Proceedings, Fifteenth International Conference on
Principles of Knowledge Representation and Reasoning (KR 2016)

Argumentative Approaches to Reasoning with Maximal Consistency

Ofer Arieli
School of Computer Science
The Academic College of Tel-Aviv, Israel *

Christian Straßer
Institute of Philosophy II,
Ruhr University Bochum, Germany †

Abstract

Reasoning with the maximally consistent subsets (MCS) of the premises is a well-known approach for handling contradictory information. We introduce two argumentation-based methods for doing so: a declarative approach that is related to Dung-style semantics for abstract argumentation, and a computational approach that is based on extensions of Gentzen-type proofs systems. This brings about a new perspective on reasoning with MCS which shows a strong link between the latter and argumentation systems, and which can be extended to related formalisms. A by-product of this is the introduction of a dynamic proof system for classical logic and rebuttal attacks, which is sound and complete with respect to Dung's stable semantics for the associated argumentation framework.

Introduction

A well-established method for handling inconsistencies in a given set of premises is to consider its maximally consistent subsets (MCS). Following the influential work of Rescher and Manor (1970) this approach has gained a considerable popularity and was applied in many AI-related areas. The goal of this work is to develop systematic methods of representing and reasoning with MCS by means of argumentation-based systems.

The relation between MCS-based reasoning and argumentation theory has been already identified in the literature (see, e.g., (Amgoud and Besnard 2013)). The contribution of this paper is the provision of a uniform argumentative approach for representing MCS-based formalisms and reasoning with them by proof theoretical tools. For this, we incorporate the sequent-based argumentation frameworks described in (Arieli 2013; Arieli and Straßer 2015). This allows us to introduce two complementary methods for reasoning with MCS: a declarative method, based on Dung's semantics for argumentation frameworks (Dung 1995), and a computational method, generalizing standard proof systems, where derivations have a non-monotonic flavor. We show that these complementary disciplines are equivalent, and that they may be carried on to some generalized argumentation-based contexts for reasoning with maximal consistency.

*Supported by the Israel Science Foundation (grant 817/15).
†Supported by the Alexander von Humboldt Foundation.

Sequent-Based Argumentation

According to Dung (1995), abstract argumentation frameworks may be viewed as directed graphs as follows:

Definition 1 An *(abstract) argumentation framework* is a pair $\mathcal{AF} = \langle Args, Attack \rangle$, where $Args$ is a denumerable set of elements, called *arguments*, and $Attack$ is a relation on $Args \times Args$, whose instances are called *attacks*.

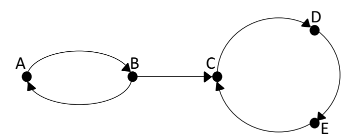

Figure 1: An argumentation framework with five arguments and six attacks.

When it comes to applications, it is often useful to provide a specific account of the structure of arguments and the concrete nature of argumentative attacks. For this we follow the sequent-based approach introduced in (Arieli 2013; Arieli and Straßer 2015), briefly described below.[1]

In what follows we restrict ourselves to classical logic $\mathsf{CL} = \langle \mathcal{L}, \vdash_{\mathsf{CL}} \rangle$, where $\mathcal{L}$ is a standard propositional language with the basic connectives $\neg, \wedge, \vee, \supset, \leftrightarrow$, and $\vdash_{\mathsf{CL}}$ is the standard consequence relation of this logic. Atomic formulas in $\mathcal{L}$ are denoted by p, q, compound formulas are denoted by ψ, ϕ, sets of formulas are denoted by $\mathcal{S}, \mathcal{T}$, and *finite* sets of formulas are denoted by Γ, Δ. Arguments in this setting are considered next:

Definition 2 Let $\mathcal{S}$ be a set of $\mathcal{L}$-formulas.

- A *sequent* is an expression $\Gamma \Rightarrow \Delta$, where Γ, Δ are finite sets of formulas, and $\Rightarrow$ is a reserved symbol (not in $\mathcal{L}$).
- An *argument* is a sequent $\Gamma \Rightarrow \psi$ where $\Gamma \vdash_{\mathsf{CL}} \psi$. We shall denote $\mathsf{Prem}(\Gamma \Rightarrow \psi) = \Gamma$ and $\mathsf{Con}(\Gamma \Rightarrow \psi) = \psi$.
- An $\mathcal{S}$-*argument* is an sequent $\Gamma \Rightarrow \psi$ in which $\Gamma \subseteq \mathcal{S}$. The set of all the $\mathcal{S}$-arguments is denoted $\mathsf{Arg}(\mathcal{S})$.

