5.5 An Abstract Model for Computing Warrant in Skeptical Argumentation Frameworks

An Abstract Model for Computing Warrant in Skeptical Argumentation Frameworks

Carlos Iván Chesñevar
Departament of Computer Science
Universitat de Lleida
Jaume II, 69 – 25001 Lleida, SPAIN
Email: cic@eps.udl.es

Guillermo Ricardo Simari
Dept. of Computer Science and Engineering
Universidad Nacional del Sur
Alem 1253 – 8000 Bahía Blanca, ARGENTINA
Email: grs@cs.uns.edu.ar

Abstract

Abstract argumentation frameworks have played a major role as a way of understanding argument-based inference, resulting in different argument-based semantics. The goal of such semantics is to characterize which are the rationally justified (or warranted) beliefs associated with a given argumentative theory. In order to make such semantics computationally attractive, suitable argument-based proof procedures are required, in which a search space of arguments is examined looking for possible candidates that warrant those beliefs. This paper introduces an abstract approach to model the computation of warrant in a skeptical abstract argumentation framework. We show that such search space can be defined as a lattice, and illustrate how the so-called dialectical constraints can play a role for guiding the efficient computation of warranted arguments.

Keywords: Argumentation, Defeasible Reasoning, Nonmonnotonic Reasoning.

Introduction and Motivations

Over the last ten years, interest in argumentation has expanded dramatically, driven in part by theoretical advances but also by successful demonstrations of a wide range of practical applications. In this context, abstract argumentation frameworks have played a major role as a way of understanding argument-based inference, resulting in different argument-based semantics. In order to compute such semantics, efficient argument-based proof procedures are required for determining when a given argument A is warranted. This involves the analysis of a potentially large search space of candidate arguments related to A by means of an attack relationship.

This paper presents a novel approach to model such search space for warrant computation in a skeptical abstract argumentation framework. We show that such search space can be defined as a lattice, and illustrate how some constraints (called dialectical constraints) can play a role for guiding the efficient computation of warranted arguments.

The rest of this paper is structured as follows. The next Section presents the basic ideas of an abstract argumentation framework with dialectical constraints, which includes several concepts common to most argument-based formalisms. The notion of argumentation line is presented, highlighting its role for modeling so-called dialectical trees as relevant useful data structures for computing warrant. After these preliminaries, the following Section shows how dialectical trees can be used to analyze the search space associated with computing warrants in an argumentation framework. We show that such search space can be represented as a lattice. Subsequently, we devote a Section to go into different criteria which can lead to compute warrant more efficiently on the basis of this lattice characterization. Finally, we discuss some related work and present the main conclusions that have been obtained.

An Abstract Argumentation Framework with Dialectical Constraints

Abstract argumentation frameworks (Dung 1993; Vreeswijk 1997; Jakobovits 1999; Jakobovits & Vermeir 1999) are formalisms for modelling defeasible argumentation in which some components remain unspecified. In such abstract frameworks usually the underlying knowledge representation language, the actual structure of an argument and the notion of attack among arguments are abstracted away, as the emphasis is put on different *argument-based semantics* which are associated with identifying sets of ultimately accepted arguments.

In this paper we are concerned with the study of warrant computation in argumentation systems, with focus on skeptical semantics for argumentation. As a basis for our analysis we will use an abstract argumentation framework (following Dung's seminal approach to abstract argumentation (Dung 1995; 1993)) enriched with the notion of *dialectical constraint*, which will allow us to model distinguished sequences of arguments. The resulting, extended framework will be called an *argumentation theory*.

Definition 1 (Dung 1995; 1993) An argumentation framework Φ is a pair $\langle \mathfrak{Args}, \mathbf{R} \rangle$, where $\mathfrak{Args}$ is a finite set of arguments and $\mathbf{R}$ is a binary relation between arguments such that $\mathbf{R} \subseteq \mathfrak{Args} \times \mathfrak{Args}$. The notation $(\mathcal{A}, \mathcal{B}) \in \mathbf{R}$ (or equivalently $\mathcal{A}\,\mathbf{R}\,\mathcal{B}$) means that $\mathcal{A}$ *attacks* $\mathcal{B}$.

Thus defined, an Argumentation Framework Φ can be seen as a collection of directed graphs (di-graphs) in which nodes correspond to arguments, and an edge between two nodes corresponds to an attack. We will write $\mathfrak{Lines}_\Phi$ to denote the set of all possible sequences of arguments

$[\mathcal{A}_0, \mathcal{A}_1, \mathcal{A}_2, \ldots, \mathcal{A}_k]$ in Φ where for any pair of arguments $\mathcal{A}_i$, $\mathcal{A}_{i+1}$ it holds that $\mathcal{A}_i$ **R** $\mathcal{A}_{i+1}$, with $0 \leq i \leq k-1$. Argumentation lines define a domain onto which different kinds of *constraints* can be defined. As such constraints are related to sequences which resemble an argumentation dialogue between two parties, we call them *dialectical constraints*. Formally:

Definition 2 Let $\Phi = \langle \mathfrak{Args}, \mathbf{R} \rangle$ be an argumentation framework. A *dialectical constraint* $\mathbf{C}$ *in the context of* Φ is any function $\mathbf{C} : \mathfrak{Lines}_\Phi \rightarrow \{True, False\}$.

A dialectical constraint imposes a restriction characterizing when a given argument sequence λ is valid in a framework Φ ($i.e.$, $\mathbf{C}(\lambda) = True$). An argumentation theory is defined by combining an argumentation framework with a particular set of dialectical constraints. Formally:

Definition 3 An *argumentation theory* T (or just a *theory* T) is a pair $(\Phi, \mathbf{DC})$, where Φ is an argumentation framework, and $\mathbf{DC} = \{\mathbf{C}_1, \mathbf{C}_2, \ldots, \mathbf{C}_k\}$ is a finite (possibly empty) set of *dialectical constraints*.

Given a theory $T = (\Phi, \mathbf{DC})$, the intended role of $\mathbf{DC}$ is to avoid *fallacious* reasoning (Aristotle ; Hamblin 1970; Rescher 1977; Walton 1995) by imposing appropriate constraints on argumentation lines to be considered rationally *acceptable*. Such constraints are usually defined on disallowing certain moves which might lead to fallacious situations. Typical constraints to be found in $\mathbf{DC}$ are *non-circularity* (repeating the same argument twice in an argumentation line is forbidden), *commitment* (parties cannnot contradict themselves when advancing arguments), etc. It must be noted that a full formalization for dialectical constraints is outside the scope of this work. We do not claim to be able to identify every one of such constraints either, as they may vary from one particular argumentation framework to another; that is the reason why $\mathbf{DC}$ is included as a parameter in T. In this respect a similar approach is adopted in (Kakas & Toni 1999), where different characterizations of constraints give rise to different logic programming semantics.

Argumentation Lines

As already discussed before, argument games provide a useful form to characterize proof procedures for argumentation logics.[1] Such games model defeasible reasoning as a dispute between two parties (*Proponent* and *Opponent* of a claim), who exchange arguments and counterarguments, generating *dialogues*. A proposition Q is provably justified on the basis of a set of arguments if its proponent has a *winning strategy* for an argument supporting Q, i.e. every counterargument (defeater) advanced by the Opponent can be ultimately defeated by the Proponent. We believe that such argument game was first used in a computational setting in (Simari, Chesñevar, & García 1994a), and similar formalizations have been also applied in other argument-based approaches, e.g. in Prakken-Sartor's framework for argumentation based on logic programming (Prakken & Sartor 1997) and in Defeasible Logic Programming (DeLP) (García & Simari 2004) and its extensions, notably P-DeLP (Chesñevar *et al.* 2004). Dialogues in such argument games have been given different names (dialogue lines, argumentation lines, dispute lines, etc.). A discussion on such aspects of different logical models of argument can be found in (Chesñevar, Maguitman, & Loui 2000; Prakken & Vreeswijk 2002). In what follows we will borrow some basic terminology from (Chesñevar, Simari, & Godo 2005) for our formalization, which will provide the necessary elements for the intended analysis.

