8 Comparison of dialogue tree with that produced by the exhaustive strategy

I am using the number of nodes in a dialogue tree as a measure of efficiency, as I am interested in redundancy in dialogue trees. However, it is not necessarily the case that dialogue trees with less nodes are preferable. For example, when making the decision about whether or not a patient should be given chemotherapy it would be desirable to be able to present to them all the arguments involved. This is a decision that the patient makes with the doctor and so they should be made aware of all the different factors involved, especially as some of the arguments will depend on the personal values of the patient. So, in this situation, it would be desirable to present the patient with the dialogue tree produced by the exhaustive strategy. However, consider the situation in which it has been decided to give the patient chemotherapy but now the decision as to what chemotherapy to give the patient must be made. This is not normally something that the patient has any input into, and so we would not be as interested in

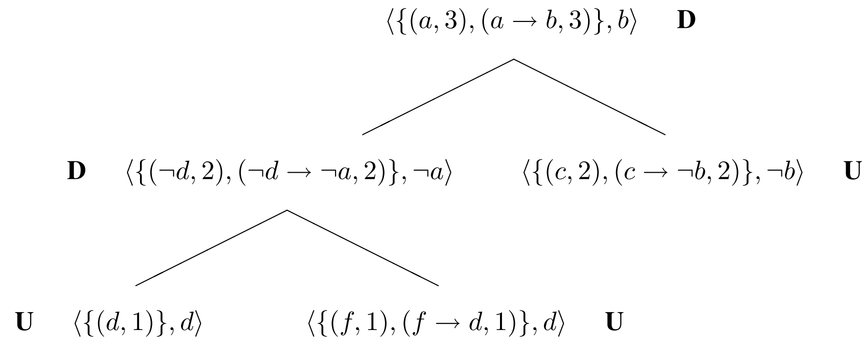


Figure 7.1: The marked dialogue tree for Example 7.8.1.

presenting all the arguments to them and would perhaps prefer a smaller tree that leads to the same outcome.

I hoped to be able to show in this section that, given a dialogue tree produced by two agents following the pruned tree strategy, it is never the case that if a node in the dialogue tree has status D, then it has more than one child whose status is U. However, this is not always the case, as I show in the following example.

Example 7.8.1 *In the following example we have an agent x_1 who wishes to enter into a dialogue with agent x_2 in order to try to find a warrant for an argument for b. We have*

$$\Sigma^{x_1} = \{(\neg d,2),(f,1),(\neg d \to \neg a,2),(f \to d,1),(d \wedge e \to \neg c,1)\}$$

$$\Sigma^{x_2} = \{(a,3),(c,2),(d,1),(a \to b,3),(c \to \neg b,2)\}$$

Agent x_1 opens a warrant inquiry dialogue with b as its topic. The dialogue proceeds as in Table 7.1.

The outcome of the dialogue depends on the dialogue tree produced, that is shown in Figure 7.1. As the status of the root node of the tree is D *the outcome of the dialogue is the empty set,* $\mathsf{Outcome}_{wi}(D_r^t) = \emptyset$*. Note that the defeated node labelled with $\langle\{(\neg d,2),(\neg d \to \neg a,2)\},\neg a\rangle$ in the dialogue tree has two child nodes that are both undefeated.*

Although I cannot show that the types of redundancy described in Chapter 6 are always avoided with the use of the pruned tree strategy, I can show that the dialogue trees produced by the pruned tree strategy are never bigger than those produced by the exhaustive strategy and that sometimes they are smaller. First I must show that if two agents following the pruned tree strategy conduct a warrant inquiry dialogue with topic ϕ, and the same two agents (i.e. their belief bases have not changed) conduct a warrant inquiry dialogue with topic ϕ but following the exhaustive strategy, then the root argument of both dialogues will be the same.

Lemma 7.8.1 *Let D_r^t be a well-formed exhaustive dialogue that terminates at t with participants x_1 and x_2 such that D_r^t is a warrant inquiry dialogue and* $\mathsf{Topic}(D_r^t) = \phi$*. Let $D_{r'}^{t'}$ be a well-formed pruned tree*

t	$CS^t_{x_1}$	m_t	$CS^t_{x_2}$	QS_t
1		$\langle x_2, open, dialogue(wi, b)\rangle$		
2		$\langle x_1, assert, \langle\{(a, 3), (a \to b, 3)\}, b\rangle\rangle$	$(a, 3)$	
			$(a \to b, 3)$	
3	$(\neg d, 2)$	$\langle x_2, assert, \langle\{(\neg d, 2), (\neg d \to \neg a, 2)\}, \neg a\rangle\rangle$		
	$(\neg d \to \neg a, 2)$			
4		$\langle x_1, assert, \langle\{(d, 1)\}, d\rangle\rangle$	$(d, 1)$	
5		$\langle x_2, close, dialogue(wi, b)\rangle$		
6		$\langle x_1, assert, \langle\{(c, 2), (c \to \neg b, 2)\}\rangle\rangle$	$(c, 2)$	
			$(c \to \neg b, 2)$	
7		$\langle x_2, open, dialogue(ai, d \wedge e \to \neg c)\rangle$		$QS_7 = \{d, e, \neg c\}$
8		$\langle x_1, close, dialogue(ai, d \wedge e \to \neg c)\rangle$		
9		$\langle x_2, assert, \langle\{(f, 1), (f \to d, 1)\}, d\rangle\rangle$	$(f, 1)$	
			$(f \to d, 1)$	
10		$\langle x_1, close, dialogue(ai, d \wedge e \to \neg c)\rangle$		
11		$\langle x_2, close, dialogue(ai, d \wedge e \to \neg c)\rangle$		
12		$\langle x_1, open, dialogue(ai, a \to b)\rangle$		$QS_{12} = \{a, b\}$
13		$\langle x_2, close, dialogue(ai, a \to b)\rangle$		
14		$\langle x_1, close, dialogue(ai, a \to b)\rangle$		
15		$\langle x_2, close, dialogue(wi, b)\rangle$		
16		$\langle x_1, close, dialogue(wi, b)\rangle$		

Table 7.1: Dialogue from Example 7.8.1.

dialogue that terminates at t' *with participants* x_1 *and* x_2 *such that* $D_{r'}^{t'}$ *is a warrant inquiry dialogue* $\mathsf{Topic}(D_{r'}^{t'}) = \phi$. $\mathsf{RootArg}(D_r^t) = \mathsf{RootArg}(D_{r'}^{t'})$.

Proof: *In order to show that* $\mathsf{RootArg}(D_r^t) = \mathsf{RootArg}(D_{r'}^{t'})$ *we must show that the first argument asserted for* ϕ *during* D_r^t *is the same as the first argument asserted for* ϕ *during* $D_{r'}^{t'}$. *Consider first the well-formed exhaustive dialogue* D_r^t. *Let us assume that the first argument that is asserted for* ϕ *during* D_r^t *is* $\langle \Phi, \phi \rangle$ *and that this is asserted at* m_s. *The exhaustive strategy (Definition 4.5.4) states that if an assert move* m_s *is made, then (assuming it is* P *who made the move,* $P \in \{x_1, x_2\}$*)* $m_s = \mathsf{Pref}_a(\mathsf{Asserts}_{exh}(D_1^{s-1}, P))$ *(i.e.* m_s *is the most preferred of all the possible legal assert moves) where* $\mathsf{Asserts}_{exh}(D_1^{s-1}, P) = \{\langle \bar(P), assert, \langle \Phi, \phi \rangle \rangle \in \Pi_{wi}^{assert}(D_1^{s-1}, P) \mid \langle \Phi, \phi \rangle \in \mathcal{A}(\Sigma^P \cup CS_{\overline{P}}^t)\}$ *(i.e. the possible legal assert moves consist of all the legal assert moves which can be constructed by the agent making the move). The warrant inquiry protocol (Definition 4.4.3) states that if* $\langle \overline{P}, assert, \langle \Phi, \phi \rangle \rangle \in \Pi_{wi}^{assert}(D_1^{s-1}, P)$*, then* $\mathsf{DialogueTree}(\mathsf{Current}(D_1^{s-1}) + \langle \overline{P}, assert, \langle \Phi, \phi \rangle \rangle) \neq \mathsf{DialogueTree}(\mathsf{Current}(D_1^{s-1})$ *(i.e. asserting the argument* $\langle \Phi, \phi \rangle$ *must cause a new node to be added to the dialogue tree). Now consider the well-formed pruned tree dialogue* $D_{r'}^{t'}$. *Let us assume that the first argument that is asserted for* ϕ *during* $D_{r'}^{t'}$ *is* $\langle \Phi', \phi \rangle$ *and that this is asserted at* $m_{s'}$. *The pruned tree strategy (Definition 6.0.1) states that if an assert move* $m_{s'}$ *is made, then (assuming it was* P' *who made the move,* $P' \in \{x_1, x_2\}$*)* $m_{s'} = \mathsf{Pref}_a(\mathsf{Asserts}_{prn}(D_1^{s'-1}, P))$ *(i.e.* m_s *is the most preferred of all the possible legal assert moves) where*

