# Deductive Argumentation by Enhanced Sequent Calculi and Dynamic Derivations

Ofer Arieli[1]

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

Christian Straßer[2]

*Institute of Philosophy II*
*Ruhr University Bochum*
*Bochum, Germany*

**Abstract**

Logic-based approaches for analyzing and evaluating arguments have been largely studied in recent years, yielding a variety of formal methods for argumentation-based reasoning. The goal of this paper is to provide an abstract, proof theoretical investigation of logical argumentation, where arguments are represented by sequents, conflicts between arguments are represented by sequent elimination rules, and deductions are made by dynamic proof systems extending standard sequent calculi.

*Keywords:* logical argumentation, sequent calculi, dynamic derivations.

## 1 Introduction

Logical argumentation (sometimes called deductive argumentation) is a logic-based approach for formalizing debates, disagreements, and entailment relations for drawing conclusions from argumentation-based settings [8,17,18,20]. The basic entities in this context are called *arguments*. An argument is a pair of a finite set of formulas ($\Gamma$, the support set) and a formula ($\psi$, the conclusion), expressed in an arbitrary propositional language, such that the latter follows, according to some underlying logic, from the former. As indicated in [1] and [4], this gives rise to the association of arguments with Gentzen's notion of *sequents* [13], where an argument is expressed

by a sequent of the form $\Gamma \Rightarrow \psi$. Accordingly, logical argumentation boils down to the exposition of formalized methods for reasoning with these syntactical objects.

A first step towards a proof theoretical investigation of sequent-based logical argumentation is done in [3]. In this paper we revise and extend that work. In particular, we introduce a modified proof theoretical approach and show that this approach is better suited to argumentation-based reasoning than that of [3] (see also Note 5 below). For this, in the next section we recall some basic notions behind abstract argumentation in general and (sequent-based) logical argumentation in particular. Then, in Section 3 we consider a generic method of drawing conclusions from a given set of sequent-based arguments, which is tolerant to different logics, languages, and attack relations among the arguments. This is achieved by introducing the notions of *dynamic proofs*, which are intended for explicating actual reasoning in an argumentation framework. Unlike 'standard' proof methods, the idea here is that an argument can be challenged by a counter-argument, and so a certain sequent may be considered as not derived at a certain stage of the proof, even if it were considered derived in an earlier stage of the proof. In Section 4 we give some demonstrations showing that despite this non-monotonic nature of dynamic derivations one may still draw solid conclusions, which are faithful to the intended semantics of the logical argumentation framework at hand. This is formally proved in Section 5, where several properties of the underlying entailment relations are discussed, including some restricted forms of reflexivity and monotonicity. Finally, in Section 6 we give a brief outlook to future work.

## 2    Preliminaries

We start by reviewing the notion of sequent-based argumentation, as defined in [4]. First, we recall the more general notion of abstract argumentation frameworks.

### 2.1    Argumentation Frameworks and Their Semantics

Abstract argumentation frameworks are directed graphs, where the nodes represent (abstract) arguments and the arrows represent attacks between arguments, as defined next.

**Definition 2.1** An *(abstract) argumentation framework* [11] is a pair $\mathcal{AF} = \langle Args, Attack \rangle$, where $Args$ is a denumerable set of elements, called *arguments*, and $Attack$ is a binary relation on $Args$, whose instances are called *attacks*.

Given an argumentation framework, a key issue in its understanding is to determine what combinations of arguments (called *extensions*) can collectively be accepted from it. For this we recall the notions of *conflict-freeness* and *defense*.

**Definition 2.2** Let $\mathcal{AF} = \langle Args, Attack \rangle$ be an argumentation framework, $A \in Args$ an argument, and $\mathcal{E} \subseteq Args$ a set of arguments. We say that $\mathcal{E}$ *attacks* an argument $A$ if there is an argument $B \in \mathcal{E}$ that attacks $A$ (i.e., $(B, A) \in Attack$). The set of arguments that are attacked by $\mathcal{E}$ is denoted $\mathcal{E}^+$. We say that $\mathcal{E}$ *defends*

$A$ if $\mathcal{E}$ attacks every argument that attacks $A$. We denote by $\mathsf{Def}(\mathcal{E})$ the set of all the elements that are defended by $\mathcal{E}$. The set $\mathcal{E}$ is called *conflict-free* if it does not attack any of its elements (i.e., $\mathcal{E}^+ \cap \mathcal{E} = \emptyset$), $\mathcal{E}$ is called *admissible* if it is conflict-free and defends all of its elements (i.e., $\mathcal{E} \subseteq \mathsf{Def}(\mathcal{E})$), and $\mathcal{E}$ is *complete* if it is admissible and contains all the arguments that it defends ($\mathcal{E} = \mathsf{Def}(\mathcal{E})$).

The requirements defined above express basic properties that every plausible extension of a framework should have. Intuitively, a set of arguments is conflict-free if all of its elements 'can stand together' (since they do not attack each other), and admissibility guarantees that such elements 'can stand on their own', i.e., are able to respond to any attack by attacks of their own members (see also [5,6]).

Next, we recall some acceptability semantics for an argumentation framework.

**Definition 2.3** [11] Let $\mathcal{AF} = \langle Args, Attack \rangle$ be an argumentation framework.

- The minimal complete subset of $Args$ is the *grounded extension* of $\mathcal{AF}$,
- A maximal complete subset of $Args$ is a *preferred extension* of $\mathcal{AF}$,
- A complete subset $\mathcal{E}$ of $Args$ that attacks every argument in $Args \setminus \mathcal{E}$ is a *stable extension* of $\mathcal{AF}$.

We denote by $\mathsf{Cmpl}(\mathcal{AF})$ (respectively, $\mathsf{Grnd}(\mathcal{AF})$, $\mathsf{Prf}(\mathcal{AF})$, $\mathsf{Stbl}(\mathcal{AF})$) the set of all the complete (respectively, all the grounded, preferred, stable) extensions of $\mathcal{AF}$.

As shown in [11], an argumentation framework always has a unique grounded extension, but not necessarily a stable extension. If a stable extension does exist, it is also preferred. Other extensions are considered, e.g., in [5,6].

**Example 2.4** Consider the following argumentation framework:
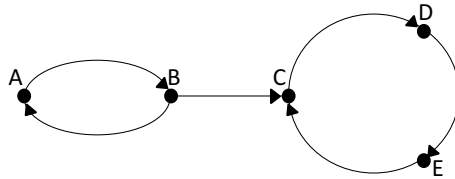


Here $\emptyset$, $\{A\}$, $\{B\}$ and $\{B, D\}$ are admissible sets, and with the exception of $\{B\}$ all of them are also complete. The grounded extension is $\emptyset$, the preferred extensions are $\{A\}$ and $\{B, D\}$, and the stable extension is $\{B, D\}$.

## 2.2   Sequent-Based Argumentation Frameworks

When it comes to specific applications of formal argumentation it is often useful to provide a specific account of the structure of arguments and the concrete nature of argumentative attacks. As indicated previously, we do so by following the sequent-based approach introduced in [1,4]. By this, we are able to fully abstract the notion of an argument and base it solely on the *logical dependency* between the support and the conclusion at hand. Furthermore, treating arguments as sequents allows to

incorporate well-established proof theoretical means for automatically constructing, identifying, and reasoning with arguments (see [1,4] for further justifications).

In what follows, we shall denote by $\mathcal{L}$ an arbitrary propositional language. Atomic formulas in $\mathcal{L}$ are denoted by $p, q, r$, arbitrary sets of formulas in $\mathcal{L}$ are denoted by $\mathcal{S}, \mathcal{T}$, and *finite* sets of formulas are denoted by $\Gamma, \Delta$.

**Definition 2.5** A (propositional) *logic* for a language $\mathcal{L}$ is a pair $\mathfrak{L} = \langle \mathcal{L}, \vdash \rangle$, where $\vdash$ is a (Tarskian) consequence relation for $\mathcal{L}$, that is, a binary relation between sets of formulas and formulas in $\mathcal{L}$, satisfying the following conditions:

*Reflexivity*: if $\psi \in \mathcal{S}$ then $\mathcal{S} \vdash \psi$.