[1] See an in-depth discussion in (Prakken 2005).

Definition 4 Let $T = (\Phi, \mathbf{DC})$ be an argumentation theory. An *argumentation line* λ in T is any finite sequence of arguments $[\mathcal{A}_0, \mathcal{A}_1, \ldots, \mathcal{A}_n]$ as defined before. We will say that λ is *rooted in* $\mathcal{A}_0$, and that the *length* of λ is $n+1$, writing $\mid \lambda \mid = s$ to denote that λ has s arguments. We will also write $\mathfrak{Lines}_\mathcal{A}$ to denote the set of all argumentation lines rooted in $\mathcal{A}$ in the theory T.

Definition 5 Let T be an argumentation theory and let $\lambda = [\mathcal{A}_0, \mathcal{A}_1, \ldots, \mathcal{A}_n]$ be an argumentation line in T. Then $\lambda' = [\mathcal{A}_0, \mathcal{A}_1, \mathcal{A}_2, \ldots, \mathcal{A}_k]$, $k \leq n$, will be called an *initial argumentation segment* in λ of length k, denoted $\lfloor \lambda \rfloor_k$. When $k < n$ we will say that λ' is a proper initial argumentation segment in λ. We will use the term *initial segment* to refer to initial argumentation segments when no confusion arises.

Example 1 Consider a theory $T = (\Phi, \mathbf{DC})$, with $\mathbf{DC} = \emptyset$, where the set $\mathfrak{Args}$ is $\{\mathcal{A}_0, \mathcal{A}_1, \mathcal{A}_2, \mathcal{A}_3, \mathcal{A}_4\}$, and assume that the following relationships hold: $\mathcal{A}_1$ defeats $\mathcal{A}_0$, $\mathcal{A}_2$ defeats $\mathcal{A}_0$, $\mathcal{A}_3$ defeats $\mathcal{A}_0$, $\mathcal{A}_4$ defeats $\mathcal{A}_1$. Three different argumentation lines rooted in $\mathcal{A}_0$ can be obtained, namely:

$$\begin{aligned} \lambda_1 &= [\mathcal{A}_0, \mathcal{A}_1, \mathcal{A}_4] \\ \lambda_2 &= [\mathcal{A}_0, \mathcal{A}_1, \mathcal{A}_2] \\ \lambda_3 &= [\mathcal{A}_0, \mathcal{A}_3] \end{aligned}$$

In particular, $\lfloor \lambda_1 \rfloor_2 = [\mathcal{A}_0, \mathcal{A}_1]$ is an initial argumentation segment in λ_1.

Example 2 Consider a theory $T' = (\Phi, \mathbf{DC})$ where the set $\mathfrak{Args}$ is $\{\mathcal{A}_0, \mathcal{A}_1\}$, and assume that the following relationships hold: $\mathcal{A}_0$ defeats $\mathcal{A}_1$, and $\mathcal{A}_1$ defeats $\mathcal{A}_0$. An infinite number of argumentation lines rooted in $\mathcal{A}_0$ can be obtained (e.g. $\lambda_1 = [\mathcal{A}_0]$, $\lambda_2 = [\mathcal{A}_0, \mathcal{A}_1]$, $\lambda_3 = [\mathcal{A}_0, \mathcal{A}_1, \mathcal{A}_0]$, $\lambda_4 = [\mathcal{A}_0, \mathcal{A}_1, \mathcal{A}_0, \mathcal{A}_1]$, etc.).

Remark 1 Note that from Def. 4, given an argumentation line $[\mathcal{A}_0, \mathcal{A}_1, \mathcal{A}_2, \ldots, \mathcal{A}_n]$ every subsequence $[\mathcal{A}_i, \mathcal{A}_{i+1}, \ldots \mathcal{A}_{i+k}]$ with $0 \leq i$, $i+k \leq n$ is also an argumentation line. In particular, every initial argumentation segment is also an argumentation line.

Intuitively, an argumentation line λ is acceptable iff it satisfies every dialectical constraint of the theory it belongs to. Formally:

Definition 6 Given an argumentation theory $T = (\Phi, \mathbf{DC})$, an argumentation line λ is *acceptable* wrt T iff $\mathbf{C}_i(\lambda) = True$, for every $\mathbf{C}_i \in \mathbf{DC}$.

In what follows, we will assume without loss of generality that the notion of acceptability imposed by dialectical constraints is such that if λ is acceptable wrt a theory $T = (\Phi, \mathbf{DC})$, then any subsequence of λ is also acceptable.

Assumption 1 If λ is an acceptable argumentation line wrt a theory $T = (\Phi, \mathbf{DC})$, then any subsequence of λ is also acceptable wrt T.

Example 3 Consider the theory T' in Ex. 2, and assume that $\mathbf{DC}$={ Repetition of arguments is not allowed }. Then λ_1 and λ_2 are acceptable argumentation lines in T', but λ_3 and λ_4 are not.

Definition 7 Let T be an argumentation theory, and let λ and λ' be two acceptable argumentation lines in T. We will say that λ' *extends* λ in T iff $\lambda = \lfloor\lambda'\rfloor_k$, for some $k < \mid \lambda' \mid$, that is, λ' extends λ iff λ is a proper initial argumentation segment of λ'.

Definition 8 Let T be an argumentation theory, and let λ be an acceptable argumentation line in T. We will say that λ is *exhaustive* if there is no acceptable argumentation line λ' in T such that $\mid \lambda \mid < \mid \lambda' \mid$, and for some k, $\lambda = \lfloor\lambda'\rfloor_k$, that is, there is no λ' such that extends λ. Non-exhaustive argumentation lines will be referred to as *partial* argumentation lines.

Example 4 Consider the theory T presented in Ex. 1. Then λ_1, λ_2 and λ_3 are exhaustive argumentation lines whereas $\lfloor\lambda_1\rfloor_2$ is a partial argumentation line. In the case of the theory T' in Ex. 2, the argumentation line λ_2 extends λ_1. Argumentation line λ_2 is exhaustive, as it cannot be further extended on the basis of T' with the dialectical constraint introduced in Ex. 3.

We will distinguish the set $S = \{\lambda_1, \lambda_2, \ldots, \lambda_k\}$ of argumentation lines rooted in the same initial argument and with the property of not containing lines that are initial subsequences of other lines in the set.

Definition 9 Given a theory T, a set $S = \{\lambda_1, \lambda_2, \ldots, \lambda_n\}$ of argumentation lines rooted in a given argument $\mathcal{A}$, denoted $S_\mathcal{A}$, is called a *bundle set* wrt T iff there is no pair $\lambda_i, \lambda_j \in S_\mathcal{A}$ such that λ_i extends λ_j.

Example 5 Consider the theory $T = (\Phi, \mathbf{DC})$ from Ex. 1, and the argumentation lines λ_1, λ_2, and λ_3. Then $S_{\mathcal{A}_0} = \{\lambda_1, \lambda_2, \lambda_3\}$ is a bundle set of argumentation lines wrt T.