$\mathsf{Asserts}_{prn}(D_1^{s'-1}, P) = \{\langle \overline{P}, assert, \langle \Phi, \phi \rangle \rangle \in \Pi_{wi}^{assert}(D_1^{s'-1}, P) \mid \langle \Phi, \phi \rangle \in \mathcal{A}(\Sigma^P \cup CS_{\overline{P}}^t)$
and either (1) $\phi = \gamma$ *or*
(2) there exists $N \in \mathsf{Nodes}(\mathsf{DialogueTree}(D_r^{s'-1}))$ *such that*
$[\mathsf{Status}(N, \mathsf{DialogueTree}(D_r^{s'-1})) = \mathsf{U}$ *and*
$\mathsf{Status}(N, \mathsf{DialogueTree}(D_r^{s'-1} + \langle \overline{P}, assert, \langle \Phi, \phi \rangle \rangle)) = \mathsf{D}$ *and*
for all $\langle \overline{P}, assert, \langle \Phi', \phi' \rangle \rangle \in \Pi_{wi}^{assert}(D_1^{s'-1}, P)$ *such that*
$\langle \Phi', \phi' \rangle \in \mathcal{A}(\Sigma^P \cup CS_{\overline{P}}^t)$
[*if* $N' \in \mathsf{Nodes}(\mathsf{DialogueTree}(D_r^{s'-1}))$ *such that* $\mathsf{Status}(N', \mathsf{DialogueTree}(D_r^{s'-1})) = \mathsf{U}$
and $\mathsf{Status}(N', \mathsf{DialogueTree}(D_r^{s'-1} + \langle \overline{P}, assert, \langle \Phi', \phi' \rangle \rangle)) = \mathsf{D}$,
then $\mathsf{Level}(N) \geq \mathsf{Level}(N')]]\}$

where
$\mathsf{cTopic}(D_1^{s'-1}) = \gamma$ *and* $\mathsf{Current}(D_1^{s'-1}) = D_r^{s'-1}$

As m_{s-1} *is the first move asserting an argument for the topic, it must be the case that* $\mathsf{RootArg}(D_r^{s'-1}) = null$ *(from the definition of a root argument, Definition 4.4.1), hence* $\mathsf{DialogueTree}(D_r^{s'-1}) = null$ *(from the definition of a dialogue tree, Definition 4.4.2). Hence, it cannot be the case that there exists* $N \in \mathsf{Nodes}(\mathsf{DialogueTree}(D_r^{s'-1}))$ *and so it must be the case that* ϕ *is the topic of the dialogue (which it is). Therefore the set* $\mathsf{Asserts}_{prn}(D_1^{s'-1}, P)$ *equals the set* $\mathsf{Asserts}_{exh}(D_1^{s-1}, P)$. *Each strategy then uses the same deterministic function to select the most preferred of their respective set of possible legal assert moves, hence they both select the same one. Hence,* $\mathsf{RootArg}(D_r^t) = \mathsf{RootArg}(D_{r'}^{t'})$. □

It was shown in Section 5.6 that the dialogue tree produced by two agents following the exhaustive tree strategy is the same as the dialectical tree constructed from the union of the agents' beliefs that has the same argument at its root. I am now able to use this result with the previous lemma and the first lemma from the previous section to show that if two agents following the pruned tree strategy conduct a dialogue with topic ϕ, then the dialogue tree that they produce is never larger than the one that they would produce if following the exhaustive strategy.

Theorem 7.8.1 *Let D_r^t be a well-formed pruned tree dialogue that terminates at t with participants x_1 and x_2 such that $\mathsf{Topic}(D_r^t) = \phi$ and D_r^t is a warrant inquiry dialogue. Let $D_{r'}^{t'}$ be a well-formed exhaustive dialogue that terminates at t' with participants x_1 and x_2 such that $\mathsf{Topic}(D_{r'}^{t'}) = \phi$ and $D_{r'}^{t'}$ is a warrant inquiry dialogue. $\mid \mathsf{Nodes}(\mathsf{DialogueTree}(D_r^t)) \mid \leq \mid \mathsf{Nodes}(\mathsf{DialogueTree}(D_{r'}^{t'}) \mid$.*

Proof: *It has been shown that a dialogue tree produced by the exhaustive strategy (i.e. $\mathsf{DialogueTree}(D_{r'}^{t'})$) is equal to the dialectical tree T_A^{Δ} where $A = \mathsf{RootArg}(D_{r'}^{t'})$ and $\Delta = \Sigma^{x_1} \cup \Sigma^{x_2}$ (Theorem 5.6.1). It has been shown that $\mathsf{RootArg}(D_{r'}^{t'}) = \mathsf{RootArg}(D_r^t)$ (Lemma 7.8.1). It has also been shown that $\mid \mathsf{Nodes}(\mathsf{DialogueTree}(D_r^t)) \mid \leq \mid \mathsf{Nodes}(\mathsf{T}_{A'}^{\Delta}) \mid$, where $A' = \mathsf{RootArg}(D_r^t)$ and $\Delta = \Sigma^P \cup \Sigma^{\overline{P}}$ (Lemma 7.7.1). $A = A'$, hence, $\mid \mathsf{Nodes}(\mathsf{DialogueTree}(D_r^t)) \mid \leq \mid \mathsf{Nodes}(\mathsf{DialogueTree}(D_{r'}^{t'}) \mid$. $\square$*

I now show, by example, that there exist circumstances in which the dialogue tree produced by the pruned tree strategy has less nodes than the one produced by the exhaustive strategy.

Theorem 7.8.2 *There exists a well-formed pruned tree dialogue D_r^t with participants x_1 and x_2 and there exists a well-formed exhaustive dialogue $D_{r'}^{t'}$ with the same participants x_1 and x_2 such that*

- *D_r^t and $D_{r'}^{t'}$ are both warrant inquiry dialogues,*
- *$\mathsf{Topic}(D_r^t) = \mathsf{Topic}(D_{r'}^{t'})$,*
- *$\mid \mathsf{Nodes}(\mathsf{DialogueTree}(D_r^t)) \mid < \mid \mathsf{Nodes}(\mathsf{DialogueTree}(D_{r'}^{t'}) \mid$.*

Proof by example: *Consider the example of a well-formed exhaustive warrant inquiry dialogue given in Section 4.7.13 and the example of a well-formed pruned tree warrant inquiry dialogue given in Section 6.1.5. Both dialogues have the same topic and the same participants, but the dialogue tree produced in Section 4.7.13 has two fewer nodes than in the dialogue tree produced in Section 6.1.5. $\square$*

7.9 Summary

In this chapter I have shown that warrant inquiry dialogues produced by the pruned tree strategy are sound and complete. Although I was not able to show that the types of redundancy I discussed in Chapter 6 are always avoided with the pruned tree strategy, I have compared the dialogue tree produced with that produced by the exhaustive strategy, and shown that the dialogue tree produced by the pruned tree strategy is never bigger, and sometimes smaller, than the dialogue tree produced by the exhaustive strategy.

Chapter 8

Conclusions

In this chapter I will summarise the contributions made by this work. I will then give a critical assessment of my system and discuss possible future work.

8.1 Contributions made by this work

The work presented here addresses the three main research questions summarised in Section 1.3.

1. *Can I define a system that allows automatic generation of inquiry dialogues between two agents?*
   In Chapter 4, I presented a dialogue system along with a protocol for the argument inquiry dialogue and a protocol for the warrant inquiry dialogue. These protocols (Definitions 4.3.2 and 4.4.3) return the set of legal moves at a point in a dialogue. I provide a specific strategy for use by an agent with either of these protocols, called the exhaustive strategy (Definition 4.5.4), that returns exactly one of the legal moves at a point in a dialogue, which is the move that the agent makes (and so allows automatic dialogue generation). The exhaustive strategy is intelligent as I show that it leads to sound and complete dialogues (see next research question).

2. *Can I propose a benchmark system against which to compare my system, and then show that the dialogues produced by my system are sound and complete in relation to the conclusions drawn by the benchmark system?*
   In Chapter 5, I proposed a benchmark against which to compare the outcome of my dialogues: I compare the outcome of my dialogues with the outcome that a single agent would reach if its beliefs were the union of the participating agents' beliefs. I used this benchmark to define what it means for an argument inquiry dialogue to be sound and complete (Definitions 5.5.1 and 5.5.2), and what it means for a warrant inquiry dialogue to be sound and complete (Definitions 5.7.1 and 5.7.2). I showed that dialogues generated by the exhaustive strategy are both sound and complete (Theorems 5.5.1, 5.5.2, 5.7.1 and 5.7.2).

3. *Can I define a second specific strategy that generates dialogues that produce a smaller output than those generated by the first strategy, and yet are still sound and complete?*
   In Chapter 6, I considered types of redundancy that appear in dialogue trees. I defined another specific strategy, called the pruned tree strategy (Definition 6.0.1), that allows automatic dialogue

generation. In Chapter 7, I showed that dialogues generated by the pruned tree strategy are both sound and complete (Theorems 7.6.1 and 7.6.2). I then showed that a dialogue tree produced by the pruned tree strategy is never larger, and is sometimes smaller, than that which would be produced if the agents were instead following the exhaustive strategy (Theorems 7.8.1 and 7.8.2).

