A Lattice-based Approach to Computing Warranted Beliefs in Skeptical Argumentation Frameworks

Carlos Chesñevar

Dept. of Computer Science – University of Lleida – C/Jaume II,69 – 25001 Lleida, Spain

Email: cic@eps.udl.es

Guillermo Simari

Dept. of Computer Science and Eng. – Universidad Nacional del Sur – 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. In order to make such semantics computationally attractive, suitable proof procedures are required, in which a search space of arguments is examined to find out which arguments are warranted or ultimately acceptable. This paper introduces a novel approach to model warrant computation 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.

1 Introduction and Motivations

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 the above 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. Section 2 presents the basic ideas of an abstract argumentation framework with dialectical constraints. Section 3 shows how so-called dialectical trees can be used to analyze the search space for computing warrants, representing it as a lattice. In Section 4 we analyze different criteria which can lead to compute warrant more efficiently on the basis of this lattice characterization. Finally, in Sections 5 and 6 we discuss some related work and present the main conclusions that have been obtained.

2 An Abstract Argumentation Framework with Dialectical Constraints

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]) enriched with the notion of *dialectical constraint*.

Definition 1 [Dung, 1995] 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 $\mathbf{R} \subseteq \mathfrak{Args} \times \mathfrak{Args}$. The notation $(\mathcal{A}, \mathcal{B}) \in \mathbf{R}$ (or equivalently $\mathcal{A}\,\mathbf{R}\,\mathcal{B}$) means that $\mathcal{A}$ *attacks* $\mathcal{B}$.

A dialectical constraint imposes a restriction characterizing when a given argument sequence λ is valid in a framework Φ. An argumentation theory is defined by combining an argumentation framework with a particular set of 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 $\mathbf{C} : \mathfrak{Lines}_\Phi \rightarrow \{True, False\}$, where $\mathfrak{Lines}_\Phi$ denotes 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 $\mathcal{A}_i\ \mathbf{R}\ \mathcal{A}_{i+1}$.

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 [Hamblin, 1970; Rescher, 1977] by imposing appropriate constraints on argumentation lines to be considered rationally *acceptable*. 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.

Argument games provide a useful form to characterize proof procedures for skeptical semantics in argumentation.[1] Such games model defeasible reasoning as a dispute between two parties (*Proponent* and *Opponent* of a claim), who

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

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. Dialogues in such argument games have been given different names (dialogue lines, argumentation lines, dispute lines, etc.). The set of all possible dialogues can also be suitably defined as a tree structure (called dialectical tree or argument tree).[2] In the next subsection we extend such definitions in the context of an argumentation theory.

2.1 Argumentation Line. Bundle set

Definition 4 Let $T = (\Phi, \mathbf{DC})$ be an argumentation theory. An *argumentation line* λ in T is any *finite* sequence of arguments $[\mathcal{A}_0, \mathcal{A}_1, \ldots, \mathcal{A}_n]$ in $\mathfrak{Lines}_\Phi$. A subsequence $\lambda' = [\mathcal{A}_0, \mathcal{A}_1, \mathcal{A}_2, \ldots, \mathcal{A}_k]$, $k \leq n$, will be called an *initial argumentation segment* (or just *initial segment*) in λ of length k, denoted $\lfloor\lambda\rfloor_k$. When $k < n$ we say that λ' is a *proper initial segment* in λ.

We will say that λ is *rooted in* $\mathcal{A}_0$, 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.

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 $\lambda_1 = [\mathcal{A}_0, \mathcal{A}_1, \mathcal{A}_4]$, $\lambda_2 = [\mathcal{A}_0, \mathcal{A}_1, \mathcal{A}_2]$, and $\lambda_3 = [\mathcal{A}_0, \mathcal{A}_3]$. 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 5 An argumentation line λ is *acceptable* wrt $T = (\Phi, \mathbf{DC})$ iff $\mathbf{C}_i(\lambda) = True$, $\forall \mathbf{C}_i \in \mathbf{DC}$.

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.

Example 3 Consider the theory T' in Ex. 2, and assume that $\mathbf{DC}$=$\{\mathbf{C}_1\}$, with $\mathbf{C}_1$ = {repetition of arguments is not allowed }.[3] Then λ_1 and λ_2 are acceptable argumentation lines in T', but λ_3 and λ_4 are not.

[2]For in-depth discussion see [Prakken and Vreeswijk, 2002].