Note 1 Unlike other definitions of arguments in deductive systems (e.g., (Besnard and Hunter 2001)), here the support set of an argument need not be consistent nor minimal.

Note 2 $\Gamma \Rightarrow \psi \in \mathsf{Arg}(\mathcal{S})$ *for a finite* $\Gamma \subseteq \mathcal{S}$, *iff* $\mathcal{S} \vdash_{\mathsf{CL}} \psi$.

[1]We refer to these papers for some justifications of our choice.

We shall use Gentzen's sequent calculus LK (Gentzen 1934) for constructing arguments from simpler arguments (see Figure 2). This is done by *inference rules* of the form:

$$\frac{\Gamma_1 \Rightarrow \Delta_1 \ldots \ \Gamma_n \Rightarrow \Delta_n}{\Gamma \Rightarrow \Delta}. \tag{1}$$

In what follows we shall say that the sequents $\Gamma_i \Rightarrow \Delta_i$ $(i = 1, \ldots, n)$ are the *conditions* (or the *prerequisites*) of the rule in (1), and that $\Gamma \Rightarrow \Delta$ is its *conclusion*.[2]

Axioms: $\psi \Rightarrow \psi$

Structural Rules:

Weakening: $\dfrac{\Gamma \Rightarrow \Delta}{\Gamma, \Gamma' \Rightarrow \Delta, \Delta'}$

Cut: $\dfrac{\Gamma_1 \Rightarrow \Delta_1, \psi \qquad \Gamma_2, \psi \Rightarrow \Delta_2}{\Gamma_1, \Gamma_2 \Rightarrow \Delta_1, \Delta_2}$

Logical Rules:

$[\wedge\Rightarrow]$ $\dfrac{\Gamma, \psi, \varphi \Rightarrow \Delta}{\Gamma, \psi \wedge \varphi \Rightarrow \Delta}$ $\qquad$ $[\Rightarrow\wedge]$ $\dfrac{\Gamma \Rightarrow \Delta, \psi \quad \Gamma \Rightarrow \Delta, \varphi}{\Gamma \Rightarrow \Delta, \psi \wedge \varphi}$

$[\vee\Rightarrow]$ $\dfrac{\Gamma, \psi \Rightarrow \Delta \quad \Gamma, \varphi \Rightarrow \Delta}{\Gamma, \psi \vee \varphi \Rightarrow \Delta}$ $\qquad$ $[\Rightarrow\vee]$ $\dfrac{\Gamma \Rightarrow \Delta, \psi, \varphi}{\Gamma \Rightarrow \Delta, \psi \vee \varphi}$

$[\supset\Rightarrow]$ $\dfrac{\Gamma \Rightarrow \psi, \Delta \quad \Gamma, \varphi \Rightarrow \Delta}{\Gamma, \psi \supset \varphi \Rightarrow \Delta}$ $\qquad$ $[\Rightarrow\supset]$ $\dfrac{\Gamma, \psi \Rightarrow \varphi, \Delta}{\Gamma \Rightarrow \psi \supset \varphi, \Delta}$

$[\neg\Rightarrow]$ $\dfrac{\Gamma \Rightarrow \Delta, \psi}{\Gamma, \neg\psi \Rightarrow \Delta}$ $\qquad$ $[\Rightarrow\neg]$ $\dfrac{\Gamma, \psi \Rightarrow \Delta}{\Gamma \Rightarrow \Delta, \neg\psi}$

Figure 2: The proof system LK

Attack rules allow for the elimination of sequents. We shall denote by $\Gamma \not\Rightarrow \psi$ the elimination of $\Gamma \Rightarrow \psi$. Alternatively, $\overline{s}$ denotes the elimination of s. Now, a *sequent elimination rule* (or an *attack rule*) resembles an inference rule, except that its conclusion is a discharging of one condition:

$$\frac{\Gamma_1 \Rightarrow \Delta_1, \ldots, \Gamma_n \Rightarrow \Delta_n}{\Gamma_n \not\Rightarrow \Delta_n}. \tag{2}$$

The prerequisites of attack rules usually consist of three ingredients. The first sequent in the rule's prerequisites is the "attacking" sequent, the last sequent is the "attacked" sequent, and the other prerequisites are the conditions for the attack. In this view, conclusions of sequent elimination rules are the eliminations of the attacked arguments.

