

A Dialectical Proof Theory for Universal Acceptance in Coherent Logic-based Argumentation Frameworks

Abdallah Arioua and Madalina Croitoru¹

Abstract. Given a logic-based argumentation framework built over a knowledge base in a logical language and a query in that language, the query is *universally accepted* if it is entailed from all extensions. As shown in [2, 14], universal acceptance is different from skeptical acceptance as a query may be entailed from different arguments distributed over all extensions but not necessarily skeptical ones. In this paper we provide a dialectical proof theory for universal acceptance in coherent logic-based argumentation frameworks. We prove its finiteness, soundness, completeness, consistency and study its dispute complexity. We give an exact characterization for non-universal acceptance and provide an upper-bound for universal acceptance.

1 Introduction

Dialectical proof theories have their roots in the dialogical approach to logic traditions [23]. In the Greek antiquity logic was studied in a dialogical context where two parties exchange arguments over a central claim. In modern logic the dialogical approach (or *dialogical logics*) is used to provide a game-theoretical semantics for logical systems. Proofs according to dialogical logics is a dialogue game between two parties arguing about a thesis while respecting some fixed rules. The dialogue is adversarial where one party plays the role of the defender of the thesis (proponent) and the other argues against the thesis (opponent). Each dialogue ends after a finite number of moves with a winner and a loser.

Since the work of Dung [16] many attempts have been made to adapt the dialogical approach to provide formal proof theories for formal argumentation, this is often referred to as *dialectical proof theories*. The works of [20, 28] define, similarly to dialogical logic, a dialectical proof theory as an argument game with a winning criterion alongside with a legal move function that decides the allowed moves to be played. Given an argumentation framework, a semantics x and an argument a , the objective is to prove whether the argument a is skeptically/credulously accepted under a semantics x .

The TPI (Two Party Immediate response) procedure proposed in [30] and further formalized in [17] is used for credulous and skeptical games in finite and coherent argumentation frameworks where two players exchange arguments (moves) until one of them cannot play. The justification status of the argument (skeptical/credulous) is decided with respect to the winning criterion. The turn in TPI-disputes shifts after one move with the move m_i attacks the precedent one (hence *immediate response*). Their dialectical proof theories are sound and complete. In [12], the same guideline is followed but with a refinement on the size of the proof, where [12] produces

shorter proofs than [17]. In [25] a different dialectical proof theory has been proposed for skeptical acceptance where, instead of exchanging arguments the proponent and the opponent exchange whole admissible sets. The goal is to construct a *block*, which is an admissible set of arguments that conflicts with all admissible sets around the argument in question [25, Theorem 6.7]. Following the same idea, [15] constructs such block in a *meta-argumentation framework* within a *meta-dialogue* where admissible sets are considered as moves, then the classical credulous proof theory of [12] is used as a sub-procedure to prove skeptical acceptance. In [29] a more general framework has been provided which is sound for any argumentation frameworks and it is complete for general classes of finitary argumentation frameworks including the class of finite argumentation frameworks using the notions of dispute derivation and base derivation. For skeptical preferred, the proof theory proposes to find a base then check whether it is complete or not. A base of an argument a is a set of admissible sets that (each of which) contains a such that whenever a is in an extension then there is an admissible set in the base that belongs to this extension. The base is complete if all extensions contains an admissible set from the base.

When it comes to logic-based argumentation the situation is quite different. In *logic-based argumentation* we differentiate between the *acceptance of an argument* and the *acceptance of a query*. A query is *universally accepted* under a semantics x if it is entailed from **every** extension. A query is *skeptically accepted* under a semantics x if it is entailed from a skeptically accepted argument. It is important to notice that the universal acceptance of a query does not necessarily mean that the query is skeptically accepted whereas the skeptical acceptance of a query necessarily yields the universal acceptance of the query. Skeptical acceptance can be easily handled by state of the art dialectical proof theories. However, already proposed dialectical proof theories fail when it comes to the universal acceptance as this one is not implied by skeptical acceptance.

In this paper, following [17], we propose a new TPI-like dialectical proof theory for universal acceptance. We limit the scope of the work to finite and coherent logic-based argumentation frameworks. In coherent argumentation frameworks the stable and preferred extensions coincide. Therefore, our dialectical proof theory works for all the above mentioned semantics. We show the soundness, completeness and finiteness of the proposed proof theory and analyse its dispute complexity properties.

2 The Dialectical Proof Theory

2.1 Preliminaries and Motivating Example

To facilitate the readability of this section, we first introduce necessary background notions then we shift directly to Example 1 (moti-

¹ GraphIK, INRIA: INRA, LIRMM and University Montpellier, France. Email: abdallaharioua@gmail.com

vating example) that shows how existing work cannot be applied to universal acceptance. Then we give a complete characterization of such acceptance.

We consider existential rules, widely used nowadays on the Semantic Web [27, 6, 26], they generalize certain Description Logics (such as \mathcal{EL} [5] and DL-Lite [11] families) with cyclicity notions and predicate arity. This language is a fragment of first-order logic (aka *Datalog* \pm [10]) composed of formulas built with the only logical connectives (\wedge, \rightarrow), the quantifiers (\exists, \forall) and the special constant \perp . An *atom* is of the form $p(t_1, \dots, t_k)$ where p is a predicate of arity k and the t_i are terms (variables or constants). A finite set of atoms A is called an *atomset*, we denote by $terms(A)$ (resp. $vars(A)$) the set of terms (resp. variables) that occur in A . Given atomsets A_1 and A_2 , a *homomorphism* π from A_1 to A_2 is a substitution of $vars(A_1)$ by $terms(A_2)$ such that $\pi(A_1) \subseteq A_2$. In this case we say $A_2 \models A_1$ where \models is the FOL entailment. An *existential rule* is of the form $R = \forall \vec{x} \forall \vec{y} (B \rightarrow \exists \vec{z} H)$, where B and H are conjunctions of atoms², with $vars(B) = \vec{x} \cup \vec{y}$, and $vars(H) = \vec{x} \cup \vec{z}$. B and H are respectively called the *body* and the *head* of R . A rule with an empty body (resp. head set to \perp) is called a *fact* (resp. *negative constraint*). A *boolean conjunctive query* (BCQ) Q has the form of a fact. From now on we use the general term *query* to mean BCQ. We denote variables by uppercase letters X, Y, Z, \dots , constants by lowercase letters a, b, c, \dots and predicate symbols by lowercase letters p, q, r, s , etc.

A *knowledge base* $\mathcal{K} = (\mathcal{F}, \mathcal{R}, \mathcal{N})$ is composed of a finite sets of facts \mathcal{F} , rules \mathcal{R} and negative constraints \mathcal{N} . Facts represent factual knowledge about the world, rules represent generic rule-based knowledge, and negative constraints represent logical falsehood (e.g. $\forall X (cat(X) \wedge dog(X) \rightarrow \perp)$). We say a rule $R \in \mathcal{R}$ is applicable on a fact $F \in \mathcal{F}$ iff there is a homomorphism from F to the body of R . This application gives a new fact F' which is the head of R with instantiated variables. For instance $\forall X (p(X) \rightarrow q(X))$ is applicable on $p(a)$ and it gives $q(a)$. The application of all rules \mathcal{R} on all facts \mathcal{F} exhaustively until termination is referred to as *the chase* and it produces the set of facts $\mathcal{C}_{1\mathcal{R}}(\mathcal{F})$ (all deducible facts). We restrict our work to the *finite expansion* fragment that guarantees that the chase halts and $\mathcal{C}_{1\mathcal{R}}(\mathcal{F})$ is finite [6]. We say a query is entailed from \mathcal{K} iff $\mathcal{C}_{1\mathcal{R}}(\mathcal{F}) \models Q$. Since the chase always halts then query entailment is **decidable**. For a given \mathcal{K} , we say that a set of facts $F \subseteq \mathcal{F}$ is inconsistent (consequently \mathcal{K}) iff $\mathcal{C}_{1\mathcal{R}}(\mathcal{F}) \models \perp$.

The definition of an argument in this language is similar to the usual definition in logic-based argumentation of [7, 1].