My system goes some way towards providing what we need for two agents in the medical domain to have useful dialogues. In Section 2.1, I gave four desirable features of a dialogue system for the medical domain.

- **Provides inquiry protocol.** I chose to focus my attentions on the inquiry dialogue as it is a cooperative dialogue that embodies one of the more general goals of the medical domain—making a justified claim, such as providing reasons for why a patient should be urgently referred to a specialist. It is also one of the dialogue types to receive the least attention in the literature so far.

- **Generative.** I am interested in defining a practical system that will allow two agents to automatically generate a dialogue. For a dialogue system to be generative it must specify exactly one move to be made at any point in the dialogue.

- **Formally specified.** I want my system to be of use in the real world. Specifying such a system formally should remove any ambiguity about how the protocol should be followed and will facilitate the investigation of the properties of the system.

- **Provides results about dialogue outcome.** As I am concerned with designing a theory that may be used in the medical domain, it is important that the behaviour of the system is well-understood and suitable to the domain. This means that it needs to be certain that the system is going to behave in the intended manner. In particular, I am interested in results about the outcome of the dialogue and need to know that a dialogue system is always going to produce the desired outcome in any given situation.

I discussed, in Chapter 2, how no existing dialogue system provided all four of these features. The system I have presented here does provide all of these features, as I will now show.

- **Provides inquiry protocol.** I have provided two protocols for two different types of inquiry dialogue, the argument inquiry protocol (Definition 4.3.2) and the warrant inquiry protocol (Definition 4.4.3).

- **Generative.** My system is generative as I have provided two strategies, each of which returns exactly one of the legal moves at any point in a dialogue. These strategies are the exhaustive strategy (Definition 4.5.4) and the pruned tree strategy (Definition 6.0.1).

- **Formally specified.** My system is formally specified in Chapters 3, 4 and 6.

- **Provides results about dialogue outcome.** I have shown that all dialogues generated by my system with either the exhaustive strategy or the pruned tree strategy are sound and complete (Theorems 5.5.1, 5.5.2, 5.7.1, 5.7.2, 7.6.1 and 7.6.2).

To summarise, I have presented a formally specified dialogue system along with protocols for two types of inquiry dialogue between two agents, argument inquiry and warrant inquiry. I have presented a strategy, called the exhaustive strategy, which allows an agent to generate an argument inquiry or a warrant inquiry dialogue. I have presented another strategy, called the pruned tree strategy, which agents may also use to generate a warrant inquiry dialogue. I have defined a benchmark against which to compare my dialogue outcome, and have shown that all dialogues generated by my system are sound and complete in relation to this benchmark. I have also shown that the dialogue tree produced by the pruned tree strategy is never larger than, and sometimes smaller than, the dialogue tree produced by the exhaustive strategy. This is a novel contribution as there are no existing, formally specified dialogue systems that allow automatic generation of inquiry dialogues and provide results about the outcome of such dialogues.

Although my system goes some way to providing what is needed for a medical inquiry dialogue system, it is far from sufficient. In the following section I will discuss the shortcomings of my system and discuss interesting future work.

8.2 Critical assessment of my system and future work

I have addressed the main questions that I intended to address and presented an inquiry dialogue system that provides each of the four main features that I believe are desirable for the medical domain. However, my system is complex, with several interacting components, and in order to provide theorems about the behaviour of my system I had to make several simplifying assumptions.

Firstly, I assume that there are always exactly two agents participating in a dialogue. Whilst dialogues between exactly two agents will indeed be useful in the medical domain, there are occasions when we would wish for more than one agent to be able to take part in a dialogue. For example, in a breast cancer clinic there is commonly a multi-disciplinary meeting once a week. All professionals involved in breast cancer care (including statisticians and pathologists, as well as nurses, oncologists etc.) attend these meetings, and there can be around twenty people present, depending on the size of the clinic. Each individual breast cancer patient is discussed at this meeting and a consensus is reached on what the best treatment method for the patient is. It would be very useful to be able to model and support such meetings, perhaps as several multi-participant, warrant inquiry dialogues, but this would not be possible with my inquiry dialogue system.

As I am concerned with the cooperative medical domain, I assume that there is full trust between the participating agents. If dealing with a less cooperative domain, it would perhaps be interesting to consider a way of weighting new beliefs depending on the trustworthiness of their source. It could also be important to consider what might happen if a rogue agent did get into the system, for example if someone hacked an agent within a medical organisation.

I also make the assumption that an agent's belief base does not change throughout the course of a dialogue. If the dialogues are very quick then perhaps this is not that unrealistic, as an agent would be able to put a hold on anything else it was doing whilst it was participating in a dialogue. However, it is likely that an agent will be carrying out several tasks at the same time. It may be involved in more than one dialogue at once, which may be updating its beliefs. It may also have many sensors to its environment that are also updating its beliefs. This is something that needs to be explored more fully. It is this assumption that allowed me to prove many of the results about my system. It would be very interesting to know how a change in an agent's beliefs during a dialogue would affect the dialogue outcome. For example, if an agent's belief base kept growing during a dialogue, would it be possible to generate infinite dialogues? And what should an agent do if it has cause to remove a belief that it asserted earlier in the dialogue?

It is interesting to note that there is no move in my system for retracting things that an agent has previously asserted. As I am interested in providing a system for a cooperative domain, as all beliefs in my system are defeasible, and, particularly, as I am interested in a growth of knowledge between agents, I have not included such a move. Both of my strategies ensure that any arguments from the union of the agents' beliefs that will have a bearing on the status of any nodes in the dialogue tree will be asserted. If an argument that can be constructed from the agents' commitment stores becomes defeated by another argument in the commitment stores then it does not need to be retracted, as it is clear to both agents that the argument is defeated. If I were to consider dialogues for a more competitive domain, where an agent may want to assert something that it does not actually believe, then it would be useful to consider adding a retract move to the set of moves available to an agent.

My system uses a restricted set of propositional logic. This is not expressive enough for the medical domain. I restricted my language to propositional logic for simplicity of presentation, however it would be possible to follow an approach used by García and Simari [22], and actually use a restricted set of first order logic in which all literals are either ground atoms or ground negated atoms, and so there are no variables in any beliefs. However, it may be the case that this is still not expressive enough for the medical domain. Before using my dialogue system in a real-world setting, it would be necessary to apply the system to real-world data and to test it thouroughly.

This is an important point. It was my intention to provide theoretical results for my system, as I believe it is important that we understand its behaviour, especially as it is intended for the safety-critical domain of breast cancer care. I have not, however, been able to apply my system to any actual data from the medical domain. The complexity of such data makes it hard to extract the defeasible rules needed for argument generation in such a way that they reflect the actual interactions between the real-world knowledge. In order to be able to show soundness and completeness results of my system, I also had to restrict defeasible rules to an ordered pair: a set of literals, the $Body$, that together provide a defeasible reason for believing in the $Head$, a literal. Rules used in real-world arguments do not always take this form. Often, the claim of an argument may include disjunctions, conjunctions and implications, and rules used in such arguments would need to be logically translated to fit the form of my defeasible rule.

Considering applying this system to real-world data raises another issue concerning my system, and in particular the warrant inquiry dialogue. The topic of a warrant inquiry dialogue is a literal. Given this topic and the beliefs of the participating agents, a root argument for the dialogue is asserted. Throughout the course of the dialogue, the agents construct a dialogue tree that has the root argument at its root, which may be found to be defeated or undefeated. However, there may be other arguments for the topic of the dialogue that may be constructed from the union of the agents' beliefs and, as my dialogues are predetermined, it will never be possible to generate a dialogue in which they are the root argument. It would be interesting to provide a new version of the warrant inquiry dialogue that instead has an argument as its topic, which becomes the root argument of the dialogue tree. Let us consider again the referral agent scenario from Section 1.2.1. In order to use my warrant inquiry dialogue in this scenario, the GP agent must have a specific topic in mind with which it opens the dialogue, for example "patient should be urgently referred". Assuming there is one, a root argument will be asserted for this dialogue and, together, the GP agent and the referral agent will decide whether this argument is warranted or not. They may find that this argument is not warranted but there may be another argument that they can assert for the topic of the argument that would be warranted if it were the root argument. However, as my system stands, this argument would never be asserted as the root argument of a warrant inquiry dialogue.

Another issue this scenario raises is how the agents would go about considering different referral options. Imagine that the GP has three choices: patient should be referred urgently, patient should be referred normally, patient should be managed in clinic. As my system stands, the agents would have to open three separate warrant inquiry dialogues, each with one of these options as its topic, and see if any were warranted. Perhaps these should combine to make up a deliberation dialogue, and it would be interesting to define a protocol and strategy for doing so.

Another area that needs serious consideration is whether preference levels are the best way of deciding whether an argument defeats another or not, and, if so, how to assign the preference level of beliefs. Where should these preference levels come from? A global ordering based on the source of the information could perhaps be applied, with data from more reliable sources being more preferred. But do we want this preference ordering to also be defeasible? I believe that including defeasible preferences would make a system such as mine more flexible and more realistic, however, this may be at the expense of being able to provide such soundness and completeness results as I have shown here. There is some work on defeasible preferences (e.g. [43, 57]) and it would be interesting to see if this could be integrated with my system and how it would affect the outcome of the dialogues.