As we will see next, a bundle set of argumentation lines rooted in a given argument $\mathcal{A}$ provides the basis for conceptualizing a tree structure called *dialectical tree*.

Dialectical Trees

A bundle set $S_\mathcal{A}$ consists of argumentation lines rooted in a given argument $\mathcal{A}$ which can be "put" together in a tree structure. Formally:

Definition 10 Let T be a theory, and let $\mathcal{A}$ be an argument in T, and let $S_\mathcal{A} = \{\lambda_1, \lambda_2, \ldots, \lambda_n\}$ be a bundle set of argumentation lines rooted in $\mathcal{A}$. Then, the *dialectical tree* rooted in $\mathcal{A}$ *based on* $S_\mathcal{A}$, denoted $\mathcal{T}_\mathcal{A}$, is a tree structure defined as follows:

1. The root node of $\mathcal{T}_\mathcal{A}$ is $\mathcal{A}$.
2. Let F=$\{\mathsf{tail}(\lambda)$, for every $\lambda \in S_\mathcal{A}\}$, and H=$\{\mathsf{head}(\lambda)$, for every $\lambda \in F\}$.[2]
   If $H = \emptyset$ then $\mathcal{T}_\mathcal{A}$ has no subtrees.
   Otherwise, if $H = \{\mathcal{B}_1, \ldots, \mathcal{B}_k\}$, then for every $\mathcal{B}_i \in H$, we define
   $$\textit{getBundle}(\mathcal{B}_i) = \{\lambda \in F \mid \mathsf{head}(\lambda) = \mathcal{B}_i\}$$
   We put $\mathcal{T}_{\mathcal{B}_i}$ as an immediate subtree of $\mathcal{A}$, where $\mathcal{T}_{\mathcal{B}_i}$ is a dialectical tree based on $\textit{getBundle}(\mathcal{B}_i)$.

We will write $\mathfrak{Tree}_\mathcal{A}$ to denote the family of all possible dialectical trees based on $\mathcal{A}$. We will represent as $\mathfrak{Tree}_T$ the family of all possible dialectical trees in the theory T.

Example 6 Consider the theory $T = (\Phi, \mathbf{DC})$ from Ex. 1. In that theory it holds that $S_{\mathcal{A}_0} = \{\lambda_1, \lambda_2, \lambda_3\}$ is a bundle set. Fig. 1(a) shows an associated dialectical tree $\mathcal{T}_{\mathcal{A}_0}$.

The above definition shows how to build a dialectical tree from a bundle set of argumentation lines rooted in a given argument. It is important to note that the "shape" of the resulting tree will depend on the order in which the subtrees are attached. Each possible order will produce a tree with a different geometric configuration. All the differently conformed trees are nevertheless "equivalent" in the sense that they will contain exactly the same argumentation lines as branches from its root to its leaves. This observation is formalized by introducing the following relation which can be trivially shown to be an equivalence relation.

Definition 11 Let T be a theory, and let $\mathfrak{Tree}_\mathcal{A}$ be the set of all possible dialectical trees rooted in an argument $\mathcal{A}$ in theory T. We will say that $\mathcal{T}_\mathcal{A}$ is equivalent to $\mathcal{T}'_\mathcal{A}$, denoted $\mathcal{T}_\mathcal{A} \equiv_\tau \mathcal{T}'_\mathcal{A}$ iff they are obtained from the the same bundle set $S_\mathcal{A}$ of argumentation lines rooted in $\mathcal{A}$.

Given an argument $\mathcal{A}$, there is a one-to-one correspondence between a bundle set $S_\mathcal{A}$ of argumentation lines rooted in $\mathcal{A}$ and the corresponding equivalence class of dialectical trees that share the same bundle set as their origin (as specified in Def. 10). In fact, a dialectical tree $\mathcal{T}_\mathcal{A}$ based

[2] The functions $\mathsf{head}(\cdot)$ and $\mathsf{tail}(\cdot)$ have the usual meaning in list processing.

on $S_{\mathcal{A}}$ is just *an alternative way* of expressing the same information already present in $S_{\mathcal{A}}$. Each member of an equivalence class represents a different way in which a tree could be built. Each particular computational method used to generate the tree from the bundle set will produce one particular member on the equivalence class. In that manner, the equivalence relation will represent a tool for exploring the computational process of warrant and as we will see later, trees provide a powerful way of conceptualize the computation of warranted arguments. Next, we will define mappings which allow to re-formulate a bundle set $S_{\mathcal{A}}$ as a dialectical tree $\mathcal{T}_{\mathcal{A}}$ and viceversa.

Definition 12 Let T be an argumentative theory, and let $S_{\mathcal{A}}$ be a bundle set of argumentation lines rooted in an argument $\mathcal{A}$ of T. We define the mapping

$$\mathbb{T} : \wp(\mathfrak{Lines}_{\mathcal{A}}) \setminus \{\emptyset\} \mapsto \overline{\mathfrak{Tree}_{\mathcal{A}}}$$

as $\mathbb{T}(S_{\mathcal{A}}) =_{def} \overline{\mathcal{T}_{\mathcal{A}}}$, where $\overline{\mathfrak{Tree}_{\mathcal{A}}}$ is the quotient set of $\mathfrak{Tree}_{\mathcal{A}}$ by $\equiv_{\tau}$, and $\overline{\mathcal{T}_{\mathcal{A}}}$ denotes the equivalence class of $\mathcal{T}_{\mathcal{A}}$.

Proposition 1 For any argument $\mathcal{A}$ in an argumentative theory T, the mapping $\mathbb{T}$ is a bijection.[3]

As the mapping $\mathbb{T}$ is a bijection, we can also define the inverse mapping $\mathbb{S} =_{def} \mathbb{T}^{-1}$ which allow us to determine the associated bundle set of argumentation lines corresponding to an arbitrary class of dialectical trees rooted in an argument $\mathcal{A}$.

In what follows, we will use indistinctly a *set notation* (a bundle set of argumentation lines rooted in an argument $\mathcal{A}$) or a *tree notation* (a dialectical tree rooted in $\mathcal{A}$), as the former mappings $\mathbb{S}$ and $\mathbb{T}$ allow us to go from any of these notation to the other.

The following proposition shows that dialectical trees can be thought of as structures in which any subtree $\mathcal{T}'_{\mathcal{A}}$ of a dialectical tree $\mathcal{T}_{\mathcal{A}}$ is also a dialectical tree.

Proposition 2 Let T be a theory, and $\mathcal{T}_{\mathcal{A}}$ a dialectical tree in T. Then it holds that any subtree $\mathcal{T}'_{\mathcal{A}}$ of $\mathcal{T}_{\mathcal{A}}$, rooted in $\mathcal{A}$, is also a dialectical tree wrt T.

Acceptable dialectical trees

The notion of acceptable argumentation line will be used to characterize acceptable dialectical trees, which will be fundamental as a basis for formalizing the computation of warrant in our setting.

Definition 13 Let T be a theory, a dialectical tree $\mathcal{T}_{\mathcal{A}}$ in T is acceptable iff every argumentation line in the associated bundle set $\mathbb{S}(\overline{\mathcal{T}_{\mathcal{A}}})$ is acceptable. We will distinguish the subset $\mathfrak{ATree}_{\mathcal{A}}$ (resp. $\mathfrak{ATree}_{T}$) of all acceptable dialectical trees in $\mathfrak{Tree}_{\mathcal{A}}$ (resp. $\mathfrak{Tree}_{T}$).