*Monotonicity*: if $\mathcal{S} \vdash \psi$ and $\mathcal{S} \subseteq \mathcal{S}'$, then $\mathcal{S}' \vdash \psi$.

*Transitivity*: if $\mathcal{S} \vdash \psi$ and $\mathcal{S}', \psi \vdash \phi$ then $\mathcal{S}, \mathcal{S}' \vdash \phi$.

In addition, we shall assume that $\mathfrak{L}$ is *finitary*, that is: if $\mathcal{S} \vdash \psi$ then there is a *finite* theory $\Gamma \subseteq \mathcal{S}$ such that $\Gamma \vdash \psi$. [3]

We shall assume that the language $\mathcal{L}$ contains at least the following connectives:

- a $\vdash$-*negation* $\neg$, satisfying: $p \not\vdash \neg p$ and $\neg p \not\vdash p$ (for every atomic $p$), [4] and
- a $\vdash$-*conjunction* $\wedge$, satisfying: $\mathcal{S} \vdash \psi \wedge \phi$ iff $\mathcal{S} \vdash \psi$ and $\mathcal{S} \vdash \phi$.

For a finite set $\Gamma$ we denote by $\bigwedge \Gamma$ the conjunction of all the formulas in $\Gamma$.

### 2.2.1 Arguments As Sequents

There are several ways of defining the structure of an argument. The next definition is derived from the understanding that sequents are useful for representing logical arguments since they can be regarded as specific kinds of judgments (see [1,4] again).

**Definition 2.6** Let $\mathfrak{L} = \langle \mathcal{L}, \vdash \rangle$ be a propositional logic and $\mathcal{S}$ a set of $\mathcal{L}$-formulas.

- An $\mathcal{L}$-*sequent* (a sequent, for short) is an expression of the form $\Gamma \Rightarrow \Delta$, where $\Gamma$ and $\Delta$ are finite sets of $\mathcal{L}$-formulas, and $\Rightarrow$ is a reserved symbol (not in $\mathcal{L}$).
- An $\mathfrak{L}$-*argument* is an $\mathcal{L}$-sequent of the form $\Gamma \Rightarrow \psi$ where $\Gamma \vdash \psi$.
- An $\mathfrak{L}$-*argument* based on $\mathcal{S}$ is an $\mathfrak{L}$-argument $\Gamma \Rightarrow \psi$, where $\Gamma \subseteq \mathcal{S}$. The set of all the $\mathfrak{L}$-arguments that are based on $\mathcal{S}$ is denoted $\mathsf{Arg}_{\mathfrak{L}}(\mathcal{S})$.

**Note 1** Clearly, $\Gamma \Rightarrow \psi \in \mathsf{Arg}_{\mathfrak{L}}(\mathcal{S})$ for some (finite) $\Gamma \subseteq \mathcal{S}$ iff $\mathcal{S} \vdash \psi$.

Proof systems that operate on sequents (and so on arguments) are called *sequent calculi* [13]. The sequent calculi considered here consist of *inference rules* of the form

(1)
$$\frac{\Gamma_1 \Rightarrow \Delta_1 \dots \ \Gamma_n \Rightarrow \Delta_n}{\Gamma \Rightarrow \Delta}.$$

---

[3] The last property is satisfied by every logic that has a decent proof system, and will be useful in what follows (see, e.g., Note 1).

[4] There is no general agreement about the properties that a negation connective should satisfy (see, e.g., the papers collection in [12]). The requirement from a negation connective in this paper is a very minimal one, taken from [2,16].

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 above, and that $\Gamma \Rightarrow \Delta$ is its *conclusion*. As usual, axioms are treated as inference rules without conditions, i.e., they are rules of the form $\frac{}{\Gamma \Rightarrow \Delta}$.

In the sequel we shall usually assume that the underlying logic has a sound and complete sequent calculus, that is, a sequent-based proof system $\mathfrak{C}$, such that $\Gamma \vdash \psi$ iff the sequent $\Gamma \Rightarrow \psi$ is provable in $\mathfrak{C}$.

**Example 2.7** In this paper we shall usually use classical logic ($\mathsf{CL}$) for our demonstrations. Gentzen's well-known sequent calculus $LK$, which is sound and complete for $\mathsf{CL}$, is represented in Figure 1.

**Axioms:** $\qquad\qquad\qquad\qquad\qquad\qquad \psi \Rightarrow \psi$

**Structural Rules:**

$$\text{Weakening:} \quad \frac{\Gamma \Rightarrow \Delta}{\Gamma, \Gamma' \Rightarrow \Delta, \Delta'}$$

$$\text{Cut:} \quad \frac{\Gamma_1 \Rightarrow \Delta_1, \psi \qquad \Gamma_2, \psi \Rightarrow \Delta_2}{\Gamma_1, \Gamma_2 \Rightarrow \Delta_1, \Delta_2}$$

**Logical Rules:**

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

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

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

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

Fig. 1. The proof system $LK$

### 2.2.2 Attacks as Elimination Rules

Different attack relations have been considered in the literature for logical argumentation frameworks (see, e.g., [8,14,17]). In our case, attacks allow for the elimination (or, the discharging) of sequents. We shall denote by $\Gamma \not\Rightarrow \psi$ the elimination of the sequent $\Gamma \Rightarrow \psi$. Alternatively, $\overline{s}$ denotes the elimination of $s$. Now, a *sequent elimination rule* (or *attack rule*) has a similar form as an inference rule, except that its conclusion is a discharging of one condition, i.e., it is a rule of the following form:

$$(2) \qquad\qquad \frac{\Gamma_1 \Rightarrow \Delta_1, \ldots, \Gamma_n \Rightarrow \Delta_n}{\Gamma_n \not\Rightarrow \Delta_n}.$$

The prerequisites of attack rules usually consist of three ingredients. To help the

reader distinguish between these ingredients, we shall write them in a certain order: the first sequent in the rule's prerequisites is the "attacking" sequent, the last sequent in the rule's prerequisites 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.

**Example 2.8** Figure 2 lists some elimination rules in the context of logical argumentation systems (see also [4]). Similar rules for deontic logics and normative reasoning can be found in [22]. In this paper we shall exploit only a small number of these rules. We refer to [4,8,14,22] for further demonstrations of their usefulness.