Something that I have not explored is the complexity of my system. My system is complicated and involves many interacting components. At each step in a warrant inquiry dialogue, the agents construct a dialogue tree from the contents of the commitment stores. This is not a trivial task, especially with the constraints on an acceptable argumentation line to consider. There may be some optimisation techniques that could be applied, such as caching a copy of the tree, and it would be interesting to consider these.

It would also be interesting to provide more results for my system. In particular, when comparing warrant inquiry dialogues generated by the pruned tree strategy with those generated by the exhaustive strategy, I have only considered the difference in the number of nodes in the dialogue tree produced by the two different strategies. As I discussed in Section 7.8, whether it is preferable for a dialogue tree to have fewer nodes will depend on the application. I would like to consider other points on which to compare the dialogues generated by the different strategies. For example, is it the case that the pruned tree strategy always leads to a shorter dialogue in terms of the number of moves made?

I would also like to further explore the benchmark against which I compare the outcome of my dialogues. I currently compare the outcome that two agents participating in a dialogue arrive at with that which would be arrived at by a single agent that had as its beliefs the union of the participating agents' beliefs. This seems like an ideal situation as there are no constraints on the sharing of beliefs. However, as discussed in Section 5.7, it is only ideal if we accept that the agents each have the same level of expertise regarding the beliefs. Consider the situation in which a medical student is discussing with a consultant whether an argument regarding a patient's treatment is warranted or not. In this situation, the ideal benchmark might be the outcome that the consultant would reach without taking into account any of the student's beliefs.

When I first started investigating this area there were many important questions that I hoped to address, which have ended up being beyond the scope of this work.

- When should an agent enter into a dialogue?
- Who should an agent enter into a dialogue with?
- What type of dialogue should an agent enter into?
- What topic should the dialogue have?
- How should an agent update its beliefs at the end of, or during, a dialogue?

Although the cognitive coherence approach of Pasquier and Chaib-draa [50, 49, 51] addresses all of these questions, their theory is a general one and they have not been able to give the kind of results about the specific outcome of dialogues that the breast cancer domain requires. I would be interested in further investigating the relationship between the goals of an agent and the pragmatic questions given above.

I would also be interested in applying work done by other groups on formalising strategies to the two strategies I have given here. Both Kakas *et al.* [28, 29] and Amgoud and Hameurlain [1] have presented formalisms for defining dialogue strategies and it would be interesting to see whether the strategies I have defined here could be represented in their formalisms, and what benefits this would give.

8.3 Summary

I have taken as a starting point the breast cancer care domain and, from this, determined some research questions to be addressed and some desirable features that should hold for a dialogue system that addresses these questions. I have formally specified a dialogue system that I have shown addresses the research questions defined and provides the desirable features discussed. Finally, in this chapter, I have identified some shortcomings of this work and discussed future work that it would be interesting to carry out regarding my system.

Appendix A

Table of uniform notation

In this appendix I give a table of uniform notation that appears in this thesis.

Natural numbers	i, j, k, m, n
Set of all natural numbers	$\mathbb{N}$
Cardinality of a set X	$\mid X \mid$
Powerset of a set X	$\wp(X)$
Defeasible rule	$\alpha_1 \wedge \ldots \wedge \alpha_n \rightarrow \beta$
Defeasible fact/Literal	$\alpha, \beta, \alpha_1, \ldots, \beta_1, \ldots, \phi, \psi$
Defeasible rule or defeasible fact	ϕ, ϕ', γ
Examples of propositional atoms	$a, b, c, \ldots$
Preference level	L, L', L_1, L_2, L_3
Belief	$(\phi, L), \phi, \phi'$
Preference level of belief ϕ	$\mathsf{pLevel}(\phi)$
Set of all beliefs	$\mathcal{B}$
Set of all state beliefs	$\mathcal{S}$
Set of all domain beliefs	$\mathcal{R}$
Set of all defeasible facts	$\mathcal{S}^\star$
Set of all defeasible rules	$\mathcal{R}^\star$
Union of set of all defeasible facts and set of all defeasible rules	$\mathcal{B}^\star = \mathcal{S}^\star \cup \mathcal{R}^\star$
Set of agent identifiers	$\mathcal{I}$
Agent	$x, x_1, x_2, P, \overline{P}$
Belief base of agent x	Σ^x
Set of beliefs	$\Psi, \Psi', \Phi, \Phi', \Delta, \Delta', \Upsilon, \Upsilon'$
Defeasible derivation of literal α from set of beliefs Ψ	$\Psi \mid\!\sim \alpha$
Set of all literals that can be defeasibly derived from set of beliefs Ψ	$\mathsf{DefDerivations}(\Psi)$

Argument	$\langle \Phi, \phi \rangle, \langle \Phi', \phi' \rangle, A, A_0, A_1, A_2, \ldots, B_1, B_2, \ldots$
Examples of arguments	$a_1, a_2, a_3 \ldots$
Support of an argument A	$\mathsf{Support}(A)$
Claim of an argument A	$\mathsf{Claim}(A)$
Set of all arguments that can be constructed from a set of beliefs Ψ	$\mathcal{A}(\Psi)$
Argument A_1 is a sub-argument of argument A_2	$A_1 \sqsubseteq A_2$
Argument A_1 is in conflict with argument A_2	$A_1 \bowtie A_2$
Argument A_1 attacks argument A_2 at disagreement sub-argument A_3	$A_1 \rhd A_2(A_3)$
Preference level of argument A	$\mathsf{pLevel}(A)$
Argument A_1 strictly preferred to argument A_2	$A_1 >_p A_2$
Argument A_1 proper defeater for argument A_2	$A_1 \Rightarrow_p A_2$
Argument A_1 blocking defeater for argument A_2	$A_1 \Rightarrow_b A_2$
Node in a tree	N, N_i, N_j
Argument tree	T
Level of a node N	$\mathsf{Level}(N)$
Label of a node N	$\mathsf{Label}(N)$
Set of all nodes in argument tree T	$\mathsf{Nodes}(T)$
Root node of an argument tree T	$\mathsf{Root}(T)$
Argumentation line	$\Lambda, \Lambda_i, \Lambda', \Lambda'_i$
Set of supporting arguments	Λ^S
Set of interfering arguments	Λ^I
Dialectical tree for argument A constructed from set of beliefs Ψ	T^{Ψ}_{A}
Status of node N in dialectical tree T^{Ψ}_{A}	$\mathsf{Status}(N, \mathsf{T}^{\Psi}_{A})$
Member of set $\{ai, wi\}$	θ
Move opening argument inquiry dialogue with defeasible fact γ as its topic whose receiver is agent P	$\langle P, open, dialogue(ai, \gamma) \rangle$
Move opening warrant inquiry dialogue with defeasible rule γ as its topic whose receiver is agent P	$\langle P, open, dialogue(wi, \gamma) \rangle$
Move asserting argument $\langle \Phi, \phi \rangle$ whose receiver is agent P	$\langle P, assert, \langle \Phi, \phi \rangle \rangle$
Move closing argument inquiry dialogue with defeasible fact γ as its topic whose receiver is agent P	$\langle P, close, dialogue(ai, \gamma) \rangle$
Move closing warrant inquiry dialogue with defeasible rule γ as its topic whose receiver is agent P	$\langle P, close, dialogue(wi, \gamma) \rangle$
Set of all moves meeting defined format	$\mathcal{M}$
Move	$m, m_1, \ldots, m', m'_1, \ldots$