As acceptable dialectical trees are a subclass of dialectical trees, all the properties previously shown apply also to them. In the sequel, we will just write “dialectical trees” to refer to acceptable dialectical trees, unless stated otherwise.

[3] Proofs not included for space reasons.

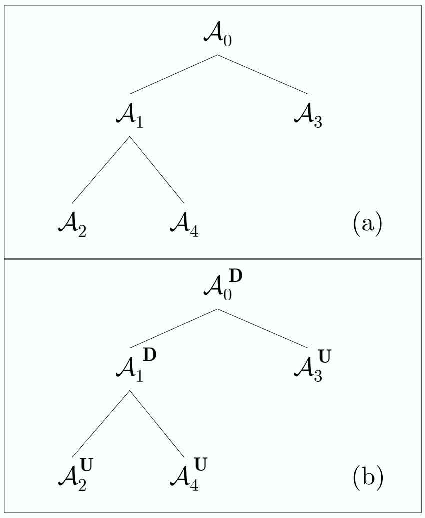

Figure 1: (a) Dialectical tree and (b) marked dialectical tree for Example 6

Definition 14 A dialectical tree $\mathcal{T}_{\mathcal{A}}$ will be called *exhaustive* iff it is constructed from the set $S_{\mathcal{A}}$ of all possible exhaustive argumentation lines rooted in $\mathcal{A}$, otherwise $\mathcal{T}_{\mathcal{A}}$ will be called *partial*.

Besides, the exhaustive dialectical tree for any argument $\mathcal{A}$ can be proven to be unique.

Proposition 3 Let T be a theory, and let $\mathcal{A}$ be an argument in T. Then there exists a unique exhaustive dialectical tree $\mathcal{T}_{\mathcal{A}}$ in T (up to an equivalence wrt $\equiv_{\tau}$ as defined in Def. 11)

Acceptable dialectical trees allow to determine whether the root node of the tree is to be accepted (ultimately *undefeated*) or rejected (ultimately *defeated*) as a rationally justified belief. A *marking function* provides a definition of such acceptance criterion. Formally:

Definition 15 Let T be a theory. A marking criterion for T is a function $Mark : \mathfrak{Tree}_{T} \rightarrow \{D, U\}$. We will write $Mark(\mathcal{T}_i) = U$ (resp. $Mark(\mathcal{T}_i) = D$) to denote that the root node of $\mathcal{T}_i$ is marked as U-node (resp. D-node).

Several marking criteria can be defined for capturing skeptical semantics for argumentation. A particular criterion (which we will later use in our analysis for strategies for computing warrant) is the AND-OR marking of a dialectical tree (Simari, Chesñevar, & García 1994a), which corresponds to Dung’s grounded semantics (Dung 1995).

Definition 16 Let T be a theory, and let $\mathcal{T}_{\mathcal{A}}$ be a dialectical tree. The and-or marking of $\mathcal{T}_{\mathcal{A}}$ is defined as follows:

1. If $\mathcal{T}_{\mathcal{A}}$ has no subtrees, then $Mark(\mathcal{T}_{\mathcal{A}}) = U$.
2. If $\mathcal{T}_{\mathcal{A}}$ has subtrees $\mathcal{T}_1, \ldots, \mathcal{T}_k$ then

(a) *Mark*$(\mathcal{T}_\mathcal{A}) = U$ iff *Mark*$(\mathcal{T}_i) = D$, for all $i = 1 \ldots k$.
(b) *Mark*$(\mathcal{T}_\mathcal{A}) = D$ iff there exists $\mathcal{T}_i$ such that *Mark*$(\mathcal{T}_i) = U$, for some $i = 1 \ldots k$.

Proposition 4 Let T be a theory, and let $\mathcal{T}_\mathcal{A}$ be a dialectical tree. The and-or marking defined in Def. 16 assigns the same mark to all the members of $\overline{\mathcal{T}_\mathcal{A}}$.

Remark 2 As a design criterion, it would be sensible to require that a particular marking criterion would respect that every member of a given equivalence class will be marked in the same way. This will provide an "invariance" of marking with respect to the particular way the algorithm introduced in Def. 10 builds the tree. In such manner, that invariance will allow to work with the bundle set disregarding the circumstantial element of the equivalence class at hand. Each marking procedure is affected by the geometric properties of the tree. For instance, the classical and-or tree traversal will work best with trees that have their shortest branches to the left (Simari, Chesñevar, & García 1994a; Simari, Chesñevar, & García 1994b; Chesñevar, Simari, & Godo 2005), but other procedures could work better on different configurations. Working with the bundle set, and transforming it in a *bundle list* by using some preprocessing algorithm could result in significant speed-ups. Pursuing these observations is outside the scope and length restrictions of this paper, but has been addressed elsewhere (Chesñevar & Simari 2005).

Definition 17 Let T be an argumentative theory and *Mark* a marking criterion for T. An argument $\mathcal{A}$ is a *warranted argument* (or just *warrant*) in T iff the exhaustive dialectical tree $\mathcal{T}_\mathcal{A}$ is such that *Mark*$(\mathcal{T}_\mathcal{A}) = U$.

Example 7 Consider the exhaustive dialectical tree $\mathcal{T}_{\mathcal{A}_0}$ in Ex. 6 shown in Fig. 1(a). Fig. 1(b) shows the corresponding marking by applying Def. 16, showing that $\mathcal{A}_0$ –the root of $\mathcal{T}_{\mathcal{A}_0}$– is an ultimately defeated argument, i.e. $Mark(\mathcal{T}_{\mathcal{A}_0}) = D$. Hence $\mathcal{A}_0$ is not a warranted argument. In Fig. 2 the and-or marking from deep-first, left to right, in (a) will have to traverse the whole tree, meanwhile in (b) only visits two modes. Both trees belong to same equivalent class.

Warrant Computation via Dialectical Trees

As stated in the introduction, our main concern is to model warrant computation in skeptical argumentation frameworks. Fix-point definitions are very expressive declaratively, but tree structures lend themselves naturally to implementation. In fact, some implementations of skeptical argumentation systems (e.g. DeLP (García & Simari 2004)) rely on tree structures (such as dialectical trees) which can be computed by performing backward chaining at two levels. On the one hand, arguments are computed by backward chaining from a query (goal) using a logic programming approach (e.g. SLD resolution). On the other hand, dialectical trees can be computed by recursively analyzing defeaters for

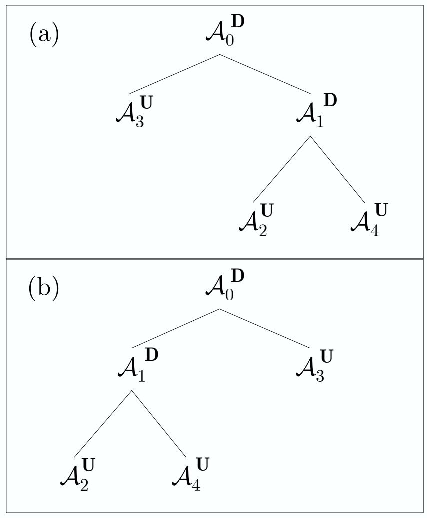

Figure 2: (a) Dialectical tree and (b) Symmetric dialectical tree for Example 6

a given argument, defeaters for those defeaters, and so on. In particular, in more complex and general settings (such as admissibility semantics) dialectical proof procedures have been developed (Dung, Kowalski, & Toni 2006) using a similar strategy to compute warranted belief.