| Defeat: | [Def] | $\dfrac{\Gamma_1 \Rightarrow \psi_1 \quad \psi_1 \Rightarrow \neg \bigwedge \Gamma_2 \quad \Gamma_2 \Rightarrow \psi_2}{\Gamma_2 \not\Rightarrow \psi_2}$ |
|---|---|---|
| Strong Defeat: | [S-Def] | $\dfrac{\Gamma_1 \Rightarrow \neg \bigwedge \Gamma_2 \quad \Gamma_2 \Rightarrow \psi_2}{\Gamma_2 \not\Rightarrow \psi_2}$ |
| Direct Defeat: | [D-Def] | $\dfrac{\Gamma_1 \Rightarrow \psi_1 \quad \psi_1 \Rightarrow \neg \phi \quad \Gamma_2, \phi \Rightarrow \psi_2}{\Gamma_2, \phi \not\Rightarrow \psi_2}$ |
| Indirect Defeat: | [I-Def] | $\dfrac{\Gamma_1 \Rightarrow \psi_1 \quad \psi_1 \Rightarrow \neg \bigwedge \Gamma_2 \quad \Gamma_2, \Gamma'_2 \Rightarrow \psi_2}{\Gamma_2, \Gamma'_2 \not\Rightarrow \psi_2}$ |
| Strong Direct Defeat: | [SD-Def] | $\dfrac{\Gamma_1 \Rightarrow \neg \phi \quad \Gamma_2, \phi \Rightarrow \psi_2}{\Gamma_2, \phi \not\Rightarrow \psi_2}$ |
| Strong Indirect Defeat: | [SI-Def] | $\dfrac{\Gamma_1 \Rightarrow \neg \bigwedge \Gamma_2 \quad \Gamma_2, \Gamma'_2 \Rightarrow \psi_2}{\Gamma_2, \Gamma'_2 \not\Rightarrow \psi_2}$ |
| Undercut: | [Ucut] | $\dfrac{\Gamma_1 \Rightarrow \psi_1 \quad \psi_1 \Rightarrow \neg \bigwedge \Gamma_2 \quad \neg \bigwedge \Gamma_2 \Rightarrow \psi_1 \quad \Gamma_2, \Gamma'_2 \Rightarrow \psi_2}{\Gamma_2, \Gamma'_2 \not\Rightarrow \psi_2}$ |
| Strong Undercut: | [S-Ucut] | $\dfrac{\Gamma_1 \Rightarrow \neg \bigwedge \Gamma_2 \quad \Gamma_2, \Gamma'_2 \Rightarrow \psi_2}{\Gamma_2, \Gamma'_2 \not\Rightarrow \psi_2}$ |
| Direct Undercut: | [D-Ucut] | $\dfrac{\Gamma_1 \Rightarrow \psi_1 \quad \psi_1 \Rightarrow \neg \gamma_2 \quad \neg \gamma_2 \Rightarrow \psi_1 \quad \Gamma_2, \gamma_2 \Rightarrow \psi_2}{\Gamma_2, \gamma_2 \not\Rightarrow \psi_2}$ |
| Strong Direct Undercut: | [SD-Ucut] | $\dfrac{\Gamma_1 \Rightarrow \neg \gamma_2 \quad \Gamma_2, \gamma_2 \Rightarrow \psi_2}{\Gamma_2, \gamma_2 \not\Rightarrow \psi_2}$ |
| Canonical Undercut: | [C-Ucut] | $\dfrac{\Gamma_1 \Rightarrow \psi_1 \quad \psi_1 \Rightarrow \neg \bigwedge \Gamma_2 \quad \neg \bigwedge \Gamma_2 \Rightarrow \psi_1 \quad \Gamma_2 \Rightarrow \psi_2}{\Gamma_2 \not\Rightarrow \psi_2}$ |
| Rebuttal: | [Reb] | $\dfrac{\Gamma_1 \Rightarrow \psi_1 \quad \psi_1 \Rightarrow \neg \psi_2 \quad \neg \psi_2 \Rightarrow \psi_1 \quad \Gamma_2 \Rightarrow \psi_2}{\Gamma_2 \not\Rightarrow \psi_2}$ |
| Strong Rebuttal: | [S-Reb] | $\dfrac{\Gamma_1 \Rightarrow \neg \psi_2 \quad \Gamma_2 \Rightarrow \psi_2}{\Gamma_2 \not\Rightarrow \psi_2}$ |
| Defeating Rebuttal: | [D-Reb] | $\dfrac{\Gamma_1 \Rightarrow \psi_1 \quad \psi_1 \Rightarrow \neg \psi_2 \quad \Gamma_2 \Rightarrow \psi_2}{\Gamma_2 \not\Rightarrow \psi_2}$ |
| Reductio Defeating Rebuttal: | [RD-Reb] | $\dfrac{\Gamma_1 \Rightarrow \psi_1 \quad \psi_2 \Rightarrow \neg \psi_1 \quad \Gamma_2 \Rightarrow \psi_2}{\Gamma_2 \not\Rightarrow \psi_2}$ |
| Indirect Rebuttal: | [I-Reb] | $\dfrac{\Gamma_1 \Rightarrow \psi_1 \quad \psi_1 \Rightarrow \varphi \quad \psi_2 \Rightarrow \neg \varphi \quad \Gamma_2 \Rightarrow \psi_2}{\Gamma_2 \not\Rightarrow \psi_2}$ |

Fig. 2. Sequent elimination rules

*2.2.3   Argumentation Settings and the Induced Logical Frameworks*

We now combine sequents and elimination rules for defining corresponding argumentation frameworks. For this, we need the following definition.

**Definition 2.9** An *argumentation setting* (a setting, for short) is a triple $\mathfrak{S} = \langle \mathfrak{L}, \mathfrak{C}, \mathfrak{A} \rangle$, where $\mathfrak{L} = \langle \mathcal{L}, \vdash \rangle$ is a propositional logic, $\mathfrak{C}$ is a sound and complete sequent calculus for $\mathfrak{L}$, and $\mathfrak{A}$ is a set of attack rules expressed in terms of $\mathcal{L}$-sequents.

**Definition 2.10** Let $\mathfrak{S} = \langle \mathfrak{L}, \mathfrak{C}, \mathfrak{A} \rangle$ be a setting, $\mathcal{S}$ a set of formulas, and $\theta$ an $\mathcal{L}$-substitution (that is, a substitution of atoms in $\mathcal{L}$ by formulas in $\mathcal{L}$).

- An inference rule $\mathcal{R}$ of the form of (1) above is $\mathsf{Arg}_{\mathfrak{L}}(\mathcal{S})$-*applicable* (for $\mathfrak{S}$, with respect to $\theta$), if for every $1 \leq i \leq n$, $\theta(\Gamma_i) \Rightarrow \theta(\Delta_i)$ is $\mathfrak{C}$-provable.

- An elimination rule $\mathcal{R}$ of the form of (2) above is $\mathsf{Arg}_{\mathfrak{L}}(\mathcal{S})$-*applicable* (for $\mathfrak{S}$, with respect to $\theta$), if $\theta(\Gamma_1) \Rightarrow \theta(\Delta_1)$ and $\theta(\Gamma_n) \Rightarrow \theta(\Delta_n)$ are in $\mathsf{Arg}_{\mathfrak{L}}(\mathcal{S})$ and for each $1 < i < n$, $\theta(\Gamma_i) \Rightarrow \theta(\Delta_i)$ is $\mathfrak{C}$-provable.

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

The induced argumentation framework is now defined as follows:

**Definition 2.11** Let $\mathfrak{S} = \langle \mathfrak{L}, \mathfrak{C}, \mathfrak{A} \rangle$ be a setting and let $\mathcal{S}$ be a set of formulas. The *sequent-based (logical) argumentation framework* for $\mathcal{S}$ (induced by $\mathfrak{S}$) is the argumentation framework $\mathcal{AF}_{\mathfrak{S}}(\mathcal{S}) = \langle \mathsf{Arg}_{\mathfrak{L}}(\mathcal{S}), Attack \rangle$, where $(s_1, s_2) \in Attack$ iff there is an $\mathcal{R} \in \mathfrak{A}$ such that $s_1$ $\mathcal{R}$-attacks $s_2$.

In what follows, somewhat abusing the notations, we shall sometimes identify *Attack* with $\mathfrak{A}$.

# 3   Dynamic Proofs

We now consider the notions of proofs (or derivations) for argumentation settings. In what follows we fix a given setting $\mathfrak{S} = \langle \mathfrak{L}, \mathfrak{C}, \mathfrak{A} \rangle$ (so that the underlying logic, a Gentzen-type proof system for it and the elimination rules are pre-determined).

**Definition 3.1** A (proof) *tuple* (also called *derivation steps* or *proof steps*) is a quadruple $\langle i, s, \mathsf{J}, \mathsf{A} \rangle$, where $i$ (the tuple's index) is a natural number, $s$ (the tuple's sequent) is either a sequent or an eliminated sequent, $\mathsf{J}$ (the tuple's justification) is a string, and $\mathsf{A}$ (the tuple's attacker) is an empty-set or a singleton of a sequent.

In what follows we shall sometimes omit the last component of a tuple in case that it is the empty-set, and omit the enclosing parentheses if it is a singleton.

As in 'standard' Gentzen-type systems, proofs are sequences of tuples, obtained by applications of rules. In our case, the underlying rules may be either introductory or eliminating, that is, applications of elements in $\mathfrak{C}$ or $\mathfrak{A}$, as defined next.