Receiver of move m	$\mathsf{Receiver}(m)$
Timepoint	$r, s, t, r_1, r_2, \ldots, s_1, s_2, \ldots, t_1, t_2, \ldots, r', s', t', \ldots$
Dialogue that is sequence of moves $[m_r, \ldots, m_t]$	D_r^t
Topic of dialogue D_r^t	$\mathsf{Topic}(D_r^t)$
Type of dialogue D_r^t	$\mathsf{Type}(D_r^t)$
Set of all dialogues	$\mathcal{D}$
Set of all top-level dialogues	$\mathcal{D}_{top}$
Extension of dialogue D_r^t by move m	$D_r^t + m$
Current dialogue of dialogue D_r^t	$\mathsf{Current}(D_r^t)$
Topic of current dialogue of dialogue D_r^t	$\mathsf{cTopic}(D_r^t)$
Type of current dialogue of dialogue D_r^t	$\mathsf{cType}(D_r^t)$
Commitment store associated with agent x at timepoint t	CS_x^t
Question store for timepoint r	QS_r
Question store of current dialogue of dialogue D_r^t	$\mathsf{cQS}(D_r^t)$
Set of moves $\{m_1, \ldots, m_n\}$ that agent P can make such that dialogue extension $D_1^t + m_i$, where $m_i \in \{m_1, \ldots, m_n\}$, is a well-formed dialogue and $\mathsf{cType}(D_1^t) = ai$	$\Pi_{ai}(D_1^t, P)$
Set of moves $\{m_1, \ldots, m_n\}$ that agent P can make such that dialogue extension $D_1^t + m_i$, where $m_i \in \{m_1, \ldots, m_n\}$, is a well-formed dialogue and m_i is of the form $\langle \overline{P}, assert, \langle \Phi, \phi \rangle \rangle$ and $\mathsf{cType}(D_1^t) = ai$	$\Pi_{ai}^{assert}(D_1^t, P)$
Set of moves $\{m_1, \ldots, m_n\}$ that agent P can make such that dialogue extension $D_1^t + m_i$, where $m_i \in \{m_1, \ldots, m_n\}$, is a well-formed dialogue and m_i is of the form $\langle \overline{P}, open, dialogue(ai, \gamma) \rangle$ and $\mathsf{cType}(D_1^t) = ai$	$\Pi_{ai}^{open}(D_1^t, P)$
Set of all well-formed argument inquiry dialogues	$\mathcal{D}_{ai}$
Argument inquiry outcome of well-formed argument inquiry dialogue D_r^t	$\mathsf{Outcome}_{ai}(D_r^t)$
Root argument of warrant inquiry dialogue D_r^t	$\mathsf{RootArg}(D_r^t)$
Dialogue tree associated with warrant inquiry dialogue D_r^t	$\mathsf{T}(D_r^t)$
Set of moves $\{m_1, \ldots, m_n\}$ that agent P can make such that dialogue extension $D_1^t + m_i$, where $m_i \in \{m_1, \ldots, m_n\}$, is a well-formed dialogue and $\mathsf{cType}(D_1^t) = wi$	$\Pi_{wi}(D_1^t, P)$
Set of moves $\{m_1, \ldots, m_n\}$ that agent P can make such that dialogue extension $D_1^t + m_i$, where $m_i \in \{m_1, \ldots, m_n\}$, is a well-formed dialogue and m_i is of the form $\langle \overline{P}, assert, \langle \Phi, \phi \rangle \rangle$ and $\mathsf{cType}(D_1^t) = wi$	$\Pi_{wi}^{assert}(D_1^t, P)$

Set of moves $\{m_1, \ldots, m_n\}$ that agent P can make such that dialogue extension $D^t_1 + m_i$, where $m_i \in \{m_1, \ldots, m_n\}$, is a well-formed dialogue and m_i is of the form $\langle \overline{P}, assert, open, dialogue(ai, \gamma) \rangle$ and $\mathsf{cType}(D^t_1) = wi$	$\Pi^{open}_{wi}(D^t_1, P)$
Set of all well-formed warrant inquiry dialogues	$\mathcal{D}_{wi}$
Warrant inquiry outcome of well-formed warrant inquiry dialogue D^t_r	$\mathsf{Outcome}_{wi}(D^t_r)$
Set of moves	Ξ
Preferred open move from a set of moves Ξ	$\mathsf{Pref}_o(\Xi)$
Preferred assert move from a set of moves Ξ	$\mathsf{Pref}_a(\Xi)$
Move returned by exhaustive strategy for agent P participating in a top-level dialogue D^t_1	$\Omega_{exh}(D^t_1, P)$
Set of possible assert moves for dialogue D^t_r	$\mathsf{PossAsserts}(D^t_r)$
Set of assert moves made during dialogue D^t_r in which agents follow exhaustive strategy at all times	$\mathsf{AssertsMade}_{exh}(D^t_r)$
Set of possible open moves for dialogue D^t_r	$\mathsf{PossOpens}(D^t_r)$
Set of open moves made during dialogue D^t_r in which agents follow exhaustive strategy at all times	$\mathsf{OpensMade}_{exh}(D^t_r)$
Set of possible moves for dialogue D^t_r	$\mathsf{PossMoves}(D^t_r)$
Set of moves made during dialogue D^t_r in which agents follow exhaustive strategy at all times	$\mathsf{MovesMade}_{exh}(D^t_r)$
Move returned by pruned tree strategy for agent P participating in a top-level dialogue D^t_1	$\Omega_{prn}(D^t_1, P)$
Set of assert moves made during dialogue D^t_r in which agents follow exhaustive strategy if current dialogue is argument inquiry but follow pruned tree if current dialogue is warrant inquiry	$\mathsf{AssertsMade}_{prn}(D^t_r)$
Set of open moves made during dialogue D^t_r in which agents follow exhaustive strategy if current dialogue is argument inquiry but follow pruned tree if current dialogue is warrant inquiry	$\mathsf{OpensMade}_{prn}(D^t_r)$
Set of moves made during dialogue D^t_r in which agents follow exhaustive strategy if current dialogue is argument inquiry but follow pruned tree if current dialogue is warrant inquiry	$\mathsf{MovesMade}_{prn}(D^t_r)$

Appendix B

Index of definitions

In this appendix I list all of the definitions that appear in this thesis and give a page reference for each.

Appendix C

Index of lemmas and theorems

In this appendix I list all of the lemmas and theorems that appear in this thesis and give a page reference for each.

Lemma 5.5.2. If D_r^t is a well-formed exhaustive argument inquiry dialogue that terminates at t, with participants x_1 and x_2, such that D_1^t is a top-dialogue of D_r^t, $\phi \in QS_r$ and there exists a domain belief $(\alpha_1 \wedge \ldots \wedge \alpha_n \rightarrow \phi, L) \in \Sigma^{x_1} \cup \Sigma^{x_2}$, then there exists t_1, $1 < t_1 < t$, such that $QS_{t_1} = \{\alpha_1, \ldots, \alpha_n, \phi\}$. page 105

Theorem 5.5.2. If D_r^t is a well-formed exhaustive argument inquiry dialogue with participants x_1 and x_2, then D_r^t is complete. page 105

Theorem 5.5.3. Let D_r^t be a well-formed exhaustive argument inquiry dialogue with participants x_1 and x_2. If $\mathsf{Topic}(D_r^t) = \alpha_1 \wedge \ldots \wedge \alpha_n \rightarrow \phi$ and there exists Φ such that $\langle \Phi, \phi \rangle \in \mathcal{A}(\Sigma^{x_1} \cup \Sigma^{x_2})$ then there exists t_1, $1 < t \leq t_1$, such that $D_r^{t_1}$ extends D_r^t and $\langle \Phi, \phi \rangle \in \mathsf{Outcome}(D_r^{t_1})$. page 105

Lemma 5.6.1. Let D_r^t be a well-formed exhaustive warrant inquiry dialogue that terminates at t with participants x_1 and x_2 such that $\mathsf{RootArg}(D_r^t) = \langle \Phi, \phi \rangle$. If there exists a path $[\langle \Phi, \phi \rangle, \langle \Phi_1, \phi_1 \rangle, \ldots, \langle \Phi_n, \phi_n \rangle]$ in $\mathsf{DialogueTree}(D_r^t)$, then there exists a path $[\langle \Phi, \phi \rangle, \langle \Phi_1, \phi_1 \rangle, \ldots, \langle \Phi_n, \phi_n \rangle]$ in the dialectical tree T_A^Δ where $A = \langle \Phi, \phi \rangle$ and $\Delta = \Sigma^{x_1} \cup \Sigma^{x_2}$. page 106

Lemma 5.6.2. Let D_r^t be a well-formed exhaustive warrant inquiry dialogue that terminates at t with participants x_1 and x_2 such that $\mathsf{RootArg}(D_r^t) = \langle \Phi, \phi \rangle$. If D_1^t is a top-dialogue of D_r^t and there exists a path $[\langle \Phi, \phi \rangle, \langle \Phi_1, \phi_1 \rangle, \ldots, \langle \Phi_n, \phi_n \rangle]$ in the dialectical tree T_A^Δ, where $A = \langle \Phi, \phi \rangle$ and $\Delta = \Sigma^{x_1} \cup \Sigma^{x_2}$, then there exists a path $[\langle \Phi, \phi \rangle, \langle \Phi_1, \phi_1 \rangle, \ldots, \langle \Phi_n, \phi_n \rangle]$ in $\mathsf{DialogueTree}(D_r^t)$. page 107

Theorem 5.6.1. If D_r^t is a well-formed exhaustive warrant inquiry dialogue that terminates at t with participants x_1 and x_2 such that $\mathsf{RootArg}(D_r^t) = \langle \Phi, \phi \rangle$, then $\mathsf{DialogueTree}(D_r^t) = \mathsf{T}_A^\Delta$ where $A = \langle \Phi, \phi \rangle$ and $\Delta = \Sigma^{x_1} \cup \Sigma^{x_2}$. page 108

Theorem 5.7.1. If D_r^t is a well-formed exhaustive warrant inquiry dialogue, then D_r^t is sound. page 109

Theorem 5.7.2. If D_r^t is a well-formed exhaustive warrant inquiry dialogue, then D_r^t is complete. page 110

Theorem 5.7.3. Let D_r^t be a well-formed exhaustive warrant inquiry dialogue with participants x_1 and x_2. There exists t' such that $r < t'$, $D_r^{t'}$ extends D_r^t, and if $\mathsf{Outcome}_{wi}(D_r^{t'}) = \{\langle \Phi, \phi \rangle\}$, then $\mathsf{Status}(\mathsf{Root}(\mathsf{T}_A^\Delta), \mathsf{T}_A^\Delta) = \mathsf{U}$ where $A = \langle \Phi, \phi \rangle$ and $\Delta = \Sigma^{x_1} \cup \Sigma^{x_2}$. page 110