[3]Note that this corresponds to a function $\mathbf{C}_1(\lambda) = True$ iff $\nexists \mathcal{A}_i, \mathcal{A}_j$ in λ such that $\mathcal{A}_i = \mathcal{A}_j$, and $False$ otherwise.

Definition 6 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 $k < \mid \lambda' \mid$ ($i.e.$ λ' extends λ iff λ is a proper initial argumentation segment of λ').
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, $\lambda = \lfloor\lambda'\rfloor_k$ ($i.e.$ $\nexists\lambda'$ such that extends λ in T). 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 all 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 7 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 a *bundle set* wrt T iff $\nexists \lambda_i, \lambda_j \in S_\mathcal{A}$ 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.

2.2 Dialectical Trees

A bundle set $S_\mathcal{A}$ is a set of argumentation lines rooted in a given argument $\mathcal{A}$. Such set can be thought of as a tree structure, where every line corresponds to a branch in the tree. Formally:

Definition 8 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 $\lambda \in F\}$.[4] 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 *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 *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 $\mathcal{T}_{\mathcal{A}_0}$.

Clearly, Definition 8 induces an equivalence relation on the set of all $\mathfrak{Tree}_\mathcal{A}$. Formally:

Definition 9 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 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 same bundle set $S_\mathcal{A}$.

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

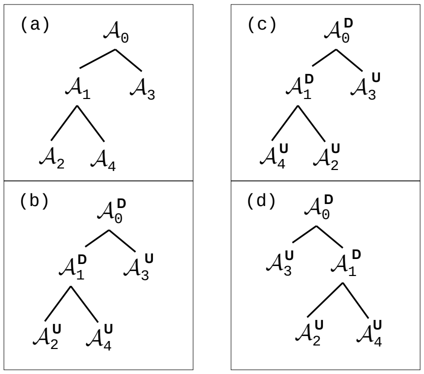

Figure 1: (a)Exhaustive dialectical tree $\mathcal{T}_{\mathcal{A}_0}$ for Ex. 6; (b)resulting tree after applying and-or marking (Def.14);(c)–(d) two other exhaustive dialectical trees belonging to the equivalence class $\overline{\mathcal{T}_{\mathcal{A}_0}}$

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. 8). 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 dialectical tree from the bundle set will produce one particular member on the equivalence class.

Definition 10 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.[5]

As the mapping $\mathbb{T}$ is a bijection, we can also define the inverse mapping $\mathbb{S} =_{def} \mathbb{T}^{-1}$. 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 notations to the other.

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.

2.3 Acceptable dialectical trees

Definition 11 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.

[5]Proofs not included for space reasons.

Definition 12 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*.

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 given in Def. 9)

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

Definition 13 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, which corresponds to Dung's grounded semantics [Dung, 1995].

Definition 14 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 $\exists\ \mathcal{T}_i$ 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. 14 assigns the same mark to all the members of $\overline{\mathcal{T}_\mathcal{A}}$.

Definition 15 Let T be an argumentative theory and *Mark* a marking criterion for T. An argument $\mathcal{A}$ is a *warranted argument* (or just a *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. 14, 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. Fig. 1(c)–(d) shows two marked dialectical trees belonging to the same equivalence class $\overline{\mathcal{T}_{\mathcal{A}_0}}$.

3 Warrant Computation via Dialectical Trees

Our main concern is to model warrant computation in skeptical argumentation frameworks, and in such a case tree structures lend themselves naturally to implementation. In fact, some implementations of skeptical argumentation systems (e.g. DeLP [García and 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 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 *et al.*, 2006] using a similar strategy to compute warranted belief.

In our abstract model 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 via a precedence relationship "$\sqsubseteq$" among trees:

Definition 16 Let T be a theory, and let $\mathcal{T}_\mathcal{A}$, $\mathcal{T}'_\mathcal{A}$ be acceptable dialectical trees rooted in an argument $\mathcal{A}$. 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 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 $\mathcal{T}_1, \mathcal{T}_2, \ldots, \mathcal{T}_k$ s.t. $\mathcal{T}_\mathcal{A} = \mathcal{T}_1 \sqsubseteq \ldots \sqsubseteq \mathcal{T}_k = \mathcal{T}'_\mathcal{A}$.

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 all possible evolutions of a given disputation.[6] In particular, note that for any argumentative theory 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 $\mathcal{T}_\mathcal{A}$. From Def. 16 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 $\mathcal{T}_i \sqsubset \mathcal{T}_j$.