**Definition 3.2** Let $\mathfrak{S} = \langle \mathfrak{L}, \mathfrak{C}, \mathfrak{A} \rangle$ be a setting and $\mathcal{S}$ a set of formulas in $\mathcal{L}$. A *simple* (dynamic) *derivation* (with respect to $\mathfrak{S}$ and $\mathcal{S}$) is a finite sequence $\mathcal{D} = \langle T_1, \ldots T_m \rangle$ of proof tuples, where each $T_i \in \mathcal{D}$ is of one of the following forms:

- *Introducing Tuples*, of the form $T_i = \langle i, \theta(\Gamma) \Rightarrow \theta(\Delta), \mathsf{J}, \emptyset \rangle$, where there is an inference rule $\mathcal{R} \in \mathfrak{C}$ of the form of (1) above that is $\mathsf{Arg}_{\mathfrak{L}}(\mathcal{S})$-applicable for some $\mathcal{L}$-substitution $\theta$, and for every $1 \leq k \leq n$ there is a proof tuple $\langle i_k, s_k, \mathsf{J}_k, \emptyset \rangle$ in which $i_k < i$ and $s_k = \theta(\Gamma_k) \Rightarrow \theta(\Delta_k)$. In this case, $\mathsf{J} = \text{``}\mathcal{R}; i_1, \ldots, i_n\text{''}$.

- *Eliminating Tuples*, of the form $T_i = \langle i, \theta(\Gamma_n) \not\Rightarrow \theta(\Delta_n), \mathsf{J}, \theta(\Gamma_1) \Rightarrow \theta(\Delta_1) \rangle$, where there is an elimination rule $\mathcal{R} \in \mathfrak{A}$ of the form of (2) above that is $\mathsf{Arg}_{\mathfrak{L}}(\mathcal{S})$-applicable for some $\mathcal{L}$-substitution $\theta$,[5] and for every $1 \leq k \leq n$ there is a proof tuple $\langle i_k, s_k, \mathsf{J}_k, \emptyset \rangle$, in which $i_k < i$ and $s_k = \theta(\Gamma_k) \Rightarrow \theta(\Delta_k)$. In this case, $\mathsf{J} = \text{``}\mathcal{R}; i_1, \ldots, i_n\text{''}$.

In the sequel we shall sometimes identify introducing tuples with their derived sequents and eliminating tuples with their attacking sequents.

Given a simple derivation $\mathcal{D}$, we shall denote by $\mathsf{Top}(\mathcal{D})$ the tuple with the highest index in $\mathcal{D}$ and by $\mathsf{Tail}(\mathcal{D})$ the simple derivation $\mathcal{D}$ without $\mathsf{Top}(\mathcal{D})$. Also, we shall denote by $\mathcal{D}' = \mathcal{D} \oplus \langle T_1, \ldots, T_n \rangle$ the simple derivation whose prefix is $\mathcal{D}$ and whose suffix is $\langle T_1, \ldots, T_n \rangle$ (Thus, for instance, when $n = 1$ we have that $T = \mathsf{Top}(\mathcal{D} \oplus T)$ and $\mathcal{D} = \mathsf{Tail}(\mathcal{D} \oplus T)$). We call $\mathcal{D}'$ the *extension* of $\mathcal{D}$ by $\langle T_1, \ldots, T_n \rangle$.

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 3.3** Given a simple derivation $\mathcal{D}$, the iterative top-down function $\mathsf{Evaluate}(\mathcal{D})$ (see Algorithm 1) computes the following three sets: $\mathsf{Elim}(\mathcal{D})$ – the sequents that (at least once in $\mathcal{D}$) are attacked by an attacker which is not already attacked, $\mathsf{Attack}(\mathcal{D})$ – the sequents that attack a sequent in $\mathsf{Elim}(\mathcal{D})$, and $\mathsf{Accept}(\mathcal{D})$ – the derived sequents in $\mathcal{D}$ that are not in $\mathsf{Elim}(\mathcal{D})$.

**Definition 3.4** A simple derivation $\mathcal{D}$ is *coherent*, if $\mathsf{Attack}(\mathcal{D}) \cap \mathsf{Elim}(\mathcal{D}) = \emptyset$.[6]

Next, we show the adequacy of Algorithm 1 for a derivation $\mathcal{D}$, in terms of the argumentation framework induced by that derivation.

**Definition 3.5** Let $\mathcal{D}$ be a simple derivation. The *sequent-based argumentation framework that is induced by* $\mathcal{D}$ is the graph $\mathcal{AF}(\mathcal{D}) = \langle Derived(\mathcal{D}), Attack(\mathcal{D}) \rangle$, where $s \in Derived(\mathcal{D})$ if there is an introducing tuple $\langle i, s, \mathsf{J}, \emptyset \rangle$ in $\mathcal{D}$, and $(r, s) \in Attack(\mathcal{D})$ if there is an eliminating tuple $\langle i, \bar{s}, \mathsf{J}, r \rangle$ in $\mathcal{D}$.[7]

---

[5] Remember that this means, in particular, that the attacking sequent $\theta(\Gamma_1) \Rightarrow \theta(\Delta_1)$ and the attacked sequent $\theta(\Gamma_n) \Rightarrow \theta(\Delta_n)$ are both in $\mathsf{Arg}_{\mathfrak{L}}(\mathcal{S})$. This prevents situations in which, e.g., $\neg p \Rightarrow \neg p$ attacks $p \Rightarrow p$, although $\mathcal{S} = \{p\}$.

[6] That is, there is no sequent that eliminates another sequent, and later on is eliminated itself.

[7] Note that while $Derived(\mathcal{D})$ is the same as the set $\mathsf{Derived}(\mathcal{D})$ produced by the function $\mathsf{Evaluate}(\mathcal{D})$ (Algorithm 1), $Attack(\mathcal{D})$ is *not* the same as the set $\mathsf{Attack}(\mathcal{D})$ produced by that function, since here just the existence of an eliminating tuple merits a directed edge from the attacker to the attacked sequent, no matter whether the attacker is counter-attacked.

function Evaluate($\mathcal{D}$)  /* $\mathcal{D}$ – a simple derivation */

Attack := $\emptyset$;  Elim := $\emptyset$;  Derived := $\emptyset$;

while ($\mathcal{D}$ is not empty) do {

    if (Top($\mathcal{D}$) = $\langle i, s, \mathsf{J}, \emptyset \rangle$) then  /* Top($\mathcal{D}$) is an introducing tuple */
        Derived := Derived $\cup \{s\}$;

    if (Top($\mathcal{D}$) = $\langle i, \overline{s}, \mathsf{J}, r \rangle$) then  /* Top($\mathcal{D}$) is an attacking tuple */
        if ($r \notin$ Elim) then Elim := Elim $\cup \{s\}$ and Attack := Attack $\cup \{r\}$;

    $\mathcal{D}$ := Tail($\mathcal{D}$); }

Accept := Derived $-$ Elim;

return (Attack, Elim, Accept)

**Algorithm 1.** Evaluation of a simple derivation

**Proposition 3.6** *For every simple derivation $\mathcal{D}$ the set* Accept($\mathcal{D}$) *is conflict-free in $\mathcal{AF}(\mathcal{D})$. If $\mathcal{D}$ is coherent then* Accept($\mathcal{D}$) *is a stable extension of $\mathcal{AF}(\mathcal{D})$.*

**Sketch of proof.** We show that Accept($\mathcal{D}$) has the property defining stable sets, namely that Accept($\mathcal{D}$) $\cup$ Accept($\mathcal{D}$)$^+$ = Derived($\mathcal{D}$), leaving the other properties (conflict-freeness, admissibility and completeness) to the reader.

Since Derived($\mathcal{D}$) = Accept($\mathcal{D}$) $\cup$ Elim($\mathcal{D}$), it is enough to show that Accept($\mathcal{D}$)$^+$ coincides with Elim($\mathcal{D}$). Indeed,

- To see that Elim($\mathcal{D}$) $\subseteq$ Accept($\mathcal{D}$)$^+$, let $s \in$ Elim($\mathcal{D}$). Hence, there is an attacking tuple $T = \langle i, \overline{s}, \mathsf{J}, t \rangle \in \mathcal{D}$ and when the algorithm reaches $T$, we have that $t \notin$ Elim($\mathcal{D}$). Thus $t \in$ Attack($\mathcal{D}$), and since $\mathcal{D}$ is coherent, $t \notin$ Elim($\mathcal{D}$) also when the algorithm terminates. It follows that $t \in$ Accept($\mathcal{D}$), and so $s \in$ Accept($\mathcal{D}$)$^+$.