Theorem 5.7.4. Let D_r^t be a well-formed exhaustive warrant inquiry dialogue with participants x_1 and x_2. There exists t' such that $r < t'$, $D_r^{t'}$ extends D_r^t, and if $\mathsf{RootArg}(D_r^t) = \langle \Phi, \phi \rangle$ and $\mathsf{Status}(\mathsf{Root}(\mathsf{T}_A^\Delta), \mathsf{T}_A^\Delta) = \mathsf{U}$ where $A = \langle \Phi, \phi \rangle$ and $\Delta = \Sigma^{x_1} \cup \Sigma^{x_2}$, then $\mathsf{Outcome}_{wi}(D_r^t) = \{\langle \Phi, \phi \rangle\}$. page 110

Lemma 7.1.1. Let D_r^t be a well-formed pruned tree dialogue with participants x_1 and x_2. If D_1^t is a top-dialogue of D_r^t and $\Omega_{prn}(D_1^t, \overline{P}) = \langle P, assert, \langle \Phi, \phi \rangle \rangle$, then $\langle \Phi, \phi \rangle \in \mathcal{A}(\Sigma^{\overline{P}} \cup CS_P^t)$. page 126

Lemma 7.3.5. If D_r^t is a well-formed pruned tree dialogue that terminates at t with participants x_1 and x_2, such that $\mathsf{Receiver}(m_{t-1}) = P$, D_r^t extends D_r^{t-1} and D_1^{t-1} is a top-dialogue of D_r^{t-1}, then the set $\mathsf{Asserts}_{prn}(D_1^{t-1}, P) = \emptyset$ and the set $\mathsf{Opens}_{prn}(D_1^{t-1}, P) = \emptyset$. page 133

Lemma 7.4.1. If D_r^t is a well-formed pruned tree dialogue that terminates at t with participants x_1 and x_2 such that $\mathsf{Topic}(D_r^t) = \phi$ and there exists $\langle \Phi, \phi \rangle \in \mathcal{A}(\Sigma^{x_1} \cup \Sigma^{x_2})$, then $\mathsf{RootArg}(D_r^t) \neq null$. page 134

Lemma 7.4.2. If D_r^t is a well-formed pruned tree dialogue with participants x_1 and x_2 such that $\mathsf{Topic}(D_r^t) = \phi$ and there exists $\langle \Phi, \phi \rangle \in \mathcal{A}(\Sigma^{x_1} \cup \Sigma^{x_2})$, then there exists t' such that $D_r^{t'}$ extends D_r^t and $\mathsf{RootArg}(D_r^t) \neq null$. page 134

Lemma 7.5.1. Let D_r^t be a well-formed pruned tree dialogue that terminates at t with participants x_1 and x_2 such that D_1^t is a top-dialogue of D_r^t and $\mathsf{RootArg}(D_r^t) = \langle \Phi, \phi \rangle$. If there exists a path $[\langle \Phi, \phi \rangle, \langle \Phi_1, \phi_1 \rangle, \ldots, \langle \Phi_n, \phi_n \rangle]$ in $\mathsf{DialogueTree}(D_r^t)$, then there exists a path $[\langle \Phi, \phi \rangle, \langle \Phi_1, \phi_1 \rangle, \ldots, \langle \Phi_n, \phi_n \rangle]$ in the dialectical tree T_A^Δ where $A = \langle \Phi, \phi \rangle$ and $\Delta = \Sigma^{x_1} \cup \Sigma^{x_2}$. page 135

Lemma 7.5.2. Let D_r^t be a well-formed pruned tree dialogue with participants x_1 and x_2 such that $\mathsf{RootArg}(D_r^t) \neq null$ and D_1^t is a top-dialogue of D_r^t. If there exists $\langle \Phi, \phi \rangle \in \mathcal{A}(\Sigma^P \cup CS_{\overline{P}}^t)$ such that there exists $N \in \mathsf{Nodes}(\mathsf{DialogueTree}(D_r^t))$ and $\mathsf{Status}(N, \mathsf{DialogueTree}(D_r^t)) = \mathsf{U}$ and $\mathsf{Status}(N, \mathsf{DialogueTree}(D_r^t + \langle P, assert, \langle \Phi, \phi \rangle \rangle)) = \mathsf{D}$, then $\mathsf{Asserts}_{prn}(D_1^t, P) \neq \emptyset$. page 136

Lemma 7.5.3. Let D_r^t be a well-formed pruned tree dialogue with participants x_1 and x_2 such that $\mathsf{RootArg}(D_r^t) \neq null$ and D_1^t is a top-dialogue of D_r^t. If there exists $(\alpha_1 \wedge \ldots \wedge \alpha_n \rightarrow \beta, L) \in \Sigma^P$ such that there exists $\langle \Phi, \phi \rangle \in \mathcal{A}(CS_P^{t-1} \cup CS_{\overline{P}}^{t-1})$ such that $\Phi \mid\sim \neg\beta$ and there exists $N \in \mathsf{Nodes}(\mathsf{DialogueTree}(D_r^{t-1}))$ such that $\mathsf{Label}(N) = \langle \Phi, \phi \rangle$ and $\mathsf{Status}(N, \mathsf{DialogueTree}(D_r^{t-1})) = \mathsf{U}$, then $\mathsf{Opens}_{prn}(D_1^t, P) \neq \emptyset$. page 136

Lemma 7.5.4. Let D_r^t be a well-formed pruned tree dialogue that terminates at t with participants x_1 and x_2 such that D_1^t is a top-dialogue of D_r^t and $\mathsf{Receiver}(m_t) = x_2$. Part 1: There does not exist $\langle \Phi, \phi \rangle \in \mathcal{A}(\Sigma^{x_1} \cup CS_{x_2}^t)$ such that there exists $N \in \mathsf{Nodes}(\mathsf{DialogueTree}(D_r^t))$ and $\mathsf{Status}(N, \mathsf{DialogueTree}(D_r^t)) = \mathsf{U}$ and $\mathsf{Status}(N, \mathsf{DialogueTree}(D_r^t + \langle x_2, assert, \langle \Phi, \phi \rangle \rangle)) = \mathsf{D}$. Part 2: There does not exist $\langle \Phi, \phi \rangle \in \mathcal{A}(\Sigma^{x_2} \cup CS_{x_1}^t)$ such that there exists $N \in \mathsf{Nodes}(\mathsf{DialogueTree}(D_r^t))$ and $\mathsf{Status}(N, \mathsf{DialogueTree}(D_r^t)) = \mathsf{U}$ and $\mathsf{Status}(N, \mathsf{DialogueTree}(D_r^t + \langle x_1, assert, \langle \Phi, \phi \rangle \rangle)) = \mathsf{D}$. page 137

Lemma 7.5.5. Let D_r^t be a well-formed pruned tree dialogue that terminates at t with participants x_1 and x_2 such that D_1^t is a top-dialogue of D_r^t and $\mathsf{Receiver}(m_t) = x_2$. Part 1: There does not exist $(\alpha_1 \wedge \ldots \wedge \alpha_n \rightarrow \beta, L) \in \Sigma^{x_1}$ such that there exists $\langle \Phi, \phi \rangle \in \mathcal{A}(CS_{x_1}^t \cup CS_{x_2}^t)$ such that $\Phi \mid\!\sim \neg\beta$ and there exists $N \in \mathsf{Nodes}(\mathsf{DialogueTree}(D_r^t))$ such that $\mathsf{Label}(N) = \langle \Phi, \phi \rangle$ and $\mathsf{Status}(N, \mathsf{DialogueTree}(D_r^t)) = \mathsf{U}$. Part 2: There does not exist $(\alpha_1 \wedge \ldots \wedge \alpha_n \rightarrow \beta, L) \in \Sigma^{x_2}$, $(L \in \mathbb{N})$, such that there exists $\langle \Phi, \phi \rangle \in \mathcal{A}(CS_{x_1}^t \cup CS_{x_2}^t)$ such that $\Phi \mid\!\sim \neg\beta$ and there exists $N \in \mathsf{Nodes}(\mathsf{DialogueTree}(D_r^t))$ such that $\mathsf{Label}(N) = \langle \Phi, \phi \rangle$ and $\mathsf{Status}(N, \mathsf{DialogueTree}(D_r^t)) = \mathsf{U}$. page 138

Lemma 7.5.6. Let D_r^t be a well-formed pruned tree dialogue that terminates at t with participants x_1 and x_2. There does not exist an argument $\langle \Phi, \phi \rangle \in \mathcal{A}(\Sigma^{x_1} \cup \Sigma^{x_2})$ such that there exists a node $N \in \mathsf{Nodes}(\mathsf{DialogueTree}(D_r^t))$ such that $\mathsf{Status}(N, \mathsf{DialogueTree}(D_r^t)) = \mathsf{U}$ and $\mathsf{Status}(N, \mathsf{DialogueTree}(D_r^t + \langle P, assert, \langle \Phi, \phi \rangle \rangle)) = \mathsf{D}$. page 139

Theorem 7.6.1. If D_r^t is a well-formed pruned tree dialogue with participants x_1 and x_2, then D_r^t is sound. page 139