Definition 1 (Argument). *Given a knowledge base $\mathcal{K} = (\mathcal{F}, \mathcal{R}, \mathcal{N})$. An argument is a tuple (H, C) such that: (1) $H \subseteq \mathcal{F}$ with $\mathcal{C}_{1\mathcal{R}}(H) \not\models \perp$ (consistency), and (2) $C \in \mathcal{C}_{1\mathcal{R}}(H)$ and $H \models C$ (entailment), and (3) there is no $H' \subset H$ that verifies (1) and (2) (minimality). The support (resp. conclusion) of an argument a are denoted by $Supp(a) = H$ (resp. $Conc(a) = C$).*

We denote arguments by subscripted lowercase letters a_i, b_i , etc. However, we may use a, b, \dots when there is no ambiguity.

Arguments may attack each other with different types of attacks identified in the literature [8]. Here we focus on the *assumption attack* of [18] as it satisfies the rationality postulates [14].

Definition 2 (Attack). *An argument a attacks b iff $\exists h \in Supp(b)$ such that $\mathcal{C}_{1\mathcal{R}}(\{Conc(a), h\}) \models \perp$.*

Definition 3 (Argumentation framework). *Let $\mathcal{K} = (\mathcal{F}, \mathcal{R}, \mathcal{N})$ be a knowledge base. The corresponding argumentation framework is a*

pair $\mathcal{H} = (\mathbf{Arg}(\mathcal{F}), \mathcal{U})$ where $\mathbf{Arg}(\mathcal{F})$ is the set of all arguments that can be constructed from \mathcal{F} and \mathcal{U} is the attack relation.

Notation 1. *Let \mathcal{K} be a knowledge base and $\mathcal{H} = (\mathcal{A}, \mathcal{U})$ its corresponding argumentation framework such that $S \subseteq \mathcal{A}$. We denote:*

- $\mathbf{range}^+(a) = \{b \mid (a, b) \in \mathcal{U}\}$, $\mathbf{range}^-(a) = \{b \mid (b, a) \in \mathcal{U}\}$.
- $\mathbf{range}^+(S) = \bigcup_{a \in S} \mathbf{range}^+(a)$ and $\mathbf{range}^-(S) = \bigcup_{a \in S} \mathbf{range}^-(a)$.
- A set of arguments S attacks an argument b if there exists an argument $a \in S$ with $(a, b) \in \mathcal{U}$.

Definition 4 (Semantics). *Let \mathcal{K} be a knowledge base and $\mathcal{H} = (\mathcal{A}, \mathcal{U})$ its corresponding argumentation framework. Let $\mathcal{E} \subseteq \mathcal{A}$ and $a, b \in \mathcal{E}$. We say that \mathcal{E} is conflict free iff there exists no arguments $a, b \in \mathcal{E}$ such that $(a, b) \in \mathcal{U}$. \mathcal{E} defends a iff for every argument $b \in \mathcal{A}$, if we have $(b, a) \in \mathcal{U}$ then \mathcal{E} attacks b . \mathcal{E} is admissible iff it is conflict free and defends all its arguments. \mathcal{E} is a preferred extension iff it is maximal (w.r.t \subseteq) admissible set. \mathcal{E} is a stable extension iff it is conflict-free and for all $a \in \mathcal{A} \setminus \mathcal{E}$, \mathcal{E} attacks a . We denote by $Ext(\mathcal{H})$ the set of all extensions of \mathcal{H} under the preferred/stable semantics. An argument is skeptically accepted if it is in all extensions, credulously accepted if it is in at least one extension and rejected if it is not in any extension.*

It has been show in [14] that argumentation frameworks in our setting are *coherent*, i.e. the stable and the preferred semantics coincide.

Definition 5 (Universal acceptance [2, 14]). *Given an argumentation framework \mathcal{H} over an inconsistent knowledge base \mathcal{K} . A query Q is universally accepted in \mathcal{H} if and only if $\forall \mathcal{E} \in Ext(\mathcal{H})$, $Concs(\mathcal{E}) \models Q$ where $Concs(\mathcal{E}) = \bigcup_{a \in \mathcal{E}} Conc(a)$.*

After introducing the universal acceptance, let us explain why it is different from skeptical acceptance. Note that a query is skeptically accepted if and only if it is entailed by a conclusion of a skeptically accepted argument.

Example 1 (Motivating example). *We consider an inconsistent knowledge base \mathcal{K} : $\mathcal{F} = \{p(a), q(a), r(a)\}$, $\mathcal{R} = \{\forall X (p(X) \rightarrow s(X)), \forall X (q(X) \rightarrow s(X))\}$ and $\mathcal{N} = \{\forall X (p(X) \wedge q(X) \rightarrow \perp)\}$.*

The arguments that can be built from \mathcal{F} are:

- $a_1 = (\{p(a)\}, \{p(a)\})$, $a_2 = (\{q(a)\}, \{q(a)\})$.
- $a_3 = (\{p(a)\}, \{s(a)\})$, $a_4 = (\{q(a)\}, \{s(a)\})$.
- $a_5 = (\{p(a), r(a)\}, \{p(a), r(a)\})$.
- $a_6 = (\{q(a), r(a)\}, \{q(a), r(a)\})$.
- $a_7 = (\{p(a), r(a)\}, \{s(a), r(a)\})$.
- $a_8 = (\{q(a), r(a)\}, \{s(a), r(a)\})$.
- $a_9 = (\{r(a)\}, \{r(a)\})$.

*The attacks are $\mathcal{U} = \{(a_1, a_2), (a_1, a_4), (a_1, a_6), (a_2, a_1), (a_2, a_3), (a_2, a_5), (a_5, a_6), (a_6, a_5), (a_2, a_7), (a_1, a_8)\}$. The preferred extensions: $\mathcal{E}_1 = \{a_1, a_3, a_5, a_7, a_9\}$ and $\mathcal{E}_2 = \{a_2, a_4, a_6, a_8, a_9\}$ with $a_9 = (r(a), r(a))$ being a **skeptical argument**. As one may notice that the query $Q = s(a)$ is **not skeptically accepted but it is universally accepted**. Indeed, $Q = s(a)$ can be deduced from every extension (precisely, from the conclusions of $\{a_3, a_7\} \subset \mathcal{E}_1$ and $\{a_4, a_8\} \subset \mathcal{E}_2$). However, $Q' = r(a)$ is universally and skeptically accepted. Note that the query $Q'' = s(a) \wedge r(a)$ is also universally accepted but not skeptically accepted.*

In what follows we give a fine-grained characterization of universal acceptance that will help us to give a clear proof theory for it. It turns out that universal acceptance can be characterized using the concepts of query supporters, reduct, proponent set and block.

² We follow [6] in considering conjunctions of atoms as atomsets.

Definition 6 (Query's supporters). *Given an argumentation framework \mathcal{H} over an inconsistent knowledge base. The set of all arguments that supports the query \mathcal{Q} is defined as follows:*

$$SUP(\mathcal{Q}) = \{a \mid a \text{ is credulously accepted and } \text{Conc}(a) \models \mathcal{Q}\}$$

Definition 7 (Reduct of extension). *Given an extension $\mathcal{E} \subseteq \mathcal{A}$ and a query \mathcal{Q} . The reduct $\mathcal{E}^\mathcal{Q} \subseteq \mathcal{E}$ of the extension \mathcal{E} w.r.t the query \mathcal{Q} is defined as the non-empty intersection $SUP(\mathcal{Q}) \cap \mathcal{E}$. The reduct of the set of all extensions $\text{Ext}(\mathcal{H})$ w.r.t \mathcal{Q} is defined as $\text{Ext}(\mathcal{H})^\mathcal{Q} = \{\mathcal{E}^\mathcal{Q} \mid \mathcal{E} \in \text{Ext}(\mathcal{H})\}$.*

The reduct $\mathcal{E}^\mathcal{Q}$ of the extension \mathcal{E} w.r.t the query \mathcal{Q} is defined as the set of all supporters of \mathcal{Q} which belong to \mathcal{E} . This means that a complete set of reducts covers the set of all extensions.