- To see that Accept($\mathcal{D}$)$^+$ $\subseteq$ Elim($\mathcal{D}$), let $s \in$ Accept($\mathcal{D}$)$^+$. Then there is a tuple $T = \langle i, \overline{s}, \mathsf{J}, t \rangle \in \mathcal{D}$ such that $t \in$ Accept($\mathcal{D}$). Since $\mathcal{D}$ is coherent, at the end of the execution of the algorithm $t \notin$ Elim($\mathcal{D}$). Thus, since Elim($\mathcal{D}$) grows monotonically during the execution, in particular $t \notin$ Elim($\mathcal{D}$) when the algorithm reaches the tuple $T$. It follows that $s \in$ Elim($\mathcal{D}$). □

Interestingly, the following proposition also holds:

**Proposition 3.7** *Let $\mathcal{D}$ be a simple derivation. If $\mathcal{E}$ is a stable extension of $\mathcal{AF}(\mathcal{D})$ then there is a coherent simple derivation $\mathcal{D}'$ such that $\mathcal{AF}(\mathcal{D}') = \mathcal{AF}(\mathcal{D})$ and $\mathcal{E} =$ Accept($\mathcal{D}'$).*

**Proof.** Let $\mathcal{D}$ be a simple derivation and let $\mathcal{E}$ be a stable extension of the sequent-based argumentation framework $\mathcal{AF}(\mathcal{D}) = \langle Derived(\mathcal{D}), Attack(\mathcal{D}) \rangle$ that is induced by $\mathcal{D}$. Consider a simple derivation $\mathcal{D}' = \mathcal{D}'_1 \oplus \mathcal{D}'_2 \oplus \mathcal{D}'_3$, where $\mathcal{D}'_1$ contains the tuples introducing the sequents in $Derived(\mathcal{D})$, $\mathcal{D}'_3$ consists of tuples of the form $\langle i, \overline{s}, \mathsf{J}, t \rangle$ where $t \in \mathcal{E}$ and $s \in \mathcal{E}^+$, and $\mathcal{D}'_2$ consists of the attacking tuples for the other

elements in $Attack(\mathcal{D})$ (the order of the elements in $\mathcal{D}'_2$ and in $\mathcal{D}'_3$ may be arbitrary, and some of these sequences may be empty for some $\mathcal{D}'$). Now, by the definition of $\mathcal{D}'$, clearly $\mathcal{AF}(\mathcal{D}') = \mathcal{AF}(\mathcal{D})$. Also, since $\mathcal{E}$ is stable, $\mathcal{E}^+ = Derived(\mathcal{D}) \setminus \mathcal{E} = Derived(\mathcal{D}') \setminus \mathcal{E}$, and so when the algorithm completes its pass over $\mathcal{D}'_3$ it holds that $\mathsf{Attack}(\mathcal{D}') = \mathcal{E}$ and $\mathsf{Elim}(\mathcal{D}') = Derived(\mathcal{D}') \setminus \mathcal{E}$. Clearly, the other tuples will not affect these sets, thus $\mathcal{D}'$ is coherent (since $\mathsf{Accept}(\mathcal{D}') \cap \mathsf{Elim}(\mathcal{D}') = \emptyset$) and $\mathsf{Accept}(\mathcal{D}') = Derived(\mathcal{D}') \setminus \mathsf{Elim}(\mathcal{D}') = \mathcal{E}$. □

Together, Propositions 3.6 and 3.7 show a correspondence between accepted sets of coherent simple derivations and the stable models of sequent-based argumentation framework that are induced by those derivations.

Now we are ready to define derivations in a dynamic proof system.

**Definition 3.8** Let $\mathfrak{S} = \langle \mathfrak{L}, \mathfrak{C}, \mathfrak{A} \rangle$ be an argumentation setting and let $\mathcal{S}$ be a set of formulas in $\mathcal{L}$. A *(dynamic) derivation* (for $\mathfrak{S}$, 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) $\mathcal{D}$ is an extension of 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 derived sequents (the $s$'s) are not in $\mathsf{Elim}(\mathcal{D})$.

 c) $\mathcal{D}$ is an extension of a dynamic derivation by a sequence $\langle T_1, \dots, T_n \rangle$ of eliminating tuples (of the form $\langle i, \bar{s}, \mathsf{J}, r \rangle$), such that:
   (i) $\mathcal{D}$ is coherent: $\mathsf{Attack}(\mathcal{D}) \cap \mathsf{Elim}(\mathcal{D}) = \emptyset$, and
   (ii) the new attacking sequents (the $r$'s) are not $\mathfrak{A}$-attacked by sequents in $\mathsf{Accept}(\mathcal{D}) \cap \mathsf{Arg}_{\mathfrak{L}}(\mathcal{S})$, where the attack is based on prerequisite conditions in $\mathcal{D}$.

**Note 2** Conditions c(i) and c(ii) above assure that the attacks are sound: by coherence neither of the attacking sequents of the additional elimination tuples is in $\mathsf{Elim}(\mathcal{D})$, and by Condition c(ii) they are not attacked by an accepted $\mathcal{S}$-based sequent. As we show below (see Footnote 11), these two conditions are not dependent.

Intuitively, one may think of a dynamic derivation as a proof that progresses over derivation steps. At each step the current derivation is extended by a 'block' of introducing or eliminating tuples (satisfying certain validity conditions), and the status of the derived sequents is updated accordingly. In particular, derived sequents may be eliminated (i.e., marked as unreliable) in light of new proof tuples, but also the other way around is possible: an eliminated sequent may be 'restored' if its attacking tuple is counter-attacked by a new eliminating tuple. It follows that previously derived data may not be derived anymore (and vice-versa) until and unless new derived information revises the state of affairs (see the examples in Section 4).

**Proposition 3.9** *Every dynamic derivation is coherent.*

**Proof.** Easily follows from Definition 3.8. □

The next definition, of the outcomes of a dynamic derivation, states that we can safely (or 'finally') derive a derived sequent only when we are sure that there is no scenario in which it will be eliminated in some extension of the derivation.

**Definition 3.10** Let $\mathfrak{S} = \langle \mathfrak{L}, \mathfrak{C}, \mathfrak{A} \rangle$ be a setting and let $\mathcal{S}$ be a set of formulas in $\mathfrak{L}$. A sequent $s$ is *finally derived* (or safely derived) in a dynamic derivation $\mathcal{D}$ (for $\mathfrak{S}$, based on $\mathcal{S}$), if $s \in \mathsf{Accept}(\mathcal{D})$, and $\mathcal{D}$ cannot be extended to a dynamic derivation $\mathcal{D}'$ (for $\mathfrak{S}$, based on $\mathcal{S}$) such that $s \in \mathsf{Elim}(\mathcal{D}')$.

**Note 3** Unlike ordinary proofs (e.g., in standard sequent calculi), the amount of derived sequents does *not* grow monotonically in the size of the derivation. However, final derivability *is* monotonic in the length of dynamic derivations. Indeed,

**Proposition 3.11** *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 3.12** Given an argumentation setting $\mathfrak{S} = \langle \mathfrak{L}, \mathfrak{C}, \mathfrak{A} \rangle$ and a set $\mathcal{S}$ of formulas, we denote by $\mathcal{S} \mathrel{|\!\sim}_{\mathfrak{S}} \psi$ that there is an $\mathcal{S}$-based dynamic derivation for $\mathfrak{S}$, in which $\Gamma \Rightarrow \psi$ is finally derived for some finite $\Gamma \subseteq \mathcal{S}$.

When the underlying argumentation setting is clear from the context we shall sometimes abbreviate $\mathrel{|\!\sim}_{\mathfrak{S}}$ by $\mathrel{|\!\sim}$.

# 4 Some Examples

We now give some examples of dynamic derivations. To simplify the reading, in the examples below we shall sometimes use abbreviations or omit some details, e.g. the tuple signs in proof steps.