The elimination rule that we shall use is the following:

$$\text{Undercut:} \quad \frac{\Gamma_1 \Rightarrow \psi_1 \quad \Rightarrow \psi_1 \leftrightarrow \neg\bigwedge\Gamma'_2 \quad \Gamma_2, \Gamma'_2 \Rightarrow \psi_2}{\Gamma_2, \Gamma'_2 \not\Rightarrow \psi_2}$$

This rule (sometimes called Ucut for short) is a sequent-based variation of a rule with a similar name that has been considered in the literature of logical argumentation frameworks (see, e.g., (Pollock 1992; Besnard and Hunter 2001)). We refer to (Arieli and Straßer 2015) for a list of further sequent elimination rules that are useful in other contexts.

[2]As usual, axioms are treated as inference rules without conditions, i.e., they are rules of the form $\frac{}{\Gamma \Rightarrow \Delta}$.

Definition 3 Let $\mathcal{S}$ be a set of formulas and let θ be a substitution.

- An inference rule of the form of (1) is $\mathsf{Arg}(\mathcal{S})$-*applicable* if for every $1 \leq i \leq n$, $\theta(\Gamma_i) \Rightarrow \theta(\Delta_i)$ is LK-provable.
- An elimination rule of the form of (2) is $\mathsf{Arg}(\mathcal{S})$-*applicable* if $\theta(\Gamma_1) \Rightarrow \theta(\Delta_1)$ and $\theta(\Gamma_n) \Rightarrow \theta(\Delta_n)$ are in $\mathsf{Arg}(\mathcal{S})$ and for every $1 < i < n$, $\theta(\Gamma_i) \Rightarrow \theta(\Delta_i)$ is LK-provable.

In the second case above we say that $\theta(\Gamma_1) \Rightarrow \theta(\Delta_1)$ *attacks* $\theta(\Gamma_n) \Rightarrow \theta(\Delta_n)$. Note that the attacker and the attacked sequents must be elements of $\mathsf{Arg}(\mathcal{S})$.[3]

Sequent-based argumentation frameworks for classical logic and Undercut are now defined as follows:

Definition 4 The *sequent-based argumentation framework* for $\mathcal{S}$ is the framework $\mathcal{AF}(\mathcal{S}) = \langle \mathsf{Arg}(\mathcal{S}), \mathit{Attack} \rangle$ where $(s_1, s_2) \in \mathit{Attack}$ iff s_1 Undercut-attacks s_2.

Reasoning with MCS

To provide argumentative approaches for reasoning with inconsistent premises by their maximally consistent subsets we need to capture the following entailments:

Definition 5 Let $\mathcal{S}$ be a set of formulas. We denote by $\mathrm{Cn}(\mathcal{S})$ the transitive closure of $\mathcal{S}$ with respect to classical logic and by $\mathsf{MCS}(\mathcal{S})$ the set of all the maximally consistent subsets of $\mathcal{S}$. We denote:

- $\mathcal{S} \mid\!\sim_{\mathsf{mcs}} \psi$ iff $\psi \in \mathrm{Cn}(\bigcap \mathsf{MCS}(\mathcal{S}))$.
- $\mathcal{S} \mid\!\sim_{\cup\mathsf{mcs}} \psi$ iff $\psi \in \bigcup_{\mathcal{T} \in \mathsf{MCS}(\mathcal{S})} \mathrm{Cn}(\mathcal{T})$.

Approach I: Using Dung-Style Semantics

Our first approach of using sequent-based argumentation for computing the entailments of Definition 5 is based on Dung's semantics for abstract argumentation frameworks. Given a framework $\mathcal{AF}$ (Definition 1), a key issue in its understanding is the question what combinations of arguments (called *extensions*) can collectively be accepted from $\mathcal{AF}$. According to Dung (1995), this is determined as follows:

Definition 6 Let $\mathcal{AF} = \langle \mathit{Args}, \mathit{Attack} \rangle$ be an argumentation framework, and let $\mathcal{E} \subseteq \mathit{Args}$.

