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 [6,12,13,15]. The basic entities in this context are called *arguments*. An argument is a pair of a finite set of formulas (Γ, the support set) and a formula (ψ, 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 [3], this gives rise to the association of arguments with Gentzen's notion of *sequents* [9], 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.

[1] Email: oarieli@mta.ac.il. Supported by the Israel Science Foundation (grant No. 817/15).

[2] Email: Christian.Strasser@ruhr-uni-bochum.de. Supported by the Alexander von Humboldt Foundation and the German Ministry for Education and Research.

A first step towards a proof theoretical investigation of sequent-based logical argumentation is done in [2]. In this paper we revise and extend that work (see also Note 4 below). First, we recall some basic notions behind abstract argumentation in general and (sequent-based) logical argumentation in particular. Then 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. We show how despite of 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.

2 Preliminaries

We start by reviewing the notion of sequent-based argumentation, as defined in [3]. 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* [8] is a pair $\mathcal{AF} = \langle \mathit{Args}, \mathit{Attack} \rangle$, where Args is an enumerable 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 $\mathcal{AF} = \langle \mathit{Args}, \mathit{Attack} \rangle$ be an argumentation framework, $A \in \mathit{Args}$ an argument, and $\mathcal{E} \subseteq \mathit{Args}$ 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., $(B, A) \in \mathit{Attack}$). 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 their own attack (see also [4,5]).

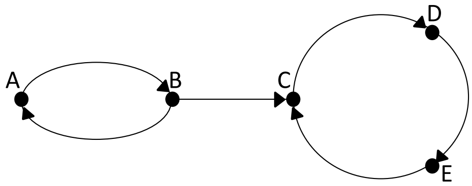
Next, we recall some acceptability semantics for an argumentation framework.

Definition 2.3 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 $\mathcal{AF}$. [3]

Example 2.4 Consider the following argumentation framework:



Here $\emptyset$, $\{A\}$, $\{B\}$ and $\{B, D\}$ are admissible sets, and except 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, here we follow the sequent-based approach introduced in [1,3] (see these papers for a justification of our choice).

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 Γ, Δ.

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 $\Gamma \vdash \psi$. [4]

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), and

[3] Properties of these extensions can be found in [8]; Further extensions are considered, e.g., in [4,5].

[4] 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).

- 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 Γ we denote by $\bigwedge\Gamma$ the conjunction of all the formulas in Γ.

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 [1,3] 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 Γ and Δ are finite sets of $\mathcal{L}$-formulas, and $\Rightarrow$ is a new 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 $\mathcal{L}$-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* [9]. The sequent calculi considered here consist of *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 *prerequisits*) of the rule above, and that $\Gamma \Rightarrow \Delta$ is its *conclusion*. [5]

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 (CL) for our demonstrations. Gentzen's well-known sequent calculus LK, which is sound and complete for CL, is represented in Figure 1.

2.2.2 Attacks as Elimination Rules

Different attack relations have been considered in the literature for logical argumentation frameworks (see, e.g., [6,10,12]). 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, expect that its conclusion is a discharging of the last condition, i.e., it is a rule of the following form:

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

The prerequisites of attack rules usually consist of three ingredients. We shall usually say that 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

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

Axioms: $\psi \Rightarrow \psi$

Structural Rules:

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

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

Logical Rules:

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

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

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

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

Fig. 1. The proof system *LK*

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 [3]). Similar rules for deontic logics and normative reasoning can be found in [16].

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 θ an $\mathcal{L}$-substitution.

- 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 θ), 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 θ), 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:

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

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 $\mathcal{AF}_{\mathfrak{S}}(\mathcal{S}) = \langle \mathsf{Arg}_{\mathfrak{L}}(\mathcal{S}), \mathit{Attack} \rangle$, where $(s_1, s_2) \in \mathit{Attack}$ 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, J (the tuple's justification) is a string, and A (the tuple's attacker) is an empty-set or a singleton of a sequent.[6]

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 $\mathcal{D} = \langle T_1, \dots T_m\rangle$ of proof tuples, where each $T_i \in \mathcal{D}$ is of one of the following forms:

- $T_i = \langle i, \theta(\Gamma) \Rightarrow \theta(\Delta), \mathsf{J}, \emptyset\rangle$, 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 θ, where 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 is the sequent $\theta(\Gamma_k) \Rightarrow \theta(\Delta_k)$. In this case, $\mathsf{J} =$"$\mathcal{R}; i_1, \dots, i_n$". In what follows we shall call T_i an *introducing tuple*.
- $T_i = \langle i, \theta(\Gamma_n) \not\Rightarrow \theta(\Delta_n), \mathsf{J}, \theta(\Gamma_1) \Rightarrow \theta(\Delta_1)\rangle$, 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 θ,[7] 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, $\mathsf{J} =$"$\mathcal{R}; i_1, \dots, i_n$". In what follows we shall call T_i an *eliminating tuple*.

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 $\mathcal{D}' = \mathcal{D} \oplus \langle T_1, \dots, T_n\rangle$ the simple derivation whose prefix is $\mathcal{D}$ and whose suffix is $\langle T_1, \dots, T_n\rangle$ (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 $\langle T_1, \dots, T_n\rangle$.