**Example 4.1** Consider the argumentation setting $\mathfrak{S} = \langle \mathsf{CL}, LK, \mathsf{Ucut} \rangle$, based on classical logic $\mathsf{CL}$, its sequent calculus $LK$ (Figure 1), and the attack rule Undercut (Figure 2). Below is a dynamic derivation for $\mathfrak{S}$, based on $\mathcal{S}_1 = \{p, \neg p, q\}$:

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

Note that $q \Rightarrow q$ is finally derived here. Indeed, the only sequents in $\mathsf{Arg}_{\mathsf{CL}}(\mathcal{S}_1)$ that can potentially attack $q \Rightarrow q$ are of the form $p, \neg p \Rightarrow \psi$ or $p, \neg p, q \Rightarrow \psi$, where $\psi$ is logically equivalent to $\neg q$, however those sequents are counter attacked by $\Rightarrow p \vee \neg p$ (which is derived in Tuple 3), using the justifications in Tuples 4 and 5. [8]

---

[8] It is important to note that Ucut-attackers of $q \Rightarrow q$ may still be derived in an extension of $\mathcal{D}$. However, as note above, by Condition (ii) in Definition 3.8(c) they cannot be used for eliminating $q \Rightarrow q$.

Thus, the above derivation cannot be extended to a derivation in which $q \Rightarrow q$ is eliminated, and so $\mathcal{S}_1 \mathrel{|\!\sim} q$.

The situation is completely different as far as $p \Rightarrow p$ is concerned. This is due to the fact that the above derivation *can* be extended by the following tuples, yielding an elimination of $p \Rightarrow p$:

| | | | |
|---|---|---|---|
| 7. | $\neg p \Rightarrow \neg p$ | Axiom | |
| 8. | $p \not\Rightarrow p$ | Ucut, 7, 7, 7, 1 | $\neg p \Rightarrow \neg p$ |

In turn, this derivation can be further extended, to get an attack on $\neg p \Rightarrow \neg p$:

| | | | |
|---|---|---|---|
| 9. | $p \Rightarrow \neg\neg p$ | ... | |
| 10. | $\neg\neg p \Rightarrow p$ | ... | |
| 11. | $\neg p \not\Rightarrow \neg p$ | Ucut, 1, 9, 10, 7 | $p \Rightarrow p$ |

In the last derivation $p \Rightarrow p$ is not eliminated anymore. Nevertheless, $p \Rightarrow p$ can be re-attacked by the sequent $\neg p \Rightarrow \neg p$. Alternatively, $p \Rightarrow p$ can be re-attacked by any sequent of the form $\neg p \Rightarrow \psi$, where $\psi$ is equivalent to $\neg p$ (e.g., $\psi = \neg^{2n+1} p$, i.e., a formula in which $p$ is preceded by an odd number of negations). As a consequence, $p \not\Rightarrow p$ is reintroduced in extensions of this derivation, and similarly $\neg p \not\Rightarrow \neg p$ may be reintroduced as well. It follows that neither of these sequents is finally derived. In an analogous way any dynamic derivation based on $\mathcal{S}_1$ can always be extended in such a way that all the sequents in $\mathsf{Arg}_{\mathfrak{L}}(\mathcal{S}_1)$ whose conclusion is $p$ (respectively, $\neg p$) are eliminated, and so $\mathcal{S}_1 \mathrel{|\!\not\sim} p$ (respectively, $\mathcal{S}_1 \mathrel{|\!\not\sim} \neg p$).

This state of affairs is intuitively justified by the fact that while $q$ is not related to the inconsistency in $\mathcal{S}_1$ and so it may safely follow from $\mathcal{S}_1$, the information in $\mathcal{S}_1$ about $p$ is contradictory, and so neither $p$ nor $\neg p$ may be safely inferred from $\mathcal{S}_1$.

**Example 4.2** Let us consider the following variation of the previous example. The underlying setting is the same as before: $\mathfrak{S} = \langle \mathsf{CL}, LK, \mathsf{Ucut} \rangle$, but now we take the conjunction of $p$ and $q$: $\mathcal{S}_1' = \{p \wedge q, \neg p\}$. Again, although both of $p \wedge q \Rightarrow p$ and $\neg p \Rightarrow \neg p$ are $LK$-derivable, neither $p$ nor $\neg p$ follows according to $\mathfrak{S}$ from $\mathcal{S}_1'$, because, e.g., the first sequent Ucut-attacks the other sequent and is Ucut-attacked by the sequent $\neg p \Rightarrow \neg(p \wedge q)$ (the details are quite similar to those in Example 4.1). This time, however, $q$ is *not* $\mathfrak{S}$-derivable from $\mathcal{S}_1'$, because both the sequents $p \wedge q \Rightarrow q$ and $\neg p, p \wedge q \Rightarrow q$ are also Ucut-attacked by the $LK$-derivable sequent $\neg p \Rightarrow \neg(p \wedge q)$ and cannot be permanently defended by sequents in $\mathsf{Arg}_{\mathsf{CL}}(\mathcal{S}_1')$.[9]

**Note 4** The last two examples show in particular that $\mathrel{|\!\sim}_{\mathfrak{S}}$ is sensitive to the syntactic form of the premises: although $\mathcal{S}_1$ (Example 4.1) and $\mathcal{S}_1'$ (Example 4.2) are $\mathsf{CL}$-equivalent, their $\mathfrak{S}$-conclusions are *not* the same. This may be intuitively justified by the fact that in $\mathcal{S}_1'$, unlike in $\mathcal{S}_1$, $q$ is not neutral with respect to the inconsistency of the set of premises and it is 'linked' to $p$ by the conjunction (as is also reflected by the above Ucut-attack on $p \wedge q$). Indeed, syntax sensitivity is

---

[9] Note that the $\mathsf{Arg}_{\mathsf{CL}}(\mathcal{S}_1')$-sequent $p \wedge q \Rightarrow p$ does not prevent the Ucut-attack on $p \wedge q \Rightarrow q$ by the $\mathsf{Arg}_{\mathsf{CL}}(\mathcal{S}_1')$-sequent $\neg p \Rightarrow \neg(p \wedge q)$, because the latter attacks both of them. This situation is different from the one in Example 4.1, where $\Rightarrow p \vee \neg p$ 'blocks' any potential Ucut-attack on $q \Rightarrow q$, since in Example 4.1 $\Rightarrow p \vee \neg p$ couldn't be counter Ucut-attacked.

not unusual in the context of non-monotonic reasoning. The same phenomenon is implied by Darwiche and Pearl's postulates for iterated belief revision [10], according to which a new information abolishes any existing data that contradicts it. Thus, for instance, if Reasoner 1 'believes' that $p \wedge q$ and Reasoner 2 'believes' that $p$ and that $q$, the introduction of $\neg p$ forces Reasoner 1 to retract $p \wedge q$, while Reasoner 2 has to withdraw $p$ only. As a consequence, Reasoner 2 still believes $q$, while Reasoner 1 has no indication that $q$ holds. A similar phenomenon happens when the maximally consistent subsets of the premises are taken into account: $q$ is in the intersection of the maximally consistent subsets of $\mathcal{S}_1$, while the intersection of the maximally consistent subsets of $\mathcal{S}'_1$ is empty (see also [19]). [10]

**Example 4.3** Consider a logic with a negation connective $\neg$ (so $p \not\vdash \neg p$ and $\neg p \not\vdash p$), which doesn't respect double-negation introduction (i.e., $p \not\vdash \neg\neg p$), and suppose that Direct Defeat (D-Def; See Figure 2) is the only attack rule. Let $\mathcal{S}_2 = \{p, \neg p, \neg\neg p, \neg\neg\neg p, \neg\neg\neg\neg p\}$. We write $s_i$ ($i \in \mathbb{N}$) for the sequent $\neg^i p \Rightarrow \neg^i p$ (where $\neg^0 p = p$). Note that by reflexivity $s_i$ is provable in any complete calculus for the base logic. Now, consider the following sequence $\mathcal{D}$ of proof tuples:

1. $s_0$      Axiom
2. $s_1$      Axiom
3. $s_2$      Axiom
4. $\overline{s_1}$      D-Def, 3, 3, 2     $s_2$
5. $s_3$      Axiom
6. $\overline{s_0}$      D-Def, 2, 2, 1     $s_1$
7. $\overline{s_2}$      D-Def, 5, 5, 3     $s_3$
8. $s_4$      Axiom

It is easy to verify that $\mathcal{D}$ is a valid derivation. Extending it only with the tuple

9. $\overline{s_3}$      D-Def, 8, 8, 5     $s_4$

yields a simple derivation $\mathcal{D}'$, in which the attacker ($s_4$) is not counter-attacked by an accepted sequent, yet $\mathcal{D}'$ is not coherent since $s_1 \in \mathsf{Attack}(\mathcal{D}') \cap \mathsf{Elim}(\mathcal{D}')$. [11] Note, however, that $\mathcal{D}$ may be extended to a coherent derivation containing Tuple 9, provided that the latter is introduced together with the following eliminating tuple:

10. $\overline{s_1}$      D-Def, 3, 3, 2     $s_2$

Indeed, the extension of $\mathcal{D}$ with the sequence $\langle T_9, T_{10}\rangle$ *is* a valid derivation. This demonstrates the need in Definition 3.8 to introduce more than one elimination tuple at a time.