- $\mathcal{E}$ *attacks* an argument A if there is an argument $B \in \mathcal{E}$ that attacks A (i.e., $(B, A) \in \mathit{Attack}$). The set of arguments that are attacked by $\mathcal{E}$ is denoted $\mathcal{E}^+$.
- $\mathcal{E}$ *defends* A if $\mathcal{E}$ attacks every argument that attacks A.
- $\mathcal{E}$ is called *conflict-free* if it does not attack any of its elements (i.e., $\mathcal{E}^+ \cap \mathcal{E} = \emptyset$), $\mathcal{E}$ is *admissible* if it is conflict-free and defends all of its elements, and $\mathcal{E}$ is *complete* if it is admissible and contains all the arguments that it defends.
- The minimal complete subset of Args is called the *grounded extension* of $\mathcal{AF}$, and a maximal complete subset of Args is called a *preferred extension* of $\mathcal{AF}$. A complete set $\mathcal{E}$ is called a *stable extension* of $\mathcal{AF}$ if $\mathcal{E} \cup \mathcal{E}^+ = \mathit{Args}$.
- We write $\mathsf{Adm}(\mathcal{AF})$ [respectively: $\mathsf{Cmp}(\mathcal{AF})$, $\mathsf{Prf}(\mathcal{AF})$, $\mathsf{Stb}(\mathcal{AF})$] for the set of all the admissible [respectively: complete, preferred, stable] extensions of $\mathcal{AF}$ and $\mathsf{Grd}(\mathcal{AF})$ for the unique grounded extension of $\mathcal{AF}$.

[3]This requirement prevents situations in which, e.g., $\neg p \Rightarrow \neg p$ attacks $p \Rightarrow p$ (by Undercut), although $\mathcal{S} = \{p\}$.

Example 1 In Figure 1, $\emptyset$, $\{A\}$, $\{B\}$ and $\{B, D\}$ are admissible, $\emptyset$, $\{A\}$, and $\{B, D\}$ are complete, $\emptyset$ is grounded, $\{A\}$ and $\{B, D\}$ are preferred, and $\{B, D\}$ is stable.

Definition 7 *Let $\mathcal{AF}(\mathcal{S}) = \langle \mathsf{Arg}(\mathcal{S}), Attack\rangle$ be a sequent-based argumentation framework. We denote:*

- $\mathcal{S} \mid\!\sim_{\mathsf{gr}} \psi$ *if there is* $s \in \mathsf{Grd}(\mathcal{AF}(\mathcal{S}))$ *and* $\mathsf{Con}(s) = \psi$.
- $\mathcal{S} \mid\!\sim_{\cap\mathsf{prf}} \psi$ *if there is* $s \in \bigcap \mathsf{Prf}(\mathcal{AF}(\mathcal{S}))$ *and* $\mathsf{Con}(s) = \psi$.
- $\mathcal{S} \mid\!\sim_{\cap\mathsf{stb}} \psi$ *if there is* $s \in \bigcap \mathsf{Stb}(\mathcal{AF}(\mathcal{S}))$ *and* $\mathsf{Con}(s) = \psi$.
- $\mathcal{S} \mid\!\sim_{\cup\mathsf{prf}} \psi$ *if there is* $s \in \bigcup \mathsf{Prf}(\mathcal{AF}(\mathcal{S}))$ *and* $\mathsf{Con}(s) = \psi$.
- $\mathcal{S} \mid\!\sim_{\cup\mathsf{stb}} \psi$ *if there is* $s \in \bigcup \mathsf{Stb}(\mathcal{AF}(\mathcal{S}))$ *and* $\mathsf{Con}(s) = \psi$.

The relation between Dung's semantics for sequent-based frameworks and MCS-reasoning is shown next:[4]

Proposition 1 *Let $\mathcal{S}$ be a set of formulas and ψ a formula.*

1. $\mathcal{S} \mid\!\sim_{\mathsf{gr}} \psi$ *iff* $\mathcal{S} \mid\!\sim_{\cap\mathsf{prf}} \psi$ *iff* $\mathcal{S} \mid\!\sim_{\cap\mathsf{stb}} \psi$ *iff* $\mathcal{S} \mid\!\sim_{\mathsf{mcs}} \psi$.
2. $\mathcal{S} \mid\!\sim_{\cup\mathsf{prf}} \psi$ *iff* $\mathcal{S} \mid\!\sim_{\cup\mathsf{stb}} \psi$ *iff* $\mathcal{S} \mid\!\sim_{\cup\mathsf{mcs}} \psi$.