Definition 8 (Complete reduct). *The set of all reducts $\text{Ext}(\mathcal{H})^\mathcal{Q}$ w.r.t a query \mathcal{Q} is complete if and only if there exists no $\mathcal{E} \in \text{Ext}(\mathcal{H})$ such that $\mathcal{E}^\mathcal{Q} \notin \text{Ext}(\mathcal{H})^\mathcal{Q}$.*

An incomplete reduct corresponds to the case where there is an extension that does not contain any supporter.

Proposition 1. *A query \mathcal{Q} is credulously accepted if and only if $\text{Ext}(\mathcal{H})^\mathcal{Q} \neq \emptyset$. A query \mathcal{Q} is universally accepted if and only if $\text{Ext}(\mathcal{H})^\mathcal{Q}$ is complete.*

The proponent set is similar to the concept of a complete base in [29]. Before defining it we need the concept of a hitting set.

Definition 9 (Hitting set). *Given a collection $\mathcal{C} = \{S_1, \dots, S_m\}$ of finite nonempty subsets of a set \mathcal{B} (the background set). A hitting set of \mathcal{C} is a set $A \subseteq \mathcal{B}$ such that $S_j \cap A \neq \emptyset$ for all $S_j \in \mathcal{C}$. A hitting set of \mathcal{C} is minimal (w.r.t \subseteq) if and only if no proper subset of it is a hitting set of \mathcal{C} . A minimum hitting set is a minimal hitting set w.r.t set-cardinality.*

Definition 10 (Proponent set). *A set of arguments $S \subseteq \mathcal{A}$ is a proponent set of \mathcal{Q} if and only if S is a minimal (w.r.t \subseteq) hitting set of $\text{Ext}(\mathcal{H})^\mathcal{Q}$.*

Proposition 2. *A query \mathcal{Q} is universally accepted if and only if it has a proponent set.*

It is clear that a proponent set holds the smallest set of arguments which are distributed over all extensions and support the query \mathcal{Q} . So, if one extension does not contain any supporter then the query is not universally accepted. The reason for the absence of such supporter is what we call the presence of a *block*. We follow the notion of a **block** from [25] and instantiate it in our setting. A block B is a set of arguments which are (1) all credulously accepted, (2) attack all the supporters of \mathcal{Q} , and (3) they can all together be extended to form an extension.

Definition 11 (Block). *Let \mathcal{Q} be a query and let $\mathcal{C} = \{\text{range}^-(a) \mid a \in SUP(\mathcal{Q})\}$. A set of arguments $B \subseteq \mathcal{A}$ is a block of \mathcal{Q} if and only if: (1) B is a hitting set of \mathcal{C} ; and, (2) there exists an admissible set $A \subseteq \mathcal{A}$ such that $B \subseteq A$.*

While a query may have more than one block or more than one proponent set, it is never the case that it has the two together.

Proposition 3. *A query \mathcal{Q} has a block iff \mathcal{Q} has no proponent set.*

Consequently, a query is not universally accepted iff it has a block.

2.2 Universal Dialectical Proof Theory

Given a query \mathcal{Q} and an argumentation framework \mathcal{H} , the universal dialectical proof theory is a two-person argument game between a proponent (PRO) and an opponent (OPP). The proponent and the opponent are engaged in an argumentation dialogue of *precisely defined* types of moves respecting a turn taking mechanism. The turn taking mechanism is deterministic where odd indexed moves are advanced by PRO and even index moves are advanced by OPP. The moves of the dialogue are defined in terms of speech acts: support, counter and retrace. The move $SUPPORT(a)$ advances an argument a which supports the query in question. The move $COUNTER(A)$ counterattacks the position of PRO by advancing a set of arguments that attack the previously advanced supporters. The move $RETRACE(A, i)$ is used to retrace to the stage i in the dialogue. The dialogue is asymmetric where $SUPPORT$ can only be played by PRO, whereas $COUNTER$ and $RETRACE$ ³ can only be played by OPP.

Definition 12 (Dialogue). *Let $\mathcal{H} = (\mathcal{A}, \mathcal{U})$ be an argumentation framework. A dialogue based on \mathcal{H} is a finite sequence $d_n = (m_1, \dots, m_n)$ of moves where each m_j is either:*

- **Support move:** $m_j = SUPPORT(a)$ such that $a \in \mathcal{A}$ (In this case we denote $\text{Arg}(m_j) = a$ and $\text{Sp}(m_j) = SUPPORT$).
- **Counter move:** $m_j = COUNTER(A)$ such that $A \subseteq \mathcal{A}$ (In this case we denote $\text{Arg}(m_j) = A$ and $\text{Sp}(m_j) = COUNTER$).
- **Retrace move:** $m_j = RETRACE(A, i)$ such that $A \subseteq \mathcal{A}$ and $i < j$ (In this case we denote $\text{Arg}(m_j) = A$, $\text{Sp}(m_j) = RETRACE$).

*Odd-indexed (resp. even-indexed) moves are played by PRO (resp. OPP). We denote by $d \cdot d'$ and $d \cdot m$ the concatenation of the dialogues d and d' and the dialogue d with the move m respectively. The retrace move has a special parameter i called the index (denoted as $\text{Idx}(m)$). The subscript of d_n refers to the stage of the dialogue.*⁴

Let \mathcal{Q} be a query, any dialogue of this dialectical theory starts by PRO advancing $SUPPORT(a)$ that supports \mathcal{Q} (i.e. $a \in SUP(\mathcal{Q})$). Then, OPP presents an argument (or a set of arguments) that attacks the previously advanced argument. Next, PRO tries to avoid this attack and reinstate the query using another argument which is not attacked by the already advanced attackers. OPP in turn, tries to extend the previous set of attackers so that it attacks all the supporters advanced so far. When OPP fails to extend the set, it retraces back and chooses another set of attackers and continues the dialogue from thereafter. By doing so OPP is somehow trying to construct a set of arguments that attack all the supporters of the query \mathcal{Q} , i.e. a *block* for \mathcal{Q} .

Following [17] we introduce the notion of a *dialectical state* which helps in controlling the dialogue.

Definition 13 (Dialectical state). *Let d_k be a dialogue at stage k . The dialectical state of d_k is a tuple $\delta_k = (\pi_k, h_k, \theta_k, \beta_k, \Delta_k)$ ⁵:*

- π_k : the set of arguments available to PRO.
- h_k : the set of arguments that have been played so far by PRO.
- θ_k : the set of arguments available to OPP.
- β_k : the current block constructed by PRO.
- Δ_k : the sets of arguments that have been shown to be not blocks.

d_0 is the empty dialogue and δ_0 is its initial dialectical state.

³ When there is no risk of ambiguity we refer to moves by their speech acts.

⁴ We may sometimes omit the subscript when it is not needed.

⁵ To be able to understand the terms think of π as the first letter of **proponent**, h as **history**, θ as **opponent** and β as **block**.

A dialectical state defines at any stage k of the dialogue d_k the set of arguments π_k available to PRO to be used in order to support the query \mathcal{Q} . In the dialectical state, we find also the set h_k that shows the arguments so far played by PRO. In addition, it presents the set θ_k of arguments that can be used to attack the arguments previously advanced by PRO. β_k presents the currently constructed block. When OPP fails to extend the current block to another that attacks all the previously played supporters, the RETRACE move is used. By doing so we keep track of the sets of arguments that cannot be extended to blocks. These are stored in Δ_k .

At the beginning stage, when the dialogue d_0 has not yet been started, the set of available arguments π_0 for PRO ranges over all the possible supporters of the query \mathcal{Q} . The played arguments h_0 , the available arguments θ_0 , current block β_0 and Δ_0 are empty.

2.3 Dialogue Rules

The advancement of moves within the dialogue are usually controlled by a legal move function [25] which can be expressed in terms of rules, called dialogue rules. Every move depends on certain *preconditions* about the actual dialectical state and the previous move advanced by the other party. Every move also determines the next move to be played (*postcondition*).

Let d_k be a dialogue and δ_k the current dialectical state of d_k . Let m_{k+1} be a move and δ_{k+1} be the dialectical state of the dialogue $d_{k+1} = d_k \cdot m_{k+1}$ after playing the move m_{k+1} . For a given move we index preconditions (resp. effects) by the first letter of the speech act of the move followed by P (resp. E) and subscripted by a number.