Let us now check what can be finally derived from $\mathcal{S}_2$. First, the sequent $s_4$ is attacked according to D-Def only by sequents whose right-hand side is $\neg^5 p$, but since double-negation introduction does not hold, such sequents cannot be in $\mathsf{Arg}_{\mathfrak{L}}(\mathcal{S}_2)$.

---

[10] Syntax dependency ceases to hold when $\mathcal{S}_1$ (or $\mathcal{S}'_1$) is consistent. This follows from Proposition 5.2 below.
[11] This shows, in particular, that the two conditions in Definition 3.8(c) are not dependent.

It follows that $s_4$ is finally derived by the above derivation, and so $\mathcal{S}_2 \hspace{-0.2em}\mid\hspace{-0.55em}\sim \neg^4 p$. Also, $s_3$ cannot be finally derived since any derivation in which it is derived can be extended by a tuple of the form $\langle i, \overline{s_3}, \text{D-Def}, s_4 \rangle$, which causes the elimination of $s_3$. Thus $\mathcal{S}_2 \hspace{-0.2em}\not\hspace{-0.25em}\mid\hspace{-0.55em}\sim \neg^3 p$. In turn, since the attacker ($s_3$) of $s_2$ is eliminated and cannot be recovered, $s_2$ is finally derived, thus $\mathcal{S}_2 \hspace{-0.2em}\mid\hspace{-0.55em}\sim \neg\neg p$. Similar considerations show that in this case $\mathcal{S}_2 \hspace{-0.2em}\not\hspace{-0.25em}\mid\hspace{-0.55em}\sim \neg p$ and that $\mathcal{S}_2 \hspace{-0.2em}\mid\hspace{-0.55em}\sim p$.

**Note 5** The last example emphasizes the basic difference between the derivation process introduced here and the one considered in [3]. While the process in [3] allows to reintroduce sequents irrespective of whether they are attacked, here the way sequents can be introduced in a proof is restricted and it depends on the already introduced elimination sequents. Thus, e.g., while according to the approach in [3] the sequent $\neg p \Rightarrow \neg p$ may be reintroduced in an extension of the dynamic derivation of Example 4.3, this is not possible according to the present formalism. Hence, according to [3] only $s_4$ is finally derivable in Example 4.3, while in our case both $s_2$ and $s_0$ are also finally derivable, although they are attacked. This allows for a better 'diffusion of attacks' and it is in line with standard extensions of the corresponding argumentation frameworks (see [11]): although $s_2$ is attacked by $s_3$, that attack is counter-attacked by $s_4$, and so $s_2$ is 'defended' or 'reinstated' by $s_4$ (see also Proposition 3.6).

# 5    Some Properties of $\hspace{-0.2em}\mid\hspace{-0.55em}\sim$

In this section we consider some properties of the entailment relations that are induced by dynamic proof systems according to Definition 3.12. We still denote by $\hspace{-0.2em}\mid\hspace{-0.55em}\sim$ an arbitrary entailment relation that is covered by this definition. Due to lack of space proofs in this section are omitted and will appear in an extended version of this paper.

*Relations between* $\hspace{-0.2em}\mid\hspace{-0.55em}\sim$ *and* $\vdash$

We start with some results concerning the relations between the base consequence relation and the entailments induced by the corresponding argumentation setting. In these propositions we refer to an entailment $\hspace{-0.2em}\mid\hspace{-0.55em}\sim$ that is induced by an argumentation setting $\mathfrak{S} = \langle \mathfrak{L}, \mathfrak{C}, \mathfrak{A} \rangle$ with a base logic $\mathfrak{L} = \langle \mathcal{L}, \vdash \rangle$.

(i) Tarskian consequence relations may be viewed as particular $\mathfrak{S}$-entailments:

**Proposition 5.1** *If* $\mathfrak{A} = \emptyset$ *then* $\hspace{-0.2em}\mid\hspace{-0.55em}\sim$ *and* $\vdash$ *coincide.*

(ii) Another case where $\hspace{-0.2em}\mid\hspace{-0.55em}\sim$ and $\vdash$ correlate is the following:

**Proposition 5.2** *If* $\mathcal{S}$ *is conflict-free with respect to* $\mathfrak{S}$ *(that is, there are no* $\mathfrak{A}$*-attacks between the elements in* $\mathsf{Arg}_{\mathfrak{L}}(\mathcal{S})$*), then* $\mathcal{S} \hspace{-0.2em}\mid\hspace{-0.55em}\sim \psi$ *iff* $\mathcal{S} \vdash \psi$.

(iii) In general, $\hspace{-0.2em}\mid\hspace{-0.55em}\sim$ is weaker than $\vdash$:

**Proposition 5.3** *If* $\mathcal{S} \hspace{-0.2em}\mid\hspace{-0.55em}\sim \psi$ *then* $\mathcal{S} \vdash \psi$.

(iv) The converse of Proposition 5.3 holds for $\vdash$-theorems and theorem-preserving rules:

**Definition 5.4** An elimination rule $R$ of the form of (2) above is *theorem-preserving* (with respect to a logic $\mathfrak{L}$), if there is no application of $R$ by a substitution $\theta$ such that $\theta(\Gamma_n) = \emptyset$.

Intuitively, a rule is theorem-preserving if it cannot be used for attacking theorems of the underlying (base) logic. The various variations of Undercut and Defeat in Figure 2 are examples of rules that are theorem-preserving with respect to any logic.

**Proposition 5.5** *If $\mathfrak{A}$ consists only of theorem-preserving rules, then $\vdash \psi$ implies that $\hspace{1pt}\mid\!\sim \psi$.*

**Corollary 5.6** *If $\mathfrak{A}$ consists only of theorem-preserving rules, then (i) $\vdash \psi$ iff $\hspace{1pt}\mid\!\sim \psi$, and (ii) $\mathfrak{C}$ is weakly sound and complete for $\hspace{1pt}\mid\!\sim$ (that is, $\hspace{1pt}\mid\!\sim \psi$ iff $\Rightarrow \psi$ is $\mathfrak{C}$-derivable).*

We note also that some properties of the base logic are 'inherited' by $\hspace{1pt}\mid\!\sim$. One of them is $\neg$-paraconsistency [9]:

**Proposition 5.7** *If $\vdash$ is $\neg$-paraconsistent (that is, there are atoms $p, q$ such that $p, \neg p \nvdash q$) then so is $\hspace{1pt}\mid\!\sim$.*

*Cautious Reflexivity*

As the examples in Section 4 show, in general $\hspace{1pt}\mid\!\sim$ is *not* reflexive: a formula $\psi$ does not necessarily follow from $\mathcal{S}$ even if $\psi \in \mathcal{S}$. Yet, the next proposition and corollary show that $\hspace{1pt}\mid\!\sim$ is *cautiously reflexive*.