Corollary 1 *Let $\mathcal{S}$ be a set of formulas and ψ a formula.*

1. *If* $\mathcal{S} \mid\!\sim_{\cap\mathsf{prf}} \psi$ *and* $\psi \vdash_{\mathsf{CL}} \varphi$ *then* $\mathcal{S} \mid\!\sim_{\cap\mathsf{prf}} \varphi$.
2. *If* $\mathcal{S} \mid\!\sim_{\cap\mathsf{prf}} \psi$ *then* $\mathcal{S} \not\mid\!\sim_{\cap\mathsf{prf}} \neg\psi$.
3. *If* $\mathcal{S} \mid\!\sim_{\cap\mathsf{prf}} \psi$ *and* $\mathcal{S} \mid\!\sim_{\cap\mathsf{prf}} \varphi$ *then* $\mathcal{S} \mid\!\sim_{\cap\mathsf{prf}} \psi \wedge \varphi$.
4. *If* $\mathcal{S} \mid\!\sim_{\cap\mathsf{prf}} \psi$ *or* $\mathcal{S} \mid\!\sim_{\cap\mathsf{prf}} \varphi$ *then* $\mathcal{S} \mid\!\sim_{\cap\mathsf{prf}} \psi \vee \varphi$.
5. *If* $\mathcal{S} \mid\!\sim_{\cap\mathsf{prf}} \neg\psi$ *or* $\mathcal{S} \mid\!\sim_{\cap\mathsf{prf}} \varphi$ *then* $\mathcal{S} \mid\!\sim_{\cap\mathsf{prf}} \psi \supset \varphi$.
6. $\mathcal{S} \mid\!\sim_{\cap\mathsf{prf}} \psi$ *iff* $\{\phi \mid \mathcal{S} \mid\!\sim_{\cap\mathsf{prf}} \phi\} \mid\!\sim_{\cap\mathsf{prf}} \psi$.
7. *Items 1–6 above hold also for* $\mid\!\sim_{\cap\mathsf{stb}}$ *and* $\mid\!\sim_{\mathsf{gr}}$.

Approach II: Using Dynamic Derivations

The second argumentation-based approach for reasoning with MCS is a proof-theoretical one.

Definition 8 A (proof) *tuple* is a quadruple $\langle i, s, \mathsf{J}, \mathsf{A}\rangle$, where i (the tuple's index) is a number, s (the tuple's sequent) is either a sequent or an eliminated sequent, J (the tuple's justification) is a string, and A (the attacker of the tuple's sequent) is the empty set or a singleton (of a sequent).

Proofs in our case are sequences of tuples obtained by applications of rules in LK and applications of Undercut.

Definition 9 A *simple* (dynamic) *derivation* (for a set $\mathcal{S}$ of formulas) is a finite sequence $\mathcal{D} = \langle T_1, \dots T_m\rangle$ of proof tuples, where each $T_i \in \mathcal{D}$ is of one of the following forms:

- An *introducing tuple for* $\Gamma \Rightarrow \Delta$:
$$T_i = \langle i, \Gamma \Rightarrow \Delta, \text{“}\mathcal{R}; i_1, \dots, i_n\text{”}, \emptyset\rangle,$$
where $\mathcal{R} \in LK$ is $\mathsf{Arg}(\mathcal{S})$-applicable for a substitution θ, the numbers $i_1, \dots, i_n < i$ are indexes of introducing tuples for the θ-substitutions of the conditions of $\mathcal{R}$, and $\Gamma \Rightarrow \Delta$ is the θ-substitution of the conclusion of $\mathcal{R}$.
- An *eliminating tuple of* $\Gamma_2 \Rightarrow \psi_2$:
$$T_i = \langle i, \Gamma_2 \not\Rightarrow \psi_2, \text{“Ucut}; i_1, j, i_2\text{”}, \Gamma_1 \Rightarrow \psi_1\rangle,$$
where $\Gamma_1 \Rightarrow \psi_1$ Ucut-attacks $\Gamma_2 \Rightarrow \psi_2$, and there are introducing tuples for the sequents $\Gamma_1 \Rightarrow \psi_1$, $\Gamma_2 \Rightarrow \psi_2$, and $\Rightarrow \psi_1 \leftrightarrow \neg\bigwedge\Gamma_2'$ (for $\Gamma_2' \subseteq \Gamma_2$), whose indexes are, respectively, i_1, i_2, and j (all of which are less than i).