In our abstract model we will use dialectical trees to formalize warrant computation. As indicated in (Chesñevar, Simari, & Godo 2005), the process of building an arbitrary dialectical tree $\mathcal{T}_{\mathcal{A}_0}$ can be thought of as a *computation* starting from an initial tree (consisting of a single node) and evolving into more complex trees by adding new arguments (nodes) stepwise. Elementary steps in this computation can be related by means of a precedence relationship "$\sqsubseteq$" among trees:

Definition 18 Let T be a theory, $\mathcal{A}$ an argument and let $\mathcal{T}_\mathcal{A}$, $\mathcal{T}'_\mathcal{A}$ be acceptable dialectical trees rooted in $\mathcal{A}$. We define a relationship $\sqsubseteq\ \subseteq \mathfrak{Tree}_\mathcal{A} \times \mathfrak{Tree}_\mathcal{A}$. We will write $\mathcal{T}_\mathcal{A} \sqsubset \mathcal{T}'_\mathcal{A}$ whenever $\mathcal{T}'_\mathcal{A}$ can be obtained from $\mathcal{T}_\mathcal{A}$ by extending some argumentation line λ in $\mathcal{T}_\mathcal{A}$ by exactly one argument. As usual, we will write $\mathcal{T}_\mathcal{A} \sqsubseteq \mathcal{T}'_\mathcal{A}$ iff $\mathcal{T}_\mathcal{A} = \mathcal{T}'_\mathcal{A}$ or $\mathcal{T}_\mathcal{A} \sqsubset \mathcal{T}'_\mathcal{A}$. We will also write $\mathcal{T}_\mathcal{A} \sqsubseteq_* \mathcal{T}'_\mathcal{A}$ iff there exists a (possibly empty) sequence $\mathcal{T}_1, \mathcal{T}_2, \ldots, \mathcal{T}_k$ such that $\mathcal{T}_\mathcal{A} = \mathcal{T}_1 \sqsubseteq \ldots \sqsubseteq \mathcal{T}_k = \mathcal{T}'_\mathcal{A}$.

From Def. 18 the notion of exhaustive dialectical tree can be recast as follows: A dialectical tree $\mathcal{T}_i$ is exhaustive iff there is no $\mathcal{T}_j \neq \mathcal{T}_i$ such that $\mathcal{T}_i \sqsubset \mathcal{T}_j$. Every dialectical tree $\mathcal{T}_i$ can be seen as a 'snapshot' of the status of a disputation between two parties (proponent and opponent), and the relationship "$\sqsubseteq$" allows to capture the evolution of such disputation.[4] In particular, note that for any argumentative theory

[4]Note however that $\mathcal{T}_i \sqsubseteq \mathcal{T}_j$ does not imply that one party has advanced some argument in $\mathcal{T}_i$ and the other party has replied in $\mathcal{T}_j$. Thus our framework provides a setup to define *unique*- and

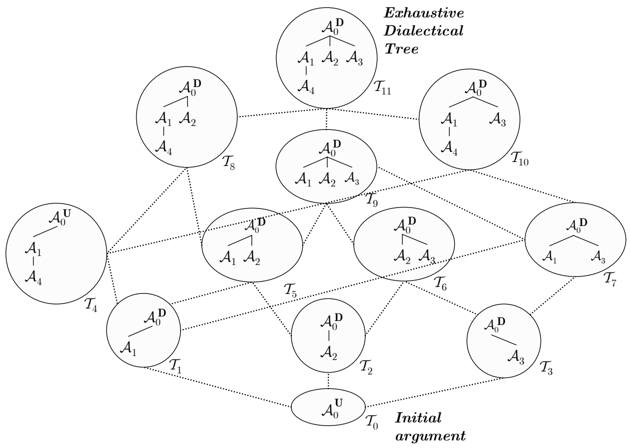

Figure 3: Lattice for all possible dialectical trees rooted in an argument $\mathcal{A}_0$ (Example 8)

T, given an argument $\mathcal{A}$ the ordered set $(\mathfrak{Tree}_{\mathcal{A}}, \sqsubseteq_*)$ is a poset, where the least element is $\mathcal{A}$ and the greatest element is the exhaustive dialectical tree $\mathcal{T}_{\mathcal{A}}$.

We are now concerned with the following question: *can we enumerate all possible ways of computing the exhaustive dialectical tree $\mathcal{T}_{\mathcal{A}}$ rooted in a given initial argument $\mathcal{A}$?* The answer is yes. In fact, as we will see in the next definitions, we can provide a lattice characterization for the space of all possible dialectical trees rooted in a given argument $\mathcal{A}$. In order to characterize a lattice for dialectical trees we will provide two operations:

- *Join* of dialectical trees ($\vee$), which given two dialectical trees $\mathcal{T}_1$ and $\mathcal{T}_2$ will compute the "union" of $\mathcal{T}_1$ and $\mathcal{T}_2$, in the sense that it will contain all defeaters present either in $\mathcal{T}_1$ *or* in $\mathcal{T}_2$.
- *Meet* of dialectical trees ($\wedge$), which given two dialectical trees $\mathcal{T}_1$ and $\mathcal{T}_2$ will compute the "intersection" of $\mathcal{T}_1$ and $\mathcal{T}_2$, in the sense that it will contain all defeaters present only in $\mathcal{T}_1$ *and* in $\mathcal{T}_2$.

Definition 19 Let T be an argumentative theory, and let $\mathcal{T}_1$ and $\mathcal{T}_2$ be dialectical trees rooted in $\mathcal{A}$. We define the *meet* and *join* of $\mathcal{T}_1$ and $\mathcal{T}_2$, (written $\mathcal{T}_1 \wedge \mathcal{T}_2$ and $\mathcal{T}_1 \vee \mathcal{T}_2$) as follows:

- λ is an argumentation line in $\mathcal{T}_1 \vee \mathcal{T}_2$ iff
  1. $\lambda \in \mathcal{T}_1$ and there is no $\lambda' \in \mathcal{T}_2$ such that λ' extends λ, or
  2. $\lambda \in \mathcal{T}_2$ and there is no $\lambda' \in \mathcal{T}_1$ such that λ' extends λ
- λ is an argumentation line in $\mathcal{T}_1 \wedge \mathcal{T}_2$ iff $\lambda = \lfloor\lambda_1\rfloor_k = \lfloor\lambda_2\rfloor_k$, for some $k > 0$ such that $\lambda_1 \in \mathcal{T}_1$ and $\lambda_2 \in \mathcal{T}_2$ and there is no λ' that extends λ satisfying this situation.

The next two results follow naturally from the previous definition.

Proposition 5 The operations $\wedge$ and $\vee$ are well-defined, i.e. for any dialectical trees $\mathcal{T}_1$ and $\mathcal{T}_2$ rooted in a given argument $\mathcal{A}$, $\mathcal{T}_1 \wedge \mathcal{T}_2$ and $\mathcal{T}_1 \vee \mathcal{T}_2$ are also dialectical trees rooted in $\mathcal{A}$.

Proposition 6 Let T be an argumentation theory, and λ an acceptable argumentation line in T. Then it holds that

1. $\lambda \in \mathcal{T}_1 \vee \mathcal{T}_2$ iff $\lambda \in \mathcal{T}_1$ or $\lambda \in \mathcal{T}_2$
2. $\lambda \in \mathcal{T}_1 \wedge \mathcal{T}_2$ iff $\lambda \in \mathcal{T}_1$ and $\lambda \in \mathcal{T}_2$
3. $\lambda \notin \mathcal{T}_1 \wedge \mathcal{T}_2$ iff $\lambda \notin \mathcal{T}_1$ or $\lambda \notin \mathcal{T}_2$

The next lemma shows that for any argumentation theory T the set of all possible acceptable dialectical trees rooted in a particular argument can be conceptualized as a lattice.