Theorem 7.6.2. If D_r^t is a well-formed pruned tree dialogue with participants x_1 and x_2, then D_r^t is complete. page 140

Lemma 7.7.1. If D_r^t is a well-formed pruned tree dialogue with participants x_1 and x_2, then $\mid \mathsf{Nodes}(\mathsf{DialogueTree}(D_r^t)) \mid \leq \mid \mathsf{Nodes}(\mathsf{T}_A^{\Delta}) \mid$ where $A = \mathsf{RootArg}(\mathsf{DialogueTree}(D_r^t))$ and $\Delta = \Sigma^P \cup \Sigma^{\overline{P}}$. page 141

Lemma 7.7.2. There exists a well-formed pruned tree dialogue D_r^t with participants x_1 and x_2, such that $\mid \mathsf{Nodes}(\mathsf{DialogueTree}(D_r^t)) \mid < \mid \mathsf{Nodes}(\mathsf{T}_A^{\Delta}) \mid$, where $A = \mathsf{RootArg}(\mathsf{DialogueTree}(D_r^t))$ and $\Delta = \Sigma^P \cup \Sigma^{\overline{P}}$. page 141

Lemma 7.8.1. Let D_r^t be a well-formed exhaustive dialogue that terminates at t with participants x_1 and x_2 such that D_r^t is a warrant inquiry dialogue and $\mathsf{Topic}(D_r^t) = \phi$. Let $D_{r'}^{t'}$ be a well-formed pruned tree dialogue that terminates at t' with participants x_1 and x_2 such that $D_{r'}^{t'}$ is a warrant inquiry dialogue $\mathsf{Topic}(D_{r'}^{t'}) = \phi$. $\mathsf{RootArg}(D_r^t) = \mathsf{RootArg}(D_{r'}^{t'})$. page 144

Theorem 7.8.1. Let D_r^t be a well-formed pruned tree dialogue that terminates at t with participants x_1 and x_2 such that $\mathsf{Topic}(D_r^t) = \phi$ and D_r^t is a warrant inquiry dialogue. Let $D_{r'}^{t'}$ be a well-formed pruned tree dialogue that terminates at t' with participants x_1 and x_2 such that $\mathsf{Topic}(D_{r'}^{t'}) = \phi$ and $D_{r'}^{t'}$ is a warrant inquiry dialogue. $\mid \mathsf{Nodes}(\mathsf{DialogueTree}(D_r^t)) \mid \leq \mid \mathsf{Nodes}(\mathsf{DialogueTree}(D_{r'}^{t'}) \mid$. page 145

Theorem 7.8.2. There exists a well-formed pruned tree dialogue D_r^t with participants x_1 and x_2 and there exists a well-formed exhaustive dialogue $D_{r'}^{t'}$ with the same participants x_1 and x_2 such that D_r^t and $D_{r'}^{t'}$ are both warrant inquiry dialogues, $\mathsf{Topic}(D_r^t) = \mathsf{Topic}(D_{r'}^{t'})$, and $\mid \mathsf{Nodes}(\mathsf{DialogueTree}(D_r^t)) \mid < \mid \mathsf{Nodes}(\mathsf{DialogueTree}(D_{r'}^{t'}) \mid$. page 145

Bibliography

[1] Leila Amgoud and Nabil Hameurlain. An argumentation-based approach for dialogue move selection. In *Third International Workshop on Argumentation in Multi-Agent Systems (ARGMAS 2006)*, pages 111–125, Hakodate, Japan, 8 - 12 May 2006.

[2] Leila Amgoud, Nicholas Maudet, and Simon Parsons. Arguments, dialogue and negotiation. In *14th European Conference on Artificial Intelligence (ECAI 2000)*, pages 338–342, Berlin, Germany 2000, 20 - 25 August 2000. IOS Press.

[3] Leila Amgoud, Nicholas Maudet, and Simon Parsons. Modelling dialogues using argumentation. In *Fourth International Conference on Multi-Agent Systems*, pages 31–38, Boston, USA, 7 - 12 July 2000. IEEE Press.

[4] John L. Austin. *How To Do Things with Words*. Oxford University Press, Oxford, UK, 1962.

[5] Trevor J. M. Bench-Capon. Specification and implementation of Toulmin dialogue game. In *Eleventh International Conference on Legal Knowledge-Based Systems (Jurix 1998)*, pages 5–20, Nijmegen: Gerard Noodt Instituut, December 1998.

[6] Philippe Besnard and Anthony Hunter. A logic-based theory of deductive arguments. *Artificial Intelligence*, 128:203–235, 2001.

[7] Philippe Besnard and Anthony Hunter. *Elements of Argumentation*. MIT Press, 2006. In preperation.

[8] Jonathan Bury, John Fox, and David Sutton. The PRO*forma* guideline specification language: Progress and prospects. In *First European Workshop on Computer-based Support for Clinical Guidelines and Protocols (EWGLP 2000)*, Leipzig, 13-14 November 2000. IOS Press.

[9] Brahim Chaib-draa, Nicholas Maudet, and Marc-Andre Labrie. Request for action reconsidered as a dialogue game based on commitments. In Marc-Philippe Huget, editor, *Communication in Multiagent Systems - Agent Communication Languages and Conversation Policies*, volume 2650 of *Lecture Notes in Computer Science (LNCS) State of the Art Survey*, pages 284–299. Springer-Verlag, July 2003.

[10] David Classen. Clinical decision support systems to improve clinical practice and quality of care. *Journal of the American Medical Association*, 280(15):1360–1361, 1998. Editorial.

[11] Philip R. Cohen and Hector J. Levesque. Communicative actions for artificial agents. In V. Lesser and L. Gasser, editors, *Proceedings of the First International Conference on Multi-Agent Systems (ICMAS 1995)*, pages 65–72, Cambridge, MA, USA, 1995. MIT Press.

[12] Frank Dignum, Barbara Dunin-Keplicz, and Rineke Verbrugge. Dialogue in team formation. In Frank Dignum and Mark Greaves, editors, *Issues in Agent Communication*, pages 264–280. Springer-Verlag, 2000.

[13] Phan Minh Dung. On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games. *Artificial Intelligence*, 77:321–357, 1995.

[14] Paul E. Dunne and Trevor J. M. Bench-Capon. Two party immediate response disputes: Properties and efficiency. *Artificial Intelligence*, 149(2):221–250, 2003.

[15] Paul E. Dunne and Peter McBurney. Optimal utterances in dialogue protocols. In *Second International Conference on Autonomous Agents and Mutli-Agent Systems (AAMAS 2003)*, pages 608–615, New York, NY, USA, 2003. ACM Press.

[16] Martin Eccles and Jeremy Grimshaw, editors. *Clinical Guidelines: From Conception to Use*. Radcliffe Medical Press, Abingdon, UK, 2000.

[17] Tim Finin, Yannis Labrou, and James Mayfield. KQML as an agent communication language. In J. Bradshaw, editor, *Software Agents*, pages 291–316. AAAI/MIT Press, 1997.

[18] Foundation for Intelligent Physical Agents (FIPA). Fipa communicative act library specification. http://www.fipa.org/specs/fipa00037/SC00037J.html, 2002.

[19] NHS Centre for Reviews and Dissemination. Getting evidence into practice. Effective Health Care Bulletin 5(1), 1999.

[20] John Fox and Subrata Das. *Safe and Sound: Artificial Intelligence in Hazardous Applications*. AAAI Press and The MIT Press, 2000.

[21] John Fox, Nicky Johns, and Ali Rahmanzadeh. Disseminating medical knowledge: The PRO*forma* approach. *Artificial Intelligence in Medicine*, 14(1-2):157–181, 1998.

[22] Alejandro J. García and Guillermo R. Simari. Defeasible logic programming an argumentative approach. *Theory and Practice of Logic Programming*, 4(1–2):95–138, 2004.

[23] Mark Greaves, Heather Holmback, and Jeffrey Bradshaw. What is a conversation policy? In Frank Dignum and Mark Greaves, editors, *Issues in Agent Communication*, pages 118–131. Springer-Verlag, 2000.

[24] David Hitchcock. Some principles of rational mutual inquiry. In Frans H. van Eemeren *et al.*, editor, *Second International Conference on Argumentation*, pages 236–243, Amsterdam, The Netherlands, 1991. SICSAT: International Society for the Study of Argumentation.

[25] David Hitchcock, Peter McBurney, and Simon Parsons. A framework for deliberation dialogues. In Hans V. Hansen *et al.*, editor, *Fourth Biennial Conference of the Ontario Society for the Study of Argumentation (OSSA 2001)*, Windsor, Ontario, Canada, 2001.

[26] Joris Hulstijn. *Dialogue Models for Inquiry and Transaction*. PhD thesis, Universiteit Twente, Enschede, The Netherlands, 2000.

[27] Institute of Medicine Institute of Medicine Comittee on Quality of Health Care in America, editor. *Crossing the Quality Chasm: A New Health System for the 21st Century*. National Academies Press, 2001.