Move:

$$m_{k+1} = \text{SUPPORT}(a).$$

Description:

advances an argument that supports the query in question.

Preconditions:

$$(\text{SP}_1) \quad k + 1 \text{ is odd.}$$

$$(\text{SP}_2) \quad a \in \pi_k.$$

Postconditions:

the next move can be either COUNTER or RETRACE.

Effects:

$$(\text{SE}_1) \quad \pi_{k+1} = \pi_k / a.$$

$$(\text{SE}_2) \quad h_{k+1} = h_k \cup \{a\}.$$

$$(\text{SE}_3) \quad \theta_{k+1} = \text{range}^-(h_{k+1}).$$

$$(\text{SE}_4) \quad \beta_{k+1} = \beta_k.$$

$$(\text{SE}_5) \quad \Delta_{k+1} = \Delta_k.$$

This move is advanced by PRO, therefore $k + 1$ should be odd (SP_1). It advances an argument a that supports the query \mathcal{Q} (SP_2). To respond, OPP should either use COUNTER or RETRACE.

As one may notice, the support move m_{k+1} changes the set of available arguments π_{k+1} of PRO. In fact a supporting argument ceases to be available once it is played (SE_1). In contrast it is added to the history h_{k+1} . The support move alters the set of available arguments of OPP by adding to θ_{k+1} all the arguments that can be played in the future by OPP (SE_3), which are those that can attack the advanced supporting arguments. As indicated in the postconditions of the support move, a counter move is allowed to be played next.

Move:

$$m_{k+1} = \text{COUNTER}(A).$$

Description:

this move advances a set of arguments that attacks all the arguments presented so far.

Preconditions:

$$(\text{CP}_1) \quad k + 1 \text{ is even.}$$

$$(\text{CP}_2) \quad A = \beta_k \cup S \text{ such that } S \subseteq \theta_k \text{ (i.e. } A \text{ extends } \beta_k \text{ by } S).$$

$$(\text{CP}_3) \quad A \text{ attacks } h_k \text{ and belongs to (or is) an admissible set.}$$

$$(\text{CP}_4) \quad \text{there is no } A' \in \Delta_k \text{ such that } A' \subseteq A.$$

Postconditions:

the next move should be SUPPORT.

Effect:

$$(\text{CE}_1) \quad \pi_{k+1} = \pi_k / \text{range}^+(A).$$

$$(\text{CE}_2) \quad h_{k+1} = h_k.$$

$$(\text{CE}_3) \quad \theta_{k+1} = \theta_k.$$

$$(\text{CE}_4) \quad \beta_{k+1} = A.$$

$$(\text{CE}_5) \quad \Delta_{k+1} = \Delta_k.$$

This move is advanced by OPP therefore $k + 1$ should be even (CP_1). It tries to extend the current block β_k to another set of arguments that attacks all the supporters presented so far (CP_2 and CP_3). OPP does so by incorporating arguments from θ_k . The new current block ($\beta_{k+1} = A$) or one of its subsets should have not been already proven to be not a block (CP_4). After advancing m_{k+1} , π_{k+1} contains all the arguments from π_{k+1} except those which are attacked by A (CE_1), thus they are spared from further use. Note that the spared arguments may be readded afterwards, this is particularly the case when we use retrace as we shall mention later.

Since A attacks all the supporting arguments so far provided, it is considered the current block (CE_4). The sets θ_{k+1} , Δ_{k+1} and h_{k+1} are left unchanged (CE_2 , CE_3 and CE_5).

After a support move, OPP can also play a retrace move. This is particularly needed when he is unable to play a counter move.

Move:

$$m_{k+1} = \text{RETRACE}(A, i).$$

Description:

this move retraces to the recent stage i from which it can extend the current block of i .

Preconditions:

$$(\text{RP}_1) \quad k + 1 \text{ is even, } i < k + 1 \text{ and } i \text{ is odd.}$$

$$(\text{RP}_2) \quad \text{there is no set of arguments } S \subseteq \theta_k \text{ such that } \beta_k \cup S \text{ is (or belongs to) an admissible set and attacks } h_k.$$

$$(\text{RP}_3) \quad A = \beta_i \cup S \text{ such that } S \subseteq \theta_i.$$

$$(\text{RP}_4) \quad A \text{ attacks } h_i \text{ and belongs to (or is) an admissible set.}$$

$$(\text{RP}_5) \quad \text{there is no } A' \in \Delta_k \text{ such that } A' \subseteq A.$$

Postconditions:

the next move should be SUPPORT.

Effect:

$$(\text{RE}_1) \quad \pi_{k+1} = \pi_i / \text{range}^+(A).$$

$$(\text{RE}_2) \quad h_{k+1} = h_i.$$

$$(\text{RE}_3) \quad \theta_{k+1} = \theta_i.$$

$$(\text{RE}_4) \quad \beta_{k+1} = A.$$

$$(\text{RE}_5) \quad \Delta_{k+1} = \Delta_k \cup \beta_k.$$

When OPP cannot extend the current block β_k with arguments from θ_k (RP_2), it should retrace back and choose other arguments.

The index i (which should be odd) determines the point of a support move from which OPP can mount another line of attack. By starting a new line of attack, OPP should opt for a new block that attacks all the supporters from stage i up to the stage 1 (RP₃) by extending β_i using θ_i . The new block $\beta_{k+1} = A$ or one of its subsets should have not been already proven to be not a block (CP₅).

When the retrace move is advanced, π_{k+1} is reset to its ancient state i in addition to excluding all the arguments that can be attacked afterwards (RE₁). The current block β_{k+1} is set to A (RE₄), and Δ_{k+1} is set to $\Delta_k \cup \beta_k$ (RE₅), i.e. the block of stage k that OPP could not extend. If one precondition is not satisfied OPP looks for other stages to build a new attack. OPP follows the procedure:

Procedure 1. Let d_n be a dialogue and m_n be the last played move such that $\text{Sp}(m_n) = \text{SUPPORT}$. If OPP cannot play a counter move m_{n+1} then it tries to play the retrace move m_{n+1} as follows:⁶

1. do $y = y - 1$ until $m_y = \text{RETRACE}(A, x)$ or $m_y = \text{SUPPORT}(a)$.
2. if $m_y = \text{RETRACE}(A, x)$ then:
 - (a) play $m_{n+1} = \text{RETRACE}(A', x)$ that respects the preconditions and exit. If there isn't such move then set $y = x$ and goto 1.
3. if $m_y = \text{SUPPORT}(a)$ then:
 - (b) play $m_{n+1} = \text{RETRACE}(A', y)$ that respects the preconditions and exit. If there is no such move then goto 1.

OPP starts by looking for the most recent retrace or support move (line 1). If a retrace move is found (line 2) then it tries to play a retrace to stage x that respects the preconditions (line a) by looking exhaustively for all possible sets A' that makes the move respect the preconditions. If he succeeds to play such move, the procedure exits. Otherwise it continues the search by setting y to x . If a support move m_y is found (line 3) then it plays a retrace with index to y . Otherwise, it continues the search for other moves from which OPP can play.

The dialogue represents a compact representation of a tree where nodes are arguments or set of arguments played by both parties. Nodes in the odd levels are played by PRO and nodes in the even level are played by OPP. This tree is called the dialogue tree. In this tree retrace moves represent branching points.

Definition 14 (Dialogue tree). Given a dialogue $d_n = (m_1, \dots, m_n)$, its dialogue tree is a labeled tree $\mathcal{T}(d_n) = (\mathcal{V}, \mathcal{D})$ where \mathcal{V} is a set of nodes and \mathcal{D} is a binary relation over \mathcal{V} . $\mathcal{T}(d_n)$ is defined as follows: while $n > 0$, $\mathcal{D}(d_n)$ is recursively defined as:

$$\begin{cases} \mathcal{D}(d_n) = \emptyset & n = 1 \\ \mathcal{D}(d_{n-1}) \cup \{(\text{Arg}(m_i), \text{Arg}(m_n))\} & n > 1, m_n = \text{RETRACE}(A, i) \\ \mathcal{D}(d_{n-1}) \cup \{(\text{Arg}(m_{n-1}), \text{Arg}(m_n))\} & n > 1, m_n \neq \text{RETRACE}(A, i) \end{cases}$$

The set of all nodes is defined as $\mathcal{V} = \{\text{Arg}(m_i) \mid m_i \in d_n\}$ with $\text{Arg}(m_1)$ as the root node of the tree. Note that $|\mathcal{T}(d_n)| = |\mathcal{V}|$ refers to the size of the tree which is equal to the number of its nodes.