Lemma 1 Let $\mathcal{A}$ be an argument in a theory T, and let $(\mathfrak{ATree}_{\mathcal{A}}, \sqsubseteq_*)$ be the associated poset. Then $(\mathfrak{ATree}_{\mathcal{A}}, \vee, \wedge)$ is a lattice.

Given the lattice $(\mathfrak{ATree}_{\mathcal{A}}, \vee, \wedge)$, we will write $\mathcal{T}_{\mathcal{A}}^{\perp}$ to denote the bottom element of the lattice (i.e., the dialectical tree involving only $\mathcal{A}$ as root node) and $\mathcal{T}_{\mathcal{A}}^{\top}$ to denote the top element of the lattice (i.e., the exhaustive dialectical tree).

Example 8 Consider the theory T from Ex. 1, and the exhaustive dialectical tree rooted in $\mathcal{A}_0$ shown in Ex. 6. The complete lattice associated with $\mathcal{A}_0$ is shown in Fig. 3.

Computing Warrant Efficiently

In the preceding Section we have shown that given an argumentative theory T, for any argument $\mathcal{A}$ in T there is a lattice $(\mathfrak{ATree}_{\mathcal{A}}, \vee, \wedge)$ whose bottom element is a dialectical tree with a single node (the argument $\mathcal{A}$ itself) and whose top element is the exhaustive dialectical tree $\mathcal{T}_{\mathcal{A}}$. In that lattice, whenever $\mathcal{T}_k = \mathcal{T}_i \vee \mathcal{T}_j$ it is the case that $\mathcal{T}_i \sqsubseteq \mathcal{T}_k$ and $\mathcal{T}_j \sqsubseteq \mathcal{T}_k$.

In Fig. 3 corresponding to Example 8 we can see that for dialectical trees $\mathcal{T}_2$ and $\mathcal{T}_3$, it holds that *Mark*$(\mathcal{T}_2)$ = *Mark*$(\mathcal{T}_3)$ = D (assuming that *Mark* is defined as in Def. 16). Clearly, it is the case that any tree $\mathcal{T}_i$ where $\mathcal{T}_2 \sqsubseteq \mathcal{T}_i$ or $\mathcal{T}_3 \sqsubseteq \mathcal{T}_i$ satisfies that *Mark*$(\mathcal{T}_i) = D$. In other words, whichever is the way the tree $\mathcal{T}_2$ (or $\mathcal{T}_3$) evolves into a new tree in $(\mathfrak{ATree}_{\mathcal{A}_0}, \vee, \wedge)$ it turns out that the associated marking remains unchanged. We formalize that situation as follows:

Definition 20 Let T be an argumentation theory, and let $\mathcal{T}_{\mathcal{A}}$ be a dialectical tree, such that for every $\mathcal{T}'_{\mathcal{A}}$ evolving from $\mathcal{T}_{\mathcal{A}}$ (*i.e.*, $\mathcal{T}_{\mathcal{A}} \sqsubseteq_* \mathcal{T}'_{\mathcal{A}}$) it holds that *Mark*$(\mathcal{T}_{\mathcal{A}})$ = *Mark*$(\mathcal{T}'_{\mathcal{A}})$. Then $\mathcal{T}_{\mathcal{A}}$ is a *settled dialectical tree* in T.

Now we have a natural, alternative way of characterizing warrant.

Proposition 7 Let T be a theory, and let $\mathcal{A}$ be an argument in T. Then $\mathcal{A}$ is a warrant wrt T iff *Mark*$(\mathcal{T}_{\mathcal{A}}) = U$, where $\mathcal{T}_{\mathcal{A}}$ is a settled dialectical tree.

Clearly, computing settled dialectical trees is less expensive than computing exhaustive dialectical trees, as fewer nodes (arguments) are involved in the former case. Following Hunter's approach (Hunter 2004), in what follows we

multi-move protocols as defined by Prakken (Prakken 2005).

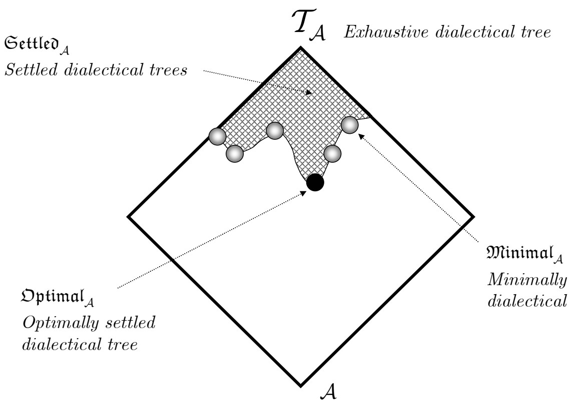

Figure 4: Search space for computing dialectical trees rooted in $\mathcal{A}$

will formalize the *cost* of computing a dialectical tree as a function $\mathsf{cost} : \mathfrak{Tree}_{T} \rightarrow \Re$. As explained in (Hunter 2004), several issues can be considered when computing such cost.

The next definition refines the class of settled dialectical trees by distinguishing those trees involving *as few arguments as possible* in order to determine whether the root of the tree is ultimately a warranted argument according to the marking procedure. From the many possible minimally settled dialectical trees rooted in a given argument $\mathcal{A}$, a dialectical tree $\mathcal{T}$ is *optimally settled* if there is no $\mathcal{T}'$ that is less expensive than $\mathcal{T}$.

Definition 21 A dialectical tree $\mathcal{T}$ is a *minimally settled dialectical tree* iff there is no $\mathcal{T}' \sqsubset \mathcal{T}$ such that $\mathcal{T}'$ is a settled dialectical tree. A dialectical tree $\mathcal{T}$ is an *optimally settled dialectical tree* iff $\mathcal{T}$ is minimally settled, and for any other settled tree $\mathcal{T}'$, $\mathsf{cost}(\mathcal{T}) \leq \mathsf{cost}(\mathcal{T}')$.

Example 9 Consider the theory T from Ex. 1, and the complete lattice $(\mathfrak{ATree}_{\mathcal{A}_0}, \vee, \wedge)$ shown in Fig. 3. Then $\mathcal{T}_2$ and $\mathcal{T}_3$ are minimally settled dialectical trees.

Let $\mathfrak{Settled}_{\mathcal{A}}$, $\mathfrak{Minimal}_{\mathcal{A}}$ and $\mathfrak{Optimal}_{\mathcal{A}}$ be the sets of all settled, minimally settled and optimally settled dialectical trees for an argument $\mathcal{A}$, resp. Clearly, it holds that

$$\mathfrak{Optimal}_{\mathcal{A}} \subseteq \mathfrak{Minimal}_{\mathcal{A}} \subseteq \mathfrak{Settled}_{\mathcal{A}} \subseteq \mathfrak{ATree}_{\mathcal{A}}.$$

The sets $\mathfrak{Settled}_{\mathcal{A}}$, $\mathfrak{Minimal}_{\mathcal{A}}$ and $\mathfrak{Optimal}_{\mathcal{A}}$ can be identified in any lattice $(\mathfrak{ATree}_{\mathcal{A}}, \vee, \wedge)$, as shown in Figure 4. The borderline on top of the lattice denotes all possible minimally settled dialectical trees $\mathcal{T}_1, \ldots, \mathcal{T}_k$ rooted in $\mathcal{A}$. Some of such trees in that set may be optimal. Any dialectical tree that evolves from settled dialectical trees $\mathcal{T}_1, \ldots, \mathcal{T}_k$ will be also a settled dialectical tree. In particular, the exhaustive dialectical tree is also settled.