[28] Antonis Kakas, Nicolas Maudet, and Pavlos Moraitis. Flexible agent dialogue strategies and societal communication protocols. In Nicholas R. Jennings, Carles Sierra, Liz Sonenberg, and Milind Tambe, editors, *Third International Joint Conference on Autonomous Agents and MultiAgent Systems (AAMAS 2004)*, pages 1434–1435. ACM Press, July 2004. Extended abstract.

[29] Antonis Kakas, Nicolas Maudet, and Pavlos Moraitis. Layered strategies and protocols for argumentation-based agent interaction. In Iyad Rahwan, Pavlos Moraitis, and Chris Reed, editors, *First International Workshop on Argumentation in MultiAgent Systems (ARGMAS 2004)*, Lecture Notes in Artificial Intelligence (LNAI) 3366, pages 66–79, New York, 2004. Springer-Verlag.

[30] Antonis Kakas, Nicolas Maudet, and Pavlos Moraitis. Modular representation of agent interaction rules through argumentation. *Journal of Autonomous Agents and Multiagent Systems*, 11(2):189–206, 2005. Special Issue on Argumentation in Multi-Agent Systems.

[31] Linda T. Kohn, Janet M. Corrigan, and Molla S. Donaldson, editors. *To Err is Human: Building a Safer Health System*. Committee on Quality of Health Care in America. National Academies Press, 2000.

[32] Yannis Labrou. *Semantics for an Agent Communication Language*. PhD thesis, Computer Science and Engineering Department, University of Maryland, Baltimore County, 1996.

[33] Yannis Labrou and Tim Finin. Semantics and conversations for an agent communication language. In Michael Huhns and Munindar Singh, editors, *Readings in Agents*, pages 235–242. Morgan Kaufmann, 1997.

[34] Yannis Labrou and Tim Finin. Semantics for an agent communication language. In Munindar P. Singh, Anand Rao, and Michael J. Wooldridge, editors, *Intelligent Agents IV: Agent Theories, Architectures, and Languages*, volume 1365 of *Lecture Notes in Computer Science*, pages 209–214. Springer-Verlag, Heidelberg, Germany, 1998.

[35] Yannis Labrou and Tim Finin. Short course: Agent communication in the semantic web era, 2003. Fifth European Agent Systems Spring School 2003, Barcelona.

[36] Yannis Labrou, Tim Finin, and Yun Peng. Agent communication languages: The current landscape. *IEEE Intelligent Systems*, 14(2):45–52, 1999.

[37] Jonathan Lomas. Words without action? the production, dissemination, and impact of consensus recommendations. *Annual Review Public Health*, 12:41–65, 1991.

[38] Nicholas Maudet and Brahim Chaib-draa. Commitment-based and dialogue-game based protocols: new trends in agent communication languages. *The Knowledge Engineering Review*, 17(2):157–179, 2002.

[39] Peter McBurney and Simon Parsons. Chance discovery using dialectical argumentation. In *New Frontiers in Artificial Intelligence: Joint JSAI 2001 Workshop Post Proceedings*, Lecture Notes in Artificial Intelligence 2253, pages 414–424. Springer, Berlin, Germany, 2001.

[40] Peter McBurney and Simon Parsons. Representing epistemic uncertainty by means of dialectical argumentation. *Annals of Mathematics and Artificial Intelligence*, 32(1-4):125–169, 2001.

[41] Peter Mcburney and Simon Parsons. Games that agents play: A formal framework for dialogues between autonomous agents. *Journal of Logic, Language and Information*, 11(3):315–334, 2002. Special issue on logic and games.

[42] Peter McBurney, Rogier van Eijk, Simon Parsons, and Leila Amgoud. A dialogue-game protocol for agent purchase negotiations. *Journal of Autonomous Agents and Multi-Agent Systems. (In press).*, 7(3):235–273, 2003.

[43] Sanjay Modgil. Hierarchical argumentation. Accepted for publication in 10th European Conference on Logics in Artificial Intelligence (JELIA 2006), September 2006.

[44] John Nealon and Antonio Moreno. The application of agent technology to health care. In *AgentCities workshop: research in large scale open agent environments*. First International Joint Conference on Autonomous Agents and Multi-Agent Systems (AAMAS 2002), 2002.

[45] OpenClinical. OpenClinical White Paper. http://www.openclinical.org/whitepaper.html, 2000.

[46] Simon Parsons, Michael Wooldridge, and Leila Amgoud. An analysis of formal inter-agent dialogues. In *First International Conference on Autonomous Agents and Mutli-Agent Systems (AAMAS 2002)*, pages 394– 401, New York, 2002. ACM Press.

[47] Simon Parsons, Michael Wooldridge, and Leila Amgoud. On the outcomes of formal inter-agent dialogues. In *Second International Conference on Autonomous Agents and Mutli-Agent Systems (AAMAS 2003)*, pages 616–623, 2003.

[48] Simon Parsons, Michael Wooldridge, and Leila Amgoud. Properties and complexity of some formal inter-agent dialogues. *Journal of Logic and Computation*, 13(3):347–376, 2003. Special issue on computational dialectics.

[49] Philippe Pasquier, Nicolas Andrillon, and Brahim Chaib-draa. An exploration in using cognitive coherence theory to automate BDI agents' communicational behaviour. In *Agent Communication Language and Dialogue Workshop*, pages 37–58. Second International Conference on Autonomous Agents and Mutli-Agent Systems (AAMAS 2003), 2003.

[50] Philippe Pasquier and Brahim Chaib-draa. The cognitive coherence approach for agent communication pragmatics. In *Second International Conference on Autonomous Agents and Mutli-Agent Systems (AAMAS 2003)*, pages 544 – 551. ACM Press, 2003.

[51] Philippe Pasquier and Brahim Chaib-draa. Agent communication pragmatics: the cognitive coherence approach. *Cognitive Systems Research*, 6(4):364–395, 2005.

[52] Philippe Pasquier, Iyad Rahwan, Frank Dignum, and Liz Sonenberg. Argumentation and persuasion in the cognitive coherence theory. In *First International Conference on Computational Models of Argument (COMMA 2006)*, pages 223–234. IOS Press, 2006.

[53] Stanley L. Pestotnik, David C. Classen, R. Scott Evans, and John P. Burke. Implementing antibiotic practice guidelines through computer-assisted decision support: Clinical and financial outcomes. *Annals of Internal Medicine*, 124:884–890, 1996.

[54] Henry Prakken. On dialogue systems with speech acts, arguments, and counterarguments. In M. Ojeda-Aciego, I.P. de Guzman, G. Brewka, and L. Moniz Pereria, editors, *Proceedings of JELIA'2000, The 7th European Workshop on Logic for Artificial Intelligence*, Springer Lecture Notes in AI 1919, pages 239–253. Springer Verlag, 2000.

[55] Henry Prakken. Relating protocols for dynamic dispute with logics for defeasible argumentation. *Synthese*, 127(1, 2):187–219, 2001.

[56] Henry Prakken. Coherence and flexibility in dialogue games for argumentation. *Journal of Logic and Computation*, 15(6):1009–1040, 2005.

[57] Henry Prakken and Giovanni Sartor. Argument-based logic programming with defeasible priorities. *Journal of Applied Non-Classical Logics*, 7:25–75, 1997. Special issue on handling inconsistency in knowledge systems.

[58] Iyad Rahwan, Peter McBurney, and Liz Sonenberg. Towards a theory of negotiation strategy (a preliminary report). In *Fifth Workshop on Game Theoretic and Decision Theoretic Agents (GTDT-2003)*, pages 73–80. Second International Conference on Autonomous Agents and Mutli-Agent Systems (AAMAS 2003), 2003.

[59] Chris Reed. Dialogue frames in agent communications. In *Third International Conference on Multi-Agent Systems (ICMAS 1998)*, pages 246–253. IEE Press, 1998.

[60] Fariba Sadri, Francesca Toni, and Paolo Torroni. Dialogues for negotiation: Agent varieties and dialogue sequences. In John-Jules Meyer and Milind Tambe, editors, *Pre-proceedings of the Eighth*

International Workshop on Agent Theories, Architectures, and Languages (ATAL-2001), pages 69–84, 2001.

[61] Fariba Sadri, Francesca Toni, and Paolo Torroni. Logic agents, dialogues and negotiation: an abductive approach. In Michael Schroeder and Kostat Stathis, editors, *Symposium on Information Agents for E-Commerce (AISB 2001)*. AISB, 2001.

[62] John Searle. *Speech Acts: An Essay in the Philosophy of Language*. Cambridge University Press, Cambridge, UK, 1969.

[63] Venky Shankararaman, Vivian Ambrosiadou, Trushar Panchal, and Brian Robinson. Agents in health care. In Venky Shankararaman, editor, *Workshop on Autonomous Agents in Health Care*, pages 1–11. ACM Press, 2000.

[64] Guillermo R. Simari and Ronald P. Loui. A mathematical treatment of defeasible reasoning and its implementation. *Artificial Intelligence*, 53:125–157, 1992.

[65] Douglas N. Walton and Erik C. W. Krabbe. *Commitment in Dialogue: Basic Concepts of Interpersonal Reasoning*. SUNY Press, 1995.

[66] Michael J. Wooldridge and Nicholas R. Jennings. Intelligent agents: Theory and practice. *Knowledge Engineering Review*, 10(2):115–152, 1995.