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 $\mathcal{A}$ on the basis of two operations: *join* of dialectical trees ($\vee$) (resulting in a new tree corresponding to the 'union" of $\mathcal{T}_1$ and $\mathcal{T}_2$) and meet of dialectical trees ($\wedge$) (resulting in a new tree corresponding to the "intersection" of $\mathcal{T}_1$ and $\mathcal{T}_2$). Formally:

Definition 17 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.

For any argumentation theory T the set of all possible acceptable dialectical trees rooted in an argument $\mathcal{A} \in T$ can be conceptualized as a lattice. Formally:

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 $\mathcal{T}^\perp_\mathcal{A}$ to denote the bottom element of the lattice (i.e., the dialectical tree involving only $\mathcal{A}$ as root node) and $\mathcal{T}^\top_\mathcal{A}$ to denote the top element of the lattice (i.e., the exhaustive dialectical tree).

[6] 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 *multi-move protocols* as defined by Prakken [Prakken, 2005].

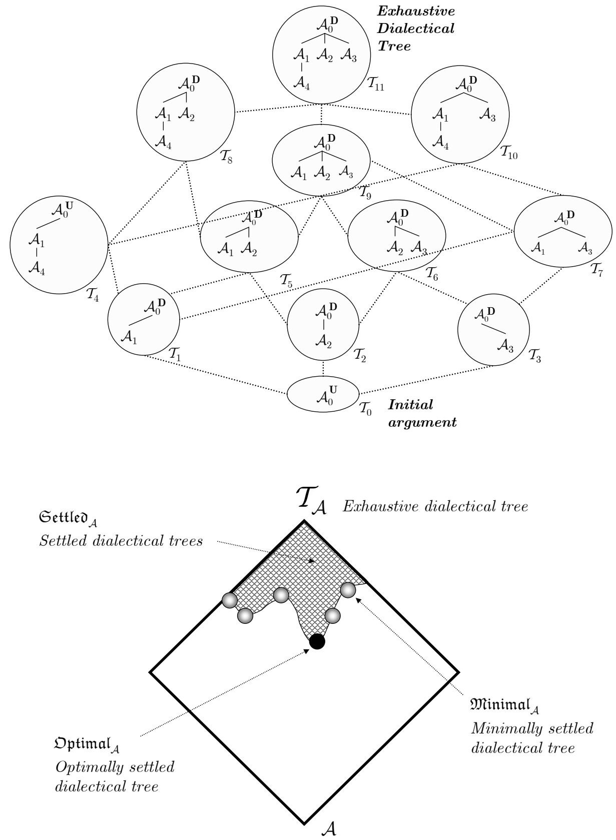

Figure 2: Lattice for all possible dialectical trees rooted in an argument $\mathcal{A}_0$ (Example 8) (top) and search space for computing dialectical trees rooted in $\mathcal{A}$ (bottom)

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. 2.

4 Computing Warrant Efficiently

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. 2(top) 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. 14). 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 18 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 5 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 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. For simplicity, in our formalization we will assume that cost is linearly related to the number of nodes in a dialectical tree, such that $\mathsf{cost}(\mathcal{T}) = C * Nodes(\mathcal{T})$, where $Nodes(\cdot)$ stands for the number of nodes in a tree.

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 $\nexists\, \mathcal{T}'$ that is less expensive than $\mathcal{T}$.

Definition 19 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. 2 (top). 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 Fig. 2 (bottom). 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.

4.1 Dialectical Constraints (Revisited)

As we have analyzed previously, 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, inference engines like DeLP [García and Simari, 2004] 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 naturally favors the computation of settled dialectical trees.

Example 10 Consider the marked dialectical trees in Fig. 1(right) belonging to the same equivalence class $\overline{\mathcal{T}_{\mathcal{A}_0}}$ (Ex. 7). Then depth-first computation using $\alpha - \beta$ pruning will perform better on the tree in Fig. 1(d) than on the tree in Fig. 1(c), as in the first case, only two nodes need to be explored to obtain the final marking of the tree ($\mathcal{A}_0$ and $\mathcal{A}_3$), whereas in the second case four nodes ($\mathcal{A}_0$, $\mathcal{A}_1$, $\mathcal{A}_3$ and $\mathcal{A}_4$) need to be traversed.