Dialectical Constraints (Revisited)

As we have analyzed in the previous Section, the lattice associated with any argument $\mathcal{A}$ accounts for the whole search space for detecting if $\mathcal{A}$ is warranted. To do so it is not necessary to compute the exhaustive dialectical tree rooted in $\mathcal{A}$; rather, it suffices to focus search on settled dialectical trees, as they involve less nodes and are consequently more efficient.

When determining whether a conclusion is warranted, argument-based inference engines are supposed to compute a sequence of dialectical trees $\mathcal{T}_1, \mathcal{T}_2, \ldots, \mathcal{T}_k$ such that $\mathcal{T}_k$ is a settled dialectical tree. For skeptical argumentation semantics, argument-based engines like DeLP (García & Simari 2004; Chesñevar *et al.* 2003; Simari, Chesñevar, & García 1994a) use *depth-first search* to generate dialectical trees for queries and determine if a given literal is warranted. Such search can be improved by applying $\alpha - \beta$ pruning, so that not every node (argument) is computed. In other words, depth-first search favors naturally the computation of settled dialectical trees.

The natural question that arises next is how to compute *minimally settled trees*. Given a theory $T = (\Phi, \mathbf{DC})$, it turns out that the set of dialectical constraints $\mathbf{DC}$ can help to provide a way of approximating such minimally settled trees, based on the fact that in depth-first search the *order* in which branches are generated is important: should shorter branches be computed before longer ones, then the resulting search space can be proven to be smaller on an average search tree (Chesñevar, Simari, & Godo 2005). Usually heuristics are required to anticipate which branches are likely to be shorter than the average.

Constraints in $\mathbf{DC}$ can help provide such kind of heuristics. Thus, for example, in Defeasible Logic Programming (García & Simari 2004; Chesñevar *et al.* 2003) and Possibilistic Defeasible Logic Programming (Chesñevar *et al.* 2004) the set $\mathbf{DC}$ includes as a constraint that *arguments advanced by the proponent (resp. opponent) should not be contradictory* in any argumentation line. The following heuristics (Chesñevar, Simari, & Godo 2005) can be shown to favor the computation of shorter argumentation lines when applying depth-first search in the context of Possibilistic Defeasible Logic Programming: *if the current argument $\mathcal{A}_0$ is a leaf node in a dialectical tree $\mathcal{T}$, and has different candidate defeaters $\mathcal{A}_1, \mathcal{A}_2, \ldots, \mathcal{A}_k$, then the $\mathcal{A}_i$ which shares as many literals as possible with $\mathcal{A}_0$ should be chosen when performing the depth-first computation of $\mathcal{T}_{\mathcal{A}_0}$.*

By applying the above heuristics it can be shown that the branching factor for arguments below $\mathcal{A}_0$ is reduced. In other words, depth-first computation of dialectical trees favors naturally the construction of minimally settled dialectical trees, whereas by applying the above heuristics an approximation to optimally settled dialectical trees is obtained.

Relevance in Dialectical Trees

In (Prakken 2000) the notion of *relevance* was introduced in the context of argument games and the characterization of protocols for liberal disputes. According to (Prakken 2000), a move is relevant in a dispute D iff it changes the disputational status of D's initial move.[5] In our context, dialectical trees correspond to such disputes. In the setting presented in (Prakken 2000), moves are performed by both parties involved in a dispute (Proponent and Opponent).

Interestingly, there is a clear relation between minimally settled dialectical trees and this notion of relevance, as the notion of extending an argumentation line by one argument (as introduced in Def. 18) can be recast as performing a move.

Definition 22 Let $T = (\Phi, \mathbf{DC})$ be an argumentation theory, and let $\mathcal{T}_{\mathcal{A}_1}$, $\mathcal{T}'_{\mathcal{A}_1}$ be acceptable dialectical trees. We will say that there is a *move* M from $\mathcal{T}_{\mathcal{A}}$ to $\mathcal{T}'_{\mathcal{A}}$, denoted as *Move*$(\mathcal{T}_{\mathcal{A}}, \mathcal{T}'_{\mathcal{A}})$, iff $\mathcal{T}_{\mathcal{A}} \sqsubset \mathcal{T}'_{\mathcal{A}}$.

It must be remarked that a proper conceptualization of move in argumentation demands more parameters, such as identifying the argumentation line in which a argument is introduced, who is the player (Proponent or Opponent) making the move, etc. Such an approach has been formalized by (Prakken 2000; 2005). Our approach in this case is intentionally over-simplified, as it just aims to relate the notion of relevance and the notion of minimally settled dialectical trees. In fact, note that Def. 22 allows us to formalize the computation of an acceptable dialectical tree $\mathcal{T}_k$ rooted in $\mathcal{A}_0$ as a sequence of moves *Move*$(\mathcal{T}_0, \mathcal{T}_1)$, *Move*$(\mathcal{T}_1, \mathcal{T}_2)$, $\ldots$, *Move*$(\mathcal{T}_{k-1}, \mathcal{T}_k)$, where $\mathcal{T}_0$ is a dialectical tree with a single node $\mathcal{T}^{\perp}_{\mathcal{A}_0}$. Following Prakken's notion of relevance, we can express this concept in our setting as follows:

Definition 23 A move M = *Move*$(\mathcal{T}_{\mathcal{A}}, \mathcal{T}'_{\mathcal{A}})$ is *relevant* iff *Mark*$(\mathcal{T}_{\mathcal{A}}) \neq$ *Mark*$(\mathcal{T}'_{\mathcal{A}})$.

The following proposition shows that minimally settled trees are only those obtained by performing a sequence of relevant moves ending in a settled dialectical tree.

Proposition 8 Let T be an argumentation theory, and let $\mathcal{T}_{\mathcal{A}}$ be a dialectical tree. Then $\mathcal{T}_{\mathcal{A}}$ is minimally settled iff there is a sequence of moves $M_1, M_2, \ldots, M_k$ such that every move M_i is relevant, and M_k results in a settled dialectical tree.

Related Work

Dialectical constraints have motivated research in argumentation theory in different directions. As stated before, the main role of such constraints is to avoid *fallacious* reasoning (Aristotle ; Hamblin 1970; Rescher 1977; Walton 1995). In our proposal dialectical constraint are left as a particular parameter to be included in the argumentation theory. It must be remarked that different formalizations of argument-based dialectical proof procedures have included particular dialectical constraints as part of their specification. In (Simari, Chesñevar, & García 1994a; Simari, Chesñevar, & García 1994b), an approach to model different dialectical constraints was presented. These constraints were applied as part of the procedure used for constructing dialectical trees by discarding "ill-formed" argumentation lines. In (Besnard & Hunter 2001) the authors present a logic of argumentation which disallows repetition of arguments in argument trees (Besnard & Hunter 2001, p.215):

> *For no node* (ϕ, β) *with ancestor nodes* (ϕ_1, β_1), $(\phi_2, \beta_2), \ldots, (\phi_k, \beta_k)$ *is* ϕ *a subset of* $\phi_1 \cup \phi_2 \cup \ldots \cup \phi_k$.