**Proposition 5.8** *If $\mathcal{S}$ is conflict-free then $\mathcal{S} \hspace{1pt}\mid\!\sim \psi$ for all $\psi \in \mathcal{S}$.*

**Corollary 5.9**

(i) *For every formula $\psi$ such that $\{\psi\}$ is conflict-free in $\mathfrak{S}$, we have that $\psi \hspace{1pt}\mid\!\sim \psi$.*

(ii) *For every atom $p$ it holds that $p \hspace{1pt}\mid\!\sim p$.*

**Note 6** The condition in the last proposition and corollary is indeed required. For instance, if $\hspace{1pt}\mid\!\sim$ is the entailment relation that is induced by $\mathfrak{S} = \langle \mathsf{CL}, LK, \{\mathsf{Ucut}\} \rangle$ (Example 4.1) then $p \wedge \neg p \hspace{1pt}\not\mid\!\sim p \wedge \neg p$.

*Restricted Monotonicity*

Clearly, $\hspace{1pt}\mid\!\sim$ is not monotonic. For instance, by Corollary 5.9 $p \hspace{1pt}\mid\!\sim p$ while Example 4.1 shows a case in which $p, \neg p, q \hspace{1pt}\not\mid\!\sim p$. Like reflexivity, monotonicity can be guaranteed in particular cases. For instance, as Proposition 5.12 below shows, when adding unrelated information to a framework with Undercut, this information should not disturb previous inferences. For this proposition we first define in precise terms what 'unrelated information' means and then recall the known notion of uniformity.

**Definition 5.10** Let $\mathcal{S}$ be a set of formulas and $\psi$ a formula in a language $\mathcal{L}$. We denote by $\mathsf{Atoms}(\mathcal{S})$ the set of atomic formulas that appear (in some subformula of a formula) in $\mathcal{S}$. We say that $\mathcal{S}$ is *relevant* to $\psi$, if $\mathsf{Atoms}(\mathcal{S}) \cap \mathsf{Atoms}(\{\psi\}) = \emptyset$ implies that $\mathcal{S} = \emptyset$. A nonempty set $\mathcal{S}$ is *irrelevant* to a (nonempty) set $\mathcal{T}$ if $\mathcal{S}$ is not relevant to any formula in $\mathcal{T}$, i.e.: $\mathsf{Atoms}(\mathcal{S}) \cap \mathsf{Atoms}(\mathcal{T}) = \emptyset$.

**Definition 5.11** Let $\mathfrak{L} = \langle \mathcal{L}, \vdash \rangle$ be a propositional logic. A set $\mathcal{S}$ of $\mathcal{L}$-formulas is $\vdash$-*consistent*, if there exists an $\mathcal{L}$-formula $\psi$ such that $\mathcal{S} \nvdash \psi$. We say that $\mathfrak{L}$ is *uniform*, if $\mathcal{S}_1 \vdash \psi$ when $\mathcal{S}_1, \mathcal{S}_2 \vdash \psi$ and $\mathcal{S}_2$ is $\vdash$-consistent and irrelevant to $\mathcal{S}_1 \cup \{\psi\}$.

**Note 7** By Łos-Suzsko Theorem [15], a finitary propositional logic $\langle \mathcal{L}, \vdash \rangle$ is uniform iff it has a single characteristic matrix (see also [23]). Thus, classical logic as well as many other logics are uniform.

**Proposition 5.12** *Let $\mathfrak{S} = \langle \mathfrak{L}, \mathfrak{C}, \{\mathsf{Ucut}\} \rangle$ be a setting whose base logic $\mathfrak{L}$ is uniform, and let $\vdash\!\!\!\sim$ be the induced entailment. If $\mathcal{S}_1 \vdash\!\!\!\sim \psi$ and $\mathcal{S}_2$ is a $\vdash$-consistent set of formulas that is irrelevant to $\mathcal{S}_1$, then $\mathcal{S}_1, \mathcal{S}_2 \vdash\!\!\!\sim \psi$.*

**Note 8** A crucial property for Proposition 5.12 is that Ucut-attacks are preserved when the premises of the attacked sequents are weakened: if $\Gamma \Rightarrow \psi$ is Ucut-attacked then $\Gamma' \Rightarrow \psi$ is Ucut-attacked (by the same attacker) whenever $\Gamma'$ contains $\Gamma$.

**Note 9** The last proposition holds also for Direct Undercut. Under the additional condition that $\mathcal{S}_2$ is irrelevant to $\{\psi\}$ the proposition holds also for Rebuttal.

## 6  Conclusion and Future Work

We have introduced a deductive method for reasoning with sequent-based argumentation systems. The non-monotonic nature of its derivations resembles other dynamic proof systems in other contexts, such as adaptive logics [7,21]. In our setting, a proof progressing over derivation steps may be viewed as a dialogue explicating actual (dynamic) reasoning in an argumentation framework. This, and other opportunities of using dynamic derivations for imitating commonsense reasoning, as well as the exact relations to related semantics, like those that are induced by Dung-style formalisms, should yet be explored in future work.

## References

[1] O. Arieli. A sequent-based representation of logical argumentation. In *Proc. CLIMA'13*, LNCS 8143, pages 69–85. Springer, 2013.

[2] O. Arieli, A. Avron, and A. Zamansky. Maximal and premaximal paraconsistency in the framework of three-valued semantics. *Studia Logica*, 97(1):31–60, 2011.

[3] O. Arieli and C. Straßer. Dynamic derivations for sequent-based logical argumentation. In *Proc. COMMA'14*, Frontiers in Artificial Intelligence and Applications 266, pages 89–100. IOS Press, 2014.

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

[5] P. Baroni, M. Caminada, and M. Giacomin. An introduction to argumentation semantics. *The Knowledge Engineering Review*, 26(4):365–410, 2011.

[6] P. Baroni and M. Giacomin. Semantics for abstract argumentation systems. In I. Rahwan and G. R. Simary, editors, *Argumentation in Artificial Intelligence*, pages 25–44. 2009.

[7] D. Batens. A universal logic approach to adaptive logics. *Logica Universalis*, 1:221–242, 2007.

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

[9] N. C. A. da Costa. On the theory of inconsistent formal systems. *Notre Dame Journal of Formal Logic*, 15:497–510, 1974.

[10] A. Darwiche and J. Pearl. On the logic of iterated belief revision. *Artificial Intelligence*, 89(1–2):1–29, 1997.

[11] P. M. Dung. On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and $n$-person games. *Artificial Intelligence*, 77:321–357, 1995.

[12] D. Gabbay and H. Wansing, editors. *What is Negation?*, volume 13 of *Applied Logic Series*. Springer, 1999.

[13] G. Gentzen. Investigations into logical deduction, 1934. In German. An English translation appears in 'The Collected Works of Gerhard Gentzen', edited by M. E. Szabo, North-Holland, 1969.

[14] N. Gorogiannis and A. Hunter. Instantiating abstract argumentation with classical logic arguments: Postulates and properties. *Artificial Intelligence*, 175(9–10):1479–1497, 2011.

[15] J. Łos and R. Suzsko. Remarks on sentential logics. *Indagationes Mathematicae*, 20:177–183, 1958.

[16] J. Marcos. On negation: Pure local rules. *Journal of Applied Logic*, 3(1):185–219, 2005.

[17] J. Pollock. How to reason defeasibly. *Artificial Intelligence*, 57(1):1–42, 1992.

[18] H. Prakken. Two approaches to the formalisation of defeasible deontic reasoning. *Studia Logica*, 57(1):73–90, 1996.

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

[20] G. R Simari and R. P. Loui. A mathematical treatment of defeasible reasoning and its implementation. *Artificial Intelligence*, 53(2–3):125–157, 1992.

[21] C. Straßer. *Adaptive Logic and Defeasible Reasoning. Applications in Argumentation, Normative Reasoning and Default Reasoning*, volume 38 of *Trends in Logic*. Springer, 2014.

[22] C. Straßer and O. Arieli. Sequent-based argumentation for normative reasoning. In *Proc. DEON'14*, LNCS 8554, pages 224–240. Springer, 2014.

[23] A. Urquhart. Many-valued logic. In D. Gabbay and F. Guenthner, editors, *Handbook of Philosophical Logic*, volume II, pages 249–295. Kluwer, 2001. second edition.