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 *et al.*, 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. In this setting, heuristics for efficient computation of dialectical trees can be understood as functions which improve the associated dialectical proof procedure *by tending to approximate optimally settled trees.*

Example 11 In DeLP 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 *et al.*, 2005] can be shown to favor the computation of shorter argumentation lines when applying depth-first search in the context of DeLP: *if the current argument $\mathcal{A}_0$ is a leaf node in a dialectical tree $\mathcal{T}$, and has different candidate defeaters $\mathcal{A}_1$, $\mathcal{A}_2$, …, $\mathcal{A}_k$, 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 $\mathcal{T}_{\mathcal{A}_0}$*. Thus, while depth-first computation of dialectical trees favors naturally the construction of minimally settled dialectical trees, by applying this heuristics an approximation to optimally settled dialectical trees is obtained.

4.2 Relevance in Dialectical Trees

In [Prakken, 2001] the notion of *relevance* was introduced in the context of argument games and the characterization of protocols for liberal disputes. According to [Prakken, 2001], a move is relevant in a dispute D iff it changes the disputational status of D's initial move.[7] In our context, dialectical trees correspond to such disputes. In the setting presented in [Prakken, 2001], 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. 16) can be recast as performing a move.

Definition 20 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 $\mathcal{T}_\mathcal{A} \sqsubset \mathcal{T}'_\mathcal{A}$.

[7] The notion of relevance as well as some interesting properties were further studied and refined [Prakken, 2005].

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, 2001; 2005]. Our approach in this case is intentionally oversimplified, as it just aims to relate the notion of relevance and the notion of minimally settled dialectical trees. In fact, note that Def. 20 allows us to formalize the computation of an acceptable dialectical tree $\mathcal{T}_k$ rooted in $\mathcal{A}_0$ as a sequence of moves $\mathit{Move}(\mathcal{T}_0, \mathcal{T}_1)$, $\mathit{Move}(\mathcal{T}_1, \mathcal{T}_2)$, …, $\mathit{Move}(\mathcal{T}_{k-1}, \mathcal{T}_k)$, where $\mathcal{T}_0$ is a dialectical tree with a single node $\mathcal{T}^{\perp}_{\mathcal{A}_0}$. In fact, Prakken's notion of *relevant move* can be stated in our setting as follows: a move M = $\mathit{Move}(\mathcal{T}_{\mathcal{A}}, \mathcal{T}'_{\mathcal{A}})$ is *relevant* iff $\mathit{Mark}(\mathcal{T}_{\mathcal{A}}) \neq \mathit{Mark}(\mathcal{T}'_{\mathcal{A}})$.

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 6 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, …, M_k such that every move M_i is relevant, and M_k results in a settled dialectical tree.

5 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. In our proposal dialectical constraints are left as a particular parameter to be included in the argumentation theory. Different argument-based proof procedures have included particular dialectical constraints as part of their specification. In [Besnard and Hunter, 2001] the authors present a logic of argumentation which disallows repetition of arguments in argument trees [Besnard and Hunter, 2001, p.215] Other approaches for computing well-founded semantics via trees (e.g. [Kakas and Toni, 1999]) defense nodes (which account for Proponent's argument in an argumentation line) cannot attack any other defense node in the tree. Similarly, in [Dung *et al.*, 2006], for computing assumption-based admissible semantics there is a further requirement in the proof procedure that "the proponent does not attack itself". Such kind of restrictions can be seen as particular dialectical constraints 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]. Although Prakken develops a very comprehensive general framework, some important computational issues (e.g. search space considerations) are not taken into account.

6 Conclusions. Future Work

In this paper we have presented a novel approach to model the search space associated with warrant computation in an 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 also 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. Research in this direction is currently being pursued.

Acknowledgements

This research was partially supported by Ramón y Cajal Program (MCyT, Spain), by CONICET (Argentina), by the Secretaría General de Ciencia y Tecnología (UNS, Argentina) and by Agencia Nacional de Promoción Científica y Tecnológica (PICT 2002 No. 13096).

References

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

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

[Chesñevar *et al.*, 2005] C. Chesñevar, G. Simari, and L. Godo. Computing dialectical trees efficiently in possibilistic defeasible logic programming. *LNAI/LNCS Series (Proc. 8th Intl. LPNMR Conf.)*, pages 158–171, 2005.

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

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

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

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

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

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

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

[Prakken, 2001] H. Prakken. Relating protocols for dynamic dispute with logics for defeasible argumentation. *Synthese*, 127(4):187:219, 2001.

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

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