Given a simple derivation $\mathcal{D}$, $\mathsf{Top}(\mathcal{D})$ is the tuple with the highest index in $\mathcal{D}$ and $\mathsf{Tail}(\mathcal{D})$ is the simple derivation $\mathcal{D}$ without $\mathsf{Top}(\mathcal{D})$. To indicate that the validity of a derived sequent (in a simple derivation) is in question due to attacks on it, we need the following evaluation process.

Definition 10 Given a simple derivation $\mathcal{D}$, the algorithm in Figure 3 computes three sets: $\mathsf{Elim}(\mathcal{D})$ – eliminated sequents whose attacker is not already eliminated, $\mathsf{Attack}(\mathcal{D})$ – the sequents that attack a sequent in $\mathsf{Elim}(\mathcal{D})$, $\mathsf{Accept}(\mathcal{D})$ – the derived sequents in $\mathcal{D}$ that are not in $\mathsf{Elim}(\mathcal{D})$.

```
function Evaluate(D)   /* D – a simple derivation */
Attack := Elim := Derived := ∅;
while (D is not empty) do {
   if (Top(D) = ⟨i, s, J, ∅⟩) then
      Derived := Derived ∪ {s};
   if (Top(D) = ⟨i, s̄, J, r⟩) then
      if (r ∉ Elim) then
         Elim := Elim ∪ {s};   Attack := Attack ∪ {r};
   D := Tail(D); }
Accept := Derived − Elim;
return (Attack, Elim, Accept)
```

Figure 3: Evaluation of a simple derivation

Definition 11 A simple derivation $\mathcal{D}$ is *coherent*, if there is no sequent that eliminates another sequent, and later is eliminated itself, that is: $\mathsf{Attack}(\mathcal{D}) \cap \mathsf{Elim}(\mathcal{D}) = \emptyset$.

Now we are ready to define derivations in our framework.

Definition 12 Let $\mathcal{S}$ be a set of formulas. A *(dynamic) derivation* (based on $\mathcal{S}$) is a simple derivation $\mathcal{D}$ of one of the following forms:

a) $\mathcal{D} = \langle T\rangle$, where $T = \langle 1, s, \mathsf{J}, \emptyset\rangle$ is a proof tuple.

b) $\mathcal{D}$ extends a dynamic derivation by a sequence $\langle T_1, \dots, T_n\rangle$ of introducing tuples (of the form $\langle i, s, \mathsf{J}, \emptyset\rangle$), whose sequents (the s's) are not in $\mathsf{Elim}(\mathcal{D})$.

c) $\mathcal{D}$ is a *coherent* extension of a dynamic derivation by a sequence $\langle T_1, \dots, T_n\rangle$ of eliminating tuples (of the form $\langle i, \overline{s}, \mathsf{J}, r\rangle$), whose attacking sequents (the r's) are not Ucut-attacked by a sequent in $\mathsf{Accept}(\mathcal{D}) \cap \mathsf{Arg}(\mathcal{S})$ and the attack is based on conditions (justifications) in $\mathcal{D}$.[5]

One may think of a dynamic derivation as a proof that progresses over derivation steps. At each step the derivation is extended by a 'block' of introducing or eliminating tuples, and the status of the derived sequents is updated accordingly. Derived sequents may be eliminated by new proof tuples, and eliminated sequents may be 'restored' if their attacking tuples are counter-attacked by a new eliminating tuple. The outcomes of a dynamic derivation are defined next.

[4]Due to lack of space proofs are omitted.

[5]These are sound attacks: by coherence neither of the attacking sequents of the additional elimination tuples is in $\mathsf{Elim}(\mathcal{D})$, and by the other condition they are not attacked by an accepted sequent.