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 $\mathsf{Attack}(\mathcal{D}) \cap \mathsf{Elim}(\mathcal{D}) = \emptyset$.[8]

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

[6] In what follows we shall sometimes omit the last component of a tuple in case that it is the empty-set, and omit the set signs (the parentheses) in case that it is a singleton.

[7] 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 $\mathcal{S} = \{p\}$.

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

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

Algorithm 1. Evaluation of a simple 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 $\langle i, \overline{s}, \mathsf{J}, r\rangle$ in $\mathcal{D}$.[9]

Proposition 3.6 *For every simple derivation $\mathcal{D}$ the set $\mathsf{Accept}(\mathcal{D})$ is conflict-free in $\mathcal{AF}(\mathcal{D})$. If $\mathcal{D}$ is coherent, $\mathsf{Accept}(\mathcal{D})$ is a stable extension of $\mathcal{AF}(\mathcal{D})$.*[10]

Note 2 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 $\mathcal{E} = \mathsf{Accept}(\mathcal{D}')$.*

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 $\langle T_1, \ldots, T_n\rangle$ 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 $\langle T_1, \ldots, T_n\rangle$ of eliminating tuples (of the form $\langle i, \overline{s}, \mathsf{J}, r\rangle$), such that:

[9] 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.

[10] Due to lack of space proofs are omitted and will appear in an extended version of this paper.

(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 prerequisit conditions in $\mathcal{D}$. [11]

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.*

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 $\mathcal{L}$. 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.

[11] Conditions (i) and (ii) assure that these are sound attacks: by coherence neither of the attacking sequents of the additional elimination tuples is in $\mathsf{Elim}(\mathcal{D})$, and by Condition (ii) they are not attacked by an accepted $\mathcal{S}$-based sequent. As we show below (see Footnote 16), these two conditions are not dependent.

Example 4.1 Consider the argumentation setting $\mathfrak{S} = \langle \mathsf{CL}, LK, \text{Ucut}\rangle$, based on classical logic 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.	$p \vee \neg p \Rightarrow \neg(p \wedge \neg p)$	...
5.	$\neg(p \wedge \neg p) \Rightarrow p \vee \neg p$	...
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 ψ 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. [12] 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 $\neg p \Rightarrow \neg p$, [13] thus reintroducing $p \not\Rightarrow p$, and so forth. As a consequence, 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 $\mathcal{S}_1 \not\mathrel{|\!\sim} p$ (respectively, $\mathcal{S}_1 \not\mathrel{|\!\sim} \neg p$).

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: $\mathfrak{S} = \langle \mathsf{CL}, LK, \text{Ucut}\rangle$, 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

[12] In is important to note that Ucut-attackers of $q \Rightarrow q$ like $p, \neg p, q \Rightarrow \neg q$ may still be derived in an extension of $\mathcal{D}$, however, they cannot be used for eliminating $q \Rightarrow q$. Any attempt to introduce an eliminating tuple with $q \not\Rightarrow q$ will fail due to Condition (ii) in Definition 3.8(c) because, as noted above, the attacker of $q \Rightarrow q$ is counter-attacked by the sequent $\Rightarrow p \vee \neg p$ in Tuple 3 of $\mathcal{D}$.

[13] Alternatively, $p \Rightarrow p$ can be re-attacked by any sequent of the form $\neg p \Rightarrow \psi$, where ψ is equivalent to $\neg p$ (for instance, $\psi = \neg^i p$, where $\neg^n p$ denotes p preceded by n negations and i is some odd number).

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 $\mathsf{Arg}_{\mathsf{CL}}(\mathcal{S}'_1)$. [14]

This example shows in particular that $\mid\!\sim_{\mathfrak{S}}$ is sensitive to the syntactic form of the premises: although $\mathcal{S}_1$ and $\mathcal{S}'_1$ are CL-equivalent, their $\mathfrak{S}$-conclusions are not the same. In our case 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 not unusual in non-monotonic reasoning and this what one expects when, e.g., maximally consistent subsets of premises are taken into account (see [14]). [15]

Example 4.3 Consider a logic with a negation $\neg$ (i.e., $p \not\vdash \neg p$ 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 $s_1 \in \mathsf{Attack}(\mathcal{D}') \cap \mathsf{Elim}(\mathcal{D}')$. [16] 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.

[14] 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.

[15] Syntax dependency ceases to hold when $\mathcal{S}_1$ (or $\mathcal{S}'_1$) is consistent. This follows from Proposition 5.2 below.

[16] This shows, in particular, that the two conditions in Definition 3.8(c) are not dependent.

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 $\mathsf{Arg}_{\mathfrak{L}}(\mathcal{S}_2)$. It follows that s_4 is finally derived by the above derivation, and so $\mathcal{S}_2 \mid\!\sim \neg^4 p$. 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 $\mathcal{S}_2 \not\mid\!\sim \neg^3 p$. In turn, since the attacker (s_3) of s_2 is eliminated and cannot be recovered, s_2 is finally derived, thus $\mathcal{S}_2 \mid\!\sim \neg\neg p$. Similar considerations show that in this case $\mathcal{S}_2 \not\mid\!\sim \neg p$ and that $\mathcal{S}_2 \mid\!\sim p$.

Note 4 The last example emphasizes the basic difference between the derivation