Note that this dialogue tree is similar to the well-known dispute tree in the work of [17, 25]. The difference is in the nature of nodes where the nodes of OPP in the dialogue tree contain set of arguments as opposed to a dispute tree.

The dialogue tree enjoys the following properties.

Proposition 4. Let d_n be a dialogue, $\mathcal{T}(d_n)$ its dialogue tree and $\text{Pre}(\mathcal{T}(d_n))$ the pre-order traversal of $\mathcal{T}(d_n)$ with $\text{Seq}(d_n) = (c_1, \dots, c_n)$ such that $c_i = \text{Arg}(m_i)$. The following hold:

⁶ Note that y is initialized to n and $x < y$, and a, A are arbitrary (set of) arguments respectively.

1. $\mathcal{T}(d_n)$ is unique.
2. $|d_n| = |\mathcal{T}(d_n)|$.
3. $\text{Pre}(\mathcal{T}(d_n)) = \text{Seq}(d_n)$.

What is left for the dialectical proof theory is to determine the termination condition.

Definition 15 (Termination and wining). A dialogue d_n is a terminated dialogue if and only if neither PRO nor OPP can play a move. The winner of d_n is the player of the last move m_n .

It is easy to determine the winner of a dialogue from its tree:

Proposition 5. Let d_n be a dialogue, $\mathcal{T}(d_n)$ its dialogue tree. The following statements are equivalent:

- the length of the rightmost path is odd.
- PRO is the winner of d_n .

We conclude the section by defining a dialectical proof.

Definition 16 (Dialectical proof). Given a query \mathcal{Q} and a terminated dialogue d_n about \mathcal{Q} . We call d_n a dialectical proof for the universal acceptance of \mathcal{Q} if and only if PRO is the winner.⁷

In Section 4 we provide the properties of the dialectical proof theory. We now give an example to better illustrate it.

3 Illustrating Example

Consider the argumentation framework \mathcal{H} of Figure 1. This argumentation framework is coherent (preferred and stable extensions coincide). Suppose that the gray-colored arguments support a query \mathcal{Q} (i.e. $\text{SUP}(\mathcal{Q}) = \{a, d, e, l, h\}$). In what follows, we show how the query \mathcal{Q} is universally accepted by providing a dialectical proof.

The dialectical proof is presented in Table 1 and its dialogue tree is shown in Figure 2.

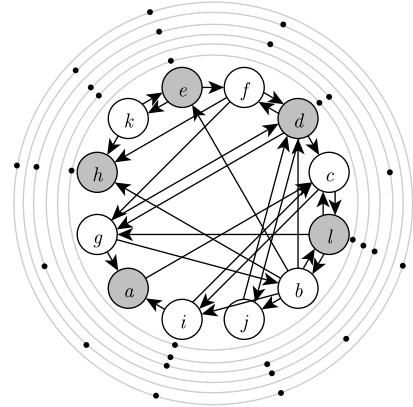


Figure 1: The argument graph. The circles are the extensions presented in an increasing order from \mathcal{E}_1 to \mathcal{E}_6 with the extension \mathcal{E}_1 being the inner circle and the extension \mathcal{E}_6 the outer circle.

At stage (0) the dialectical state is initialized as defined previously. The dialogue starts at stage (1) by PRO playing the supporter a from the available supporters in π_0 . When PRO plays a , the argument a is moved from the available supporters π_1 to the history of advanced arguments h_1 by PRO. The set of available attackers θ_1 becomes the set of all attackers that can attack h_1 . This means when the turn of

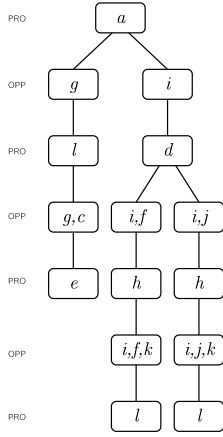
⁷ Otherwise it is called a dialectical proof for non-universal acceptance of \mathcal{Q} .

i	Move	π_i	h_i	θ_i	β_i	Δ_i
0	-	$\{a, d, e, l, h\}$	\emptyset	\emptyset	\emptyset	\emptyset
1	S(a)	$\{d, e, l, h\}$	$\{a\}$	$\{g, i\}$	\emptyset	\emptyset
2	C($\{g\}$)	$\{e, l, h\}$	$\{a\}$	$\{g, i\}$	$\{g\}$	\emptyset
3	S(l)	$\{e, h\}$	$\{a, l\}$	$\{g, i, c, b\}$	$\{g\}$	\emptyset
4	C($\{g, c\}$)	$\{e, h\}$	$\{a, l\}$	$\{g, i, c, b\}$	$\{g, c\}$	\emptyset
5	S(e)	$\{h\}$	$\{a, l, e\}$	$\{g, i, c, b, k\}$	$\{g, c\}$	\emptyset
6	R($\{i\}, 1$)	$\{d, e, l, h\}$	$\{a\}$	$\{g, i\}$	$\{i\}$	$\{\beta_5\}$
7	S(d)	$\{e, l, h\}$	$\{a, d\}$	$\{g, i, f, j, b\}$	$\{i\}$	$\{\beta_5\}$
8	C($\{i, f\}$)	$\{l, h\}$	$\{a, d\}$	$\{g, i, f, j, b\}$	$\{i, f\}$	$\{\beta_5\}$
9	S(h)	$\{l\}$	$\{a, d, h\}$	$\{g, i, f, j, b, k\}$	$\{i, f\}$	$\{\beta_5\}$
10	C($\{i, f, k\}$)	$\{l\}$	$\{a, d, h\}$	$\{g, i, f, j, b, k\}$	$\{i, f, k\}$	$\{\beta_5\}$
11	S(l)	\emptyset	$\{a, d, h, l\}$	$\{g, i, f, j, b, k, c\}$	$\{i, f, k\}$	$\{\beta_5\}$
12	R($\{i, j\}, 7$)	$\{e, l, h\}$	$\{a, d\}$	$\{g, i, f, j, b\}$	$\{i, j\}$	$\{\beta_5, \beta_{11}\}$
13	S(h)	$\{e, l\}$	$\{a, d, h\}$	$\{g, i, f, j, b, k\}$	$\{i, j\}$	Δ_{12}
14	C($\{i, j, k\}$)	$\{l\}$	$\{a, d, h\}$	$\{g, i, f, j, b, k\}$	$\{i, j, k\}$	Δ_{12}
15	S(l)	\emptyset	$\{a, d, h, l\}$	$\{g, i, f, j, b, k, c\}$	$\{i, j, k\}$	$\Delta_{12} \cup \{\beta_{14}\}$

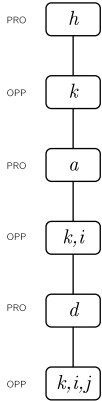
i	Move	π_i	h_i	θ_i	β_i
0	-	$\{a, d, e, h\}$	\emptyset	\emptyset	\emptyset
1	S(h)	$\{a, d, e\}$	$\{h\}$	$\{k, f, b\}$	\emptyset
2	C($\{k\}$)	$\{a, d\}$	$\{h\}$	$\{k, f, b\}$	$\{k\}$
3	S(a)	$\{d\}$	$\{a, h\}$	$\{k, f, b, g, i\}$	$\{k\}$
4	C($\{k, i\}$)	$\{d\}$	$\{a, h\}$	$\{k, f, b, g, i\}$	$\{k, i\}$
5	S(d)	\emptyset	$\{d, a, h\}$	$\{k, f, b, g, i, j\}$	$\{k, i\}$
6	C($\{k, i, j\}$)	\emptyset	$\{d, a, h\}$	$\{k, f, b, g, i, j\}$	$\{k, i, j\}$

Table 2: A dialectical proof for the non-universal acceptance of \mathcal{Q}' . Note that we omit Δ_i as it is always empty in this example.

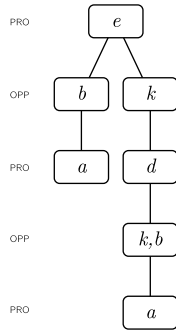
Table 1: A dialectical proof for the query \mathcal{Q} . For space reasons S(), C() and R() denote SUPPORT(), COUNTER() RETRACE() respectively.



The dialogue tree for universal acceptance \mathcal{Q} .



(a) Dialogue tree for the non-universal acceptance of \mathcal{Q}' .



(b) Another dialogue tree for the universal acceptance of \mathcal{Q} .

Figure 3: Dialogue trees for the the illustrating example.

OPP comes at stage (2) he shall choose from this set. At stage (2) OPP advances a counter move with argument g that attacks all the advanced supporters (i.e. $h_1 = \{a\}$). After advancing such move, the argument d is removed from the set of available arguments π_2 since g attacks d , thus PRO will not be able to play d . Since $\{g\}$ attacks

all the supporters advanced so far, it becomes the current block, i.e. $\beta_2 = \{g\}$. At stage (3), PRO responds by a support move with the argument l that is not attacked by the current block. At stage (4), OPP extends the current block $\beta_3 = \{g\}$ by the argument c which attacks l . Note that $\{g, c\}$ is a subset of the admissible set $\{g, c, e\}$. Now, $\beta_4 = \{g, c\}$ attacks all the presented supporters. At stage (5), PRO presents another unattacked supporter (i.e. e). Note that the choice of the supporters is arbitrary.

At stage (6), OPP could not extend the current block β_5 into another that attacks e too. Therefore OPP plays a retrace move $R(\{i\}, 1)$ that can be read as “retrace to stage (1) and play a counter move with $\{i\}$ ”. By doing so, OPP creates another line of dialogue and roll back all the changes that have been made on the dialectical state up to the stage (1). That is why at stage (6) the sets π_6, h_6 and θ_6 are computed with respect to π_1, h_1 and θ_1 . The current block is changed to $\{i\}$ and the ancient block β_5 is moved to $\Delta_6 = \{\beta_5\}$. The former would say that this set or any of its supper sets will never form a block. This is important to avoid unnecessary moves. The same thing happens at stage (12) where OPP retraces to stage (7) because he could not retrace to the stage (9). The current block β_{12} is set to $\{i, j\}$ which extends β_7 .

The dialogue continues until stage (15) where PRO plays a support move with argument l against which OPP could neither attack nor retrace to previous stages. For instance, OPP has not been able to extend $\{i, j, k\}$ by c because it would be conflicting (not admissible). At this stage the dialogue ends and PRO is declared as the winner.

The dialogue tree in Figure 2 shows clearly the relation between the advanced arguments played by both parties. The tree in Figure 3b is another dialogue tree for another dialogue where PRO is the winner. This can be easily observed since all leaf nodes are in odd levels.

Let us now take an example where the query is not universally accepted. Consider a query \mathcal{Q}' that happens to have the supporters $SUP(\mathcal{Q}') = \{a, d, e, h\}$. The dialogue is presented in Table 2 and its dialogue tree is shown in Figure 3a. In this example, OPP has been able to construct the block $\beta_6 = \{k, i, j\}$ in the last move which attacks all the supporters. This made PRO unable to continue the dialogue. Note that we do not allow retracing for PRO because one block is sufficient to prove the non-universal acceptance.

4 Dialectical Proof Theory Properties

4.1 Finiteness, Soundness and Completeness

As indicated in [3, 21], finiteness or termination is an important property for any dialogue, since a possibly infinite dialogue will fail to meet the intended goal. In what follows we show how our dialectical theory produces always finite dialogues.

To establish such property we need to show that for any dialogue d its dialogue tree is finite. Such result can be established by showing that the height of the tree is finite and that for each node the number of its child nodes is finite.

Lemma 1. *Let \mathcal{H} be an argumentation framework, \mathcal{D}^∞ be the set of all possible dialogues over \mathcal{H} , $\text{Height}(\mathcal{T}(d))$ is the height of the tree $\mathcal{T}(d)$ and $\mathcal{C}(v)$ is the set of all child nodes of v . Given $\mathcal{T}(d) = (\mathcal{V}, \mathcal{D})$ of any $d \in \mathcal{D}^\infty$ the following hold:*

1. $\text{Height}(\mathcal{T}(d)) \in \mathbb{N}$.
2. $\forall v \in \mathcal{V}, |\mathcal{C}(v)| \in \mathbb{N}$.

Proof. Let us suppose that $\text{Height}(\mathcal{T}(d))$ is infinite, and let P be the longest path in $\mathcal{T}(d)$ starting from the root node. This means either there are infinitely many supporting arguments used in P , or there are some infinity repeated supporting arguments used in P . The first one is impossible since we are dealing with finite argumentation framework (the set of all arguments is finite). The second is impossible since once an argument is played it cannot be advanced afterwards in the same path (see SE_1 of SUPPORT move).

Let us suppose that $|\mathcal{C}(v)|$ is infinite. This means that either (i) v is a supporting argument and it has infinitely many attacker; or (ii) v contains arguments that are advanced to attack previous supporters. The first case is impossible since the argumentation framework is finite, and the second is impossible since if it were the case then PRO would be allowed to retrace against counter moves, which is forbidden in our framework. \square

Now we can proceed to finiteness by showing the following.

Proposition 6 (Finiteness). *Let \mathcal{H} be an argumentation framework and \mathcal{D}^∞ be the set of all possible dialogues over \mathcal{H} . Then for every $d \in \mathcal{D}^\infty$: $|d| \in \mathbb{N}$.*

Proof. Let us suppose that d is infinite. This means, either (i) $\text{Height}(\mathcal{T}(d))$ is infinite; or (ii) there is a node in $\mathcal{T}(d_n)$ with infinitely many child nodes. From the previous lemma, the two cases are impossible. \square

In [3] an additional constraint has been added to finiteness, i.e. the finiteness of the moves' contents. This constraint ensures that the arguments advanced within the dialogue are finite. In our context we distinguish two cases, (i) the argument in the support moves should be finite, and (ii) the set of arguments advanced in the counter moves should be finite too. Fortunately, the two cases are verified in our argumentation framework because the set of arguments \mathcal{A} for any argumentation framework over a possibly inconsistent knowledge base (in our logical setting) is finite and the set of attackers for a given argument is finite. All in all, the argumentation framework is finite.

Before proceeding to soundness let us show that the dialectical proof theory is *consistent* in the sense that there is no two dialogues about a query \mathcal{Q} such that PRO wins in the former and loses in the later. Put differently, if one of the participant wins a dialogue about a given query \mathcal{Q} then we are sure that he will win all the other dialogues about \mathcal{Q} .

Proposition 7 (Consistency). *Let $\mathcal{D}_\mathcal{Q}^\infty \subseteq \mathcal{D}^\infty$ be the set of all dialogues about \mathcal{Q} in \mathcal{H} and let $d \in \mathcal{D}_\mathcal{Q}^\infty$. Then, if d is won by PRO (resp. OPP) then so are all $d \in \mathcal{D}_\mathcal{Q}^\infty$.*

This property is very important since we do not want to have a dialectical proof theory that is contradictory. It turns out that this property is important for soundness. In what follows, soundness is characterized by the existence of a winning dialogue (by PRO or OPP).

Proposition 8 (Soundness). *Given a dialogue d about the query \mathcal{Q} , if d is won by PRO then \mathcal{Q} is universally accepted.*

Proof. Let us proceed by contradiction. Suppose that d is won by PRO but \mathcal{Q} is not universally accepted. On the one hand, recall that if \mathcal{Q} is not universally accepted then there exists a block B against all \mathcal{Q} 's supporters. On the other hand, if PRO has won d then PRO could not find any block that attacks all supporters advanced in d . This means that either (i) OPP search was not exhaustive or (ii) there is no such block. As one can see, (ii) is in contradiction with the assumption and (i) is in contradiction with the fact that the move procedure is exhaustive. \square

If the dialectical proof theory is sound but does not provide dialectical proofs for all universally (resp. non-universally) accepted queries then it would be incomplete.