Definition 13 A sequent s is *finally* (or *safely*) *derived* in a derivation $\mathcal{D}$ (for $\mathcal{S}$) if $s \in \mathsf{Accept}(\mathcal{D})$ and $\mathcal{D}$ cannot be extended to a derivation $\mathcal{D}'$ (for $\mathcal{S}$) such that $s \in \mathsf{Elim}(\mathcal{D}')$.

Proposition 2 *If s is finally derived in $\mathcal{D}$ then it is finally derived in any extension of $\mathcal{D}$.*

The induced entailment is now defined as follows:

Definition 14 $\mathcal{S} \mid\!\sim \psi$ if there is a dynamic derivation for $\mathcal{S}$, in which $\Gamma \Rightarrow \psi$ is finally derived for some finite $\Gamma \subseteq \mathcal{S}$.

Example 2 A dynamic derivation for $\mathcal{S} = \{p, \neg p, q\}$:

1	$p \Rightarrow p$	Axiom
2	$\Rightarrow p, \neg p$	$[\Rightarrow\neg]$, 1
3	$\Rightarrow p \vee \neg p$	$[\Rightarrow\vee]$, 2
4	$\Rightarrow p \vee \neg p \leftrightarrow \neg(p \wedge \neg p)$	...
5	$q \Rightarrow q$	Axiom

Note that $q \Rightarrow q$ is finally derived. Any attempt to attack this sequent by sequents of the form $p, \neg p \Rightarrow \psi$ or $p, \neg p, q \Rightarrow \psi$, where ψ is logically equivalent to $\neg q$, is counter attacked by $\Rightarrow p \vee \neg p$ (introduced in Tuple 3), using the justification in Tuples 4 (These are the only attacks on $q \Rightarrow q$ by attackers in $\mathsf{Arg}(\mathcal{S})$). Thus, the above derivation cannot be extended to a derivation in which $q \Rightarrow q$ is attacked, and so $\mathcal{S} \mid\!\sim q$.

Note, however, that the above derivation *can* be extended by the following tuples, yielding an elimination of $p \Rightarrow p$:

6	$\neg p \Rightarrow \neg p$	Axiom	
7	$\Rightarrow \neg p \leftrightarrow \neg p$	...	
8	$p \not\Rightarrow p$	Ucut, 6, 7, 1	$\neg p \Rightarrow \neg p$

In turn, this derivation can be extended as follows, yielding an attack on $\neg p \Rightarrow \neg p$:

9	$\Rightarrow p \leftrightarrow \neg\neg p$	...	
10	$\neg p \not\Rightarrow \neg p$	Ucut, 1, 9, 6	$p \Rightarrow p$

and now $p \Rightarrow p$ is not eliminated anymore. Still, $p \Rightarrow p$ can be re-attacked by $\neg p \Rightarrow \neg p$, and so forth. As a consequence, we have that $\mathcal{S} \not\mid\!\sim p$ and $\mathcal{S} \not\mid\!\sim \neg p$.

Note 3 The present setting of dynamic derivations may be viewed as an improvement of a similar setting, introduced in (Arieli and Straßer 2014). The main difference is that here the way sequents can be introduced in a proof depends on the already introduced elimination sequents. This in particular guarantees the correspondence, shown in Theorem 1, between entailments induced by dynamic derivations and entailments induced by some Dung-style semantics of the associated sequent-based frameworks. This correspondence does not hold for the formalism in (Arieli and Straßer 2014).

Now we can show how dynamic derivations allow for reasoning with maximally consistent sets of premises.

Proposition 3 *Let $\mathcal{S}$ be a finite set of formulas and ψ a formula. Then $\mathcal{S} \mid\!\sim_{\mathsf{mcs}} \psi$ iff $\mathcal{S} \mid\!\sim \psi$.*

Example 3 $\mathcal{S} = \{p, \neg p, q\}$, $\mathsf{MCS}(\mathcal{S}) = \{\{p, q\}, \{\neg p, q\}\}$, and so $\bigcap \mathsf{MCS}(\mathcal{S}) = \{q\}$. By Proposition 3, $\mathcal{S} \mid\!\sim q$ while $\mathcal{S} \not\mid\!\sim p$ and $\mathcal{S} \not\mid\!\sim \neg p$, as indeed shown in Example 2.