In a similar manner, other approaches (like (Kakas & Toni 1999)) compute different semantics for logic programming on the basis of an argumentative approach formalized in terms of trees. Some properties can be used to render the construction of such trees more efficient. Thus, in the case of computing well-founded semantics via trees, defense nodes (which account for Proponent's argument in an argumentation line) cannot attack any other defense node in the tree. Similarly, in (Dung, Kowalski, & Toni 2006) the notion of dispute tree is used to compute assumption-based, admissible argumentation. As the authors indicate, in order for an abstract dispute tree to be *admissible*, there is a further requirement that "the proponent does not attack itself". Such kind of restrictions can be seen as particular dialectical constraint in the context of our proposal.

Recently there have been other research oriented towards formalizing dialectical proof procedures for argumentation. To the best of our knowledge, none of such works formalizes the dialectical search space through a lattice as presented in this paper. Our work complements previous research concerning the dynamics of argumentation, notably (Prakken 2001) and (Brewka 2001). In particular, Prakken (Prakken 2001) has analyzed the exchange of arguments in the context of dynamic disputes. Our approach can also be understood in the light of his characterization of dialectical proof theories (Prakken 2005). However, although Prakken develops a very comprehensive general framework, in our understanding some important computational issues (e.g. search space considerations) are not taken into account. Hunter (Hunter 2004) analyzes the search space associated with dialectical trees taking into account novel features such as the *resonance* of arguments. His interesting formalization combines a number of features that allow to assess the impact of dialectical trees, contrasting shallow vs. deep trees. However, search space considerations as modeled in this paper are outside the scope of his approach. In (Kakas & Toni 1999) a throughout analysis of various argumentation semantics for logic programming is presented on the basis of parametric variations of derivation trees. In contrast with that approach, our aim in this paper was not to characterize different emerging semantics, but rather to focus on the role of dialectical trees as a way of modeling the search space when computing warrants. Besides, in (Kakas & Toni 1999) the authors concentrate in normal logic programming, whereas our approach is more generic.

Conclusions. Future Work

In this paper we have presented a novel approach to model the search space associated with warrant computation in an

[5] The notion of relevance as well as some interesting properties were further studied and refined (Prakken 2005).

abstract argumentation framework. We have shown how the notion of dialectical tree can be used constructively to model different stages in the process of computing warranted arguments. We have shown how the process of computing warrant can be recast into computing dialectical trees within a lattice, illustrating how dialectical constraints can play a role for guiding an efficient computation of warranted literals.

Part of our future work is related to studying theoretical properties of the proposed framework, analyzing their incidence for developing efficient argument-based inference engines. In this context we think that the notion of equivalence classes associated with dialectical trees can be specially useful as discussed in Remark 2. Research in this direction is currently being pursued.

Acknowledgements We thank anonymous reviewers for comments which helped to improve the final version of this paper. This research was partially supported by Projects TIC2003-00950, TIN2004-07933-C03-01/03, by Ramón y Cajal Program (Ministerio de Ciencia y Tecnología, Spain), by CONICET (Argentina), by the Secretaría General de Ciencia y Tecnología de la Universidad Nacional del Sur and by Agencia Nacional de Promoción Científica y Tecnológica (PICT 2002 No. 13096).

References

Aristotle. *On Sophistical Refutations*. eBooks@Adelaide, Translated by W. A. Pickard-Cambridge, web edition at `http://etext.library.adelaide.edu.au/a/aristotle/sophistical/`, The University of Adelaide Library, 2004 edition.

Besnard, P., and Hunter, A. 2001. A logic-based theory of deductive arguments. *Artificial Intelligence* 1:2(128):203–235.

Brewka, G. 2001. Dynamic argument systems: A formal model of argumentation processes based on situation calculus. *J. of Logic and Computation* 11(2):257–282.

Chesñevar, C., and Simari, G. 2005. Computing warrant in an abstract model for skeptical argumentation frameworks: formalization and properties. In *Technical Report 343 – University of LLeida & LIDIA*.

Chesñevar, C.; Dix, J.; Stolzenburg, F.; and Simari, G. 2003. Relating Defeasible and Normal Logic Programming through Transformation Properties. *Theoretical Computer Science* 290(1):499–529.

Chesñevar, C. I.; Simari, G.; Alsinet, T.; and Godo, L. 2004. A Logic Programming Framework for Possibilistic Argumentation with Vague Knowledge. In *Proc. of the Intl. Conf. in Uncertainty in Art. Intelligence. (UAI 2004). Banff, Canada*, 76–84.

Chesñevar, C.; Maguitman, A.; and Loui, R. 2000. Logical Models of Argument. *ACM Computing Surveys* 32(4):337–383.

Chesñevar, C.; Simari, G.; and Godo, L. 2005. Computing dialectical trees efficiently in possibilistic defeasible logic programming. *LNAI/LNCS Springer Series (Proc. of the 8th Intl. Conference on Logic Programming and Nonmonotonic Reasoning LPNMR 2005)* 158–171.

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

Dung, P. 1993. On the Acceptability of Arguments and its Fundamental Role in Nomonotonic Reasoning and Logic Programming. In *Proceedings of the 13th. International Joint Conference in Artificial Intelligence (IJCAI), Chambáery, Francia*, 852–857.

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

García, A., and Simari, G. 2004. Defeasible Logic Programming: An Argumentative Approach. *Theory and Practice of Logic Programming* 4(1):95–138.

Hamblin, C. L. 1970. *Fallacies*. Methuen, London.

Hunter, A. 2004. Towards Higher Impact Argumentation. In *Proc. of the 19th American National Conf. on Artificial Intelligence (AAAI'2004)*, 275–280. MIT Press.

Jakobovits, H., and Vermeir, D. 1999. Dialectic semantics for argumentation frameworks. In *ICAIL*, 53–62.

Jakobovits, H. 1999. Robust semantics for argumentation frameworks. *Journal of Logic and Computation* 9(2):215–261.

Kakas, A., and Toni, F. 1999. Computing argumentation in logic programming. *Journal of Logic Programming* 9(4):515:562.

Prakken, H., and Sartor, G. 1997. Argument-based extended logic programming with defeasible priorities. *Journal of Applied Non-classical Logics* 7:25–75.

Prakken, H., and Vreeswijk, G. 2002. Logical Systems for Defeasible Argumentation. In Gabbay, D., and F.Guenther., eds., *Handbook of Philosophical Logic*. Kluwer Academic Publishers. 219–318.

Prakken, H. 2000. Relating protocols for dynamic dispute with logics for defeasible argumentation. *Synthese (to appear)*.

Prakken, H. 2001. Relating protocols for dynamic dispute with logics for defeasible argumentation. *Synthese (special issue on New Perspectives in Dialogical Logic)* 127(4):187:219.

Prakken, H. 2005. Coherence and flexibility in dialogue games for argumentation. *Journal of Logic and Computation* 15:1009–1040.

Rescher, N. 1977. *Dialectics, a Controversy-Oriented Approach to the Theory of Knowledge*. Albany, USA: State University of New York Press.

Simari, G.; Chesñevar, C.; and García, A. 1994a. The Role of Dialectics in Defeasible Argumentation. In *Anales de la XIV Conferencia Internacional de la Sociedad Chilena para Ciencias de la Computación*, 260–281. Universidad de Concepción, Concepción (Chile).

Simari, G. R.; Chesñevar, C. I.; and García, A. J. 1994b. Focusing Inference in Defeasible Argumentation. In *Proc. of IV Iberoamerican Congress on Artificial Intelligence,IBERAMIA'94*.

Vreeswijk, G. 1997. Abstract Argumentation Systems. *Artificial Intelligence* 90:225–279.

Walton, D. 1995. *A Pragmatic Theory of Fallacy (Studies in Rhetoric and Communication Series)*. The University of Alabama Press, Tuscaloosa.