Proposition 9 (Completeness). *Given a query \mathcal{Q} . If \mathcal{Q} is universally accepted then PRO wins any dialogue about \mathcal{Q} .*

Proof. By contradiction, if \mathcal{Q} is universally accepted and PRO loses then OPP has constructed a block β_n for \mathcal{Q} . This means that \mathcal{Q} is not universally accepted, which is a contradiction. \square

In this subsection we have proved the finiteness, completeness and soundness of the proposed theory as well as its consistency.

4.2 Dispute Complexity

In this subsection we are interested in the question of how many moves the dialogue would contain for a query (at best-case) to establish its universal acceptance (non-universal acceptance). The work of [17] introduced the so-called dispute complexity for a given argument in a given argumentation framework. We adapt this definition and define the dispute complexity for a given query over a given instantiated argumentation framework as follows.

Definition 17 (Dispute complexity). *Let \mathcal{H} be an argumentation framework and \mathcal{Q} be a query. The dispute complexity $\delta(\mathcal{H}, \mathcal{Q})$ of the query \mathcal{Q} in \mathcal{H} is defined as follows:*

$$\delta(\mathcal{H}, \mathcal{Q}) = \min\{|d| : d \text{ is a terminated dialogue about } \mathcal{Q} \text{ in } \mathcal{H}\}$$

The dispute complexity is the minimal number of moves that can be used to prove that \mathcal{Q} is universally accepted or not universally accepted. The work of [17] has given an exact characterization of such complexity for credulous acceptance by considering as an input the argumentation framework and all admissible sets. Our goal in what follows is to propose some bounds for such complexity in universal (or non-universal) acceptance.

Let \mathcal{Q} be a query, \mathcal{H} an argumentation framework such that \mathcal{Q} is not universally accepted in \mathcal{H} and $\mathcal{C} = \{\text{range}^-(a) | a \in \text{SUP}(\mathcal{Q})\}$. We will use the following notations:

- $MHS(\mathcal{H}, \mathcal{Q})$ is the set of all minimal (w.r.t \subseteq) hitting sets of \mathcal{C} .
- $MinBS(\mathcal{H}, \mathcal{Q})$ denotes the set of all minimal (w.r.t \subseteq) blocks.

- the block number of \mathcal{Q} in \mathcal{H} is the size of the minimum block:
 $\tau(\mathcal{H}, \mathcal{Q}) = \min(|B| : B \in \text{MinBS}(\mathcal{H}, \mathcal{Q}))$.
- the hitting set number is the size of the minimum hitting set of \mathcal{C} :
 $\alpha(\mathcal{H}, \mathcal{Q}) = \min(|S| : S \in \text{MHS}(\mathcal{H}, \mathcal{Q}))$.

The block number corresponds to the minimum block which is the smallest block (*w.r.t set-cardinality*) among all blocks. Note that it is not necessary that every minimum hitting set of \mathcal{C} is a minimum block (because a block imposes that its members have to belong to the same admissible set). Therefore it is possible to have a block which is minimum but does not correspond to any minimum/minimal hitting set. In contrast, a minimum block has to be a hitting set. We get the following straightforward relation.

Corollary 1. $\tau(\mathcal{H}, \mathcal{Q}) \geq \alpha(\mathcal{H}, \mathcal{Q})$.

In the context of a dialogue about a query \mathcal{Q} , the minimum block represents what the opponent would play in order to finish the dialogue as fast as possible. Therefore, the dispute complexity of non-universal acceptance can be characterized by such number.

Proposition 10. For any terminated dialogue d about \mathcal{Q} in an argumentation framework \mathcal{H} where \mathcal{Q} is not universally accepted:

$$\delta(\mathcal{H}, \mathcal{Q}) = 2 \times \tau(\mathcal{H}, \mathcal{Q}).$$

Sketch. If the size of the minimum block B equals n then at each stage OPP will extend his current block by advancing one attacker at each stage. Therefore, for each SUPPORT move we will have a COUNTER move that extends the current block by one argument. When the current block reaches the size n , that means OPP has played all the arguments of the minimum block, PRO will have no supporting argument to advance, thus the dialogue terminates after $2 \times n$ moves. \square

The property above provides an exact bound for the dispute complexity of the non-universal acceptance. The proposition below gives the upper bound for universal acceptance. In order to define this we define the attack degree of \mathcal{Q} in \mathcal{H} as $\text{deg}(\mathcal{H}, \mathcal{Q}) = \max(|\text{range}^-(a)| : a \in \mathcal{P}(\mathcal{Q}))$ such that $\mathcal{P}(\mathcal{Q})$ is the set all supporting arguments that belongs to at least one minimum proponent set. And the proponent number is the size of the minimum proponent set $\rho(\mathcal{H}, \mathcal{Q}) = \min(|S| : S \text{ is a proponent set of } \mathcal{Q} \text{ in } \mathcal{H})$. It is obvious that dialogues where PRO plays with minimum proponent sets are shorter than all other dialogues. Because in the latter dialogues PRO will play only the support moves that are needed to terminate the dialogue. Let $\Theta(\mathcal{H}, \mathcal{Q})$ be the size of the shortest dialogue where PRO plays only with a minimum proponent set. It is clear that $\delta(\mathcal{H}, \mathcal{Q}) \leq \Theta(\mathcal{H}, \mathcal{Q})$. To bound $\delta(\mathcal{H}, \mathcal{Q})$ we need to bound $\Theta(\mathcal{H}, \mathcal{Q})$. The latter can be bounded by imagining that the dialogue tree would have at worst-case the height equals to $2 \times \rho(\mathcal{H}, \mathcal{Q})$ and each proponent node (even-indexed) has exactly $\text{deg}(\mathcal{H}, \mathcal{Q})$ child (worst-case). Therefore, the upper-bound for $\Theta(\mathcal{H}, \mathcal{Q})$ is the size (number of nodes) of the this dialogue tree.

Proposition 11. For any dialogue d about \mathcal{Q} in an argumentation framework \mathcal{H} where \mathcal{Q} is universally accepted the following holds:

$$\delta(\mathcal{H}, \mathcal{Q}) \leq 2 \times \left(\frac{\text{deg}(\mathcal{H}, \mathcal{Q})^{\rho(\mathcal{H}, \mathcal{Q})} - 1}{\text{deg}(\mathcal{H}, \mathcal{Q}) - 1} - 1 \right)$$

Example 2. Consider the argumentation framework of Figure 1. Let $\text{SUP}(\mathcal{Q}'') = \{a, d, e, l, h, k\}$ be the set of supporters for an arbitrary universally accepted query \mathcal{Q}'' . Then, the query has two minimum proponent sets $\mathcal{P} = \{\{k, h\}, \{k, e\}\}$. The attack degree of \mathcal{Q}''

is $\max(1, 2, 3) = 3$ for k, e, h respectively. The proponent number is $\rho(\mathcal{H}, \mathcal{Q}'') = 2$. The upper-bound is:

$$\delta(\mathcal{H}, \mathcal{Q}'') \leq 2 \times \left(\frac{3^2 - 1}{3 - 1} - 1 \right) = 6$$

The real dispute complexity which corresponds to the shortest dialogue $(s(e), c(\{k\}), s(k), r(\{b\}, 1), s(k))$ is equal in this case to $\delta(\mathcal{H}, \mathcal{Q}'') = 5 < 6$.

It is clear that the bounds proposed for the dispute complexity of universal and non-universal acceptance are estimated in terms of the proponent and the block numbers. Unfortunately, these numbers are not given as inputs and they should be computed. It is obvious that computing such numbers is hard [19]⁸. Fortunately if one can estimate the cardinality of the minimum hitting set one can easily estimate the proponent or the block number. This can be achieved by using the results from [19] on the independence number in hypergraphs which is the complement of the hitting set number (also known as the transversal number).

5 Discussion and Conclusion

In this paper we have provided a dialectical proof theory for universal acceptance in coherent logic-based argumentation frameworks. We proved its finiteness, soundness, completeness, consistency and studied its dispute complexity.

It is important to point out that this dialectical proof theory can also be used in abstract settings like the one in [2]. In this work, Dung's abstract framework [16] is used in decision-support systems where arguments support different options (or decisions) and the final decision is computed using Dung's semantics. The author have introduced the concept of universal acceptance for a given option and shown that skeptical and universal are different. In fact, the distinction is important and practical since in certain decision making situations we may opt for an option that is supported by different arguments from different extensions but not supported by skeptical arguments (as there may be none). Our dialectical proof theory can offer an interesting feature in such settings, **explanation**. Dialectical proof theories in general provide, as argued by [25], explanation as to why a given output (option, conclusion, argument, etc.) is believed to be accepted. So, alongside to its capability of computing the accepted outputs, it can explain *why* and *how* the output are accepted. Aside from abstract settings, this intrinsic quality of explanation in our dialectical proof theory can lend itself to other domains such as (deductive) databases systems, more precisely in consistent query answering over inconsistent knowledge bases [4, 22, 9]. In fact it has been proven in [14] that the universal acceptance is equivalent to the well-known consistent query answering semantics [4], so our dialectical theory can be used in explaining why certain queries are entailed or not under the consistent query answering semantics which would have a great impact on the usability of such systems (as stipulated by [24]).

As a final remark, it seems that the concept of “arguments supporting a query” in our dialectical proof theory can somewhat be related to bipolar argumentation frameworks [13] that extends Dung's framework by a support relation *between arguments*. This will be the subject of our future work. Another future work is to look at the behavior of this dialectical proof theory on non-coherent or infinite argumentation frameworks.

⁸ In fact they are equivalent to finding a minimum proponent or block for the query, which would solve the problem in the first place.

REFERENCES

- [1] Leila Amgoud, 'Postulates for logic-based argumentation systems', *International Journal of Approximate Reasoning*, **55**(9), 2028–2048, (2014).
- [2] Leila Amgoud, Yannis Dimopoulos, and Pavlos Moraitis, 'Making decisions through preference-based argumentation', *Proceedings of the Eleventh International Conference on Principles of Knowledge Representation and Reasoning KR'08*, **8**, 963–970, (2008).
- [3] Leila Amgoud and Florence Dupin de Saint-Cyr, 'An axiomatic approach for persuasion dialogs', in *IEEE 25th International Conference on Tools with Artificial Intelligence (ICTAI'13)*, pp. 618–625. IEEE, (2013).
- [4] Marcelo Arenas, Leopoldo Bertossi, and Jan Chomicki, 'Consistent query answers in inconsistent databases', in *Proceedings of the eighteenth ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*, pp. 68–79. ACM, (1999).
- [5] Franz Baader, Sebastian Brandt, and Carsten Lutz, 'Pushing the envelope', in *Proceedings of International Joint Conference on Artificial Intelligence (IJCAI'05)*, (2005).
- [6] Jean-François Baget, Michel Leclère, Marie-Laure Mugnier, and Eric Salvat, 'On rules with existential variables: Walking the decidability line', *Artificial Intelligence*, **175**(9-10), 1620–1654, (2011).
- [7] Philippe Besnard and Anthony Hunter, *Elements of Argumentation*, MIT Press, 2008.
- [8] Philippe Besnard and Anthony Hunter, 'Constructing argument graphs with deductive arguments: a tutorial', *Argument & Computation*, **5**(1), 5–30, (2014).
- [9] Meghyn Bienvenu, 'On the complexity of consistent query answering in the presence of simple ontologies', in *Proceedings of AAAI'12 Conference on Artificial Intelligence*, (2012).
- [10] Andrea Cali, Georg Gottlob, and Thomas Lukasiewicz, 'A general datalog-based framework for tractable query answering over ontologies', *Web Semantics: Science, Services and Agents on the World Wide Web*, **14**, 57–83, (2012).
- [11] Diego Calvanese, Giuseppe De Giacomo, Domenico Lembo, Maurizio Lenzerini, and Riccardo Rosati, 'Tractable reasoning and efficient query answering in description logics: The dl-lite family', *J. Autom. Reasoning*, **39**(3), 385–429, (2007).
- [12] Claudette Cayrol, Sylvie Doutre, and Jérôme Mengin, 'On decision problems related to the preferred semantics for argumentation frameworks', *Journal of logic and computation*, **13**(3), 377–403, (2003).
- [13] Claudette Cayrol and Marie-Christine Lagasque-Schiev, 'On the acceptability of arguments in bipolar argumentation frameworks', in *Symbolic and Quantitative Approaches to Reasoning with Uncertainty (ECSQARU'05)*, 378–389, Springer, (2005).
- [14] Madalina Croitoru and Srdjan Vesic, 'What can argumentation do for inconsistent ontology query answering?', in *Scalable Uncertainty Management (SUM'13)*, 15–29, Springer, (2013).
- [15] Sylvie Doutre and Jérôme Mengin, 'On sceptical versus credulous acceptance for abstract argument systems', in *Logics in Artificial Intelligence*, 462–473, Springer, (2004).
- [16] Phan Minh Dung, 'On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games', *Artificial intelligence*, **77**(2), 321–357, (1995).
- [17] Paul E Dunne and Trevor JM Bench-Capon, 'Two party immediate response disputes: properties and efficiency', *Artificial Intelligence*, **149**(2), 221–250, (2003).
- [18] Morten Elvang-Gøransson, Paul J Krause, and John Fox, 'Acceptability of arguments as logical uncertainty', in *Symbolic and Quantitative Approaches to Reasoning and Uncertainty (ECSQARU'93)*, 85–90, Springer, (1993).
- [19] Alexander Eustis, *Hypergraph Independence Numbers*, Ph.D. dissertation, University of California San Diego, 2013.
- [20] Hadassa Jakobovits and Dirk Vermeir, 'Dialectic semantics for argumentation frameworks', in *Proceedings of the 7th International Conference on Artificial intelligence and Law*, pp. 53–62. ACM, (1999).
- [21] Mark W Johnson, Peter McBurney, and Simon Parsons, 'When are two protocols the same?', in *Communication in Multiagent Systems*, 253–268, Springer, (2003).
- [22] Domenico Lembo, Maurizio Lenzerini, Riccardo Rosati, Marco Ruzzi, and Domenico Fabio Savo, 'Inconsistency-tolerant semantics for description logics', in *Proceedings of the Fourth International Conference on Web Reasoning and Rule Systems, RR'10*, pp. 103–117, Berlin, Heidelberg, (2010). Springer-Verlag.
- [23] Kuno Lorenz, 'Basic objectives of dialogue logic in historical perspective', *Synthese*, **127**(1-2), 255–263, (2001).
- [24] Deborah L McGuinness and Peter F Patel-Schneider, 'Usability issues in knowledge representation systems', in *Proceedings of AAAI'98*, pp. 608–614, (1998).
- [25] Sanjay Modgil and Martin Caminada, 'Proof theories and algorithms for abstract argumentation frameworks', in *Argumentation in artificial intelligence*, eds., Guillermo Simari and Iyad Rahwan, 105–129, Springer, (2009).
- [26] Marie-Laure Mugnier, Marie-Christine Rousset, and Federico Ulliana, 'Ontology-mediated queries for nosql databases', in *Proceedings of AAAI'16 Conference on Artificial Intelligence*, (2016).
- [27] Antonella Poggi, Domenico Lembo, Diego Calvanese, Giuseppe De Giacomo, Maurizio Lenzerini, and Riccardo Rosati, 'Linking data to ontologies', in *Journal on data semantics X*, 133–173, Springer, (2008).
- [28] Henry Prakken, 'Relating protocols for dynamic dispute with logics for defeasible argumentation', *Synthese*, **127**(1-2), 187–219, (2001).
- [29] Phan Minh Thang, Phan Minh Dung, and Nguyen Duy Hung, 'Towards a common framework for dialectical proof procedures in abstract argumentation', *Journal of Logic and Computation*, **19**(6), 1071–1109, (2009).
- [30] Gerard AW Vreeswijk and Henry Prakken, 'Credulous and sceptical argument games for preferred semantics', in *Logics in Artificial Intelligence*, 239–253, Springer, (2000).