By Propositions 1 and 3 we therefore have two complementary sequent-based argumentation methods for reasoning with maximal consistency:

Theorem 1 *For finite sets of premises $\mid\!\sim_{\mathsf{gr}}$, $\mid\!\sim_{\cap\mathsf{prf}}$, $\mid\!\sim_{\cap\mathsf{stb}}$ and $\mid\!\sim$ are the same, and all of them are equivalent to $\mid\!\sim_{\mathsf{mcs}}$.*

Reasoning with Consistent Subsets

The entailment relation in Definition 5 may be strengthened in a various ways. We conclude by considering one of them.

Definition 15 (Benferhat, Dubois, and Prade 1997) Let $\mathcal{S}$ be a set of formulas and ψ a formula. Then $\mathcal{S} \mid\mid\!\sim_{\mathsf{mcs}} \phi$ if:

1. It holds that $\mathcal{T} \vdash_{\mathsf{CL}} \phi$ for some consistent subset $\mathcal{T}$ of $\mathcal{S}$.
2. There is no consistent subset $\mathcal{T}$ of $\mathcal{S}$ such that $\mathcal{T} \vdash_{\mathsf{CL}} \neg\phi$.

Let us denote by $\mid\mid\!\sim_{\mathsf{gr}}$ the entailment that is defined like $\mid\!\sim_{\mathsf{gr}}$ (Definition 7), except that instead of Undercut the attack relations are Consistency Undercut and Defeating Rebuttal:

$$\text{Consistency Undercut:} \quad \frac{\Rightarrow \neg\bigwedge\Gamma_2' \quad \Gamma_2, \Gamma_2' \Rightarrow \psi_2}{\Gamma_2, \Gamma_2' \not\Rightarrow \psi_2}$$

$$\text{Defeating Rebuttal:} \quad \frac{\Gamma_1 \Rightarrow \psi_1 \quad \Rightarrow \psi_1 \supset \neg\psi_2 \quad \Gamma_2 \Rightarrow \psi_2}{\Gamma_2 \not\Rightarrow \psi_2}$$

Proposition 4 *Let $\mathcal{S}$ be a set of formulas and ψ a formula. Then $\mathcal{S} \mid\mid\!\sim_{\mathsf{mcs}} \psi$ iff $\mathcal{S} \mid\mid\!\sim_{\mathsf{gr}} \psi$.*

For the computational counterpart of this generalization, we denote by $\mid\mid\!\sim$ the consequence relation of our dynamic proof system (see Definition 14), using attacks by Consistency Undercut and Defeating Rebuttal (instead of Ucut).

Proposition 5 *Let $\mathcal{S}$ be a finite set of formulas and ψ a formula. Then $\mathcal{S} \mid\mid\!\sim_{\mathrm{mcs}} \psi$ iff $\mathcal{S} \mid\mid\!\sim \psi$.*

By Propositions 4 and 5 we have:

Theorem 2 *For finite sets of premises $\mid\mid\!\sim_{\mathsf{gr}}$ and $\mid\mid\!\sim$ are the same, and both of them are equivalent to $\mid\mid\!\sim_{\mathsf{mcs}}$.*

References

Amgoud, L., and Besnard, P. 2013. Logical limits of abstract argumentation frameworks. *Journal of Applied Non-Classical Logics* 23(3):229–267.

Arieli, O., and Straßer, C. 2014. Dynamic derivations for sequent-based logical argumentation. *Proc. COMMA'14*, 89–100. IOS Press.

Arieli, O., and Straßer, C. 2015. Sequent-based logical argumentation. *Argument and Computation* 6(1):73–99.

Arieli, O. 2013. A sequent-based representation of logical argumentation. *CLIMA'13*, LNCS 8143, 69–85. Springer.

Benferhat, S.; Dubois, D.; and Prade, H. 1997. Some syntactic approaches to the handling of inconsistent knowledge bases: A comparative study, part 1: The flat case. *Studia Logica* 58(1):17–45.

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

Dung, P. M. 1995. On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games. *J. Artif. Intell.* 77:321–357.

Gentzen, G. 1934. Investigations into logical deduction. (In German). English translation: 'The Collected Works of Gerhard Gentzen', edited by M. E. Szabo, North-Holland, 1969.

Pollock, J. 1992. How to reason defeasibly. *J. Artif. Intell.* 57(1):1–42.

Rescher, N., and Manor, R. 1970. On inference from inconsistent premises. *Theory and Decision* 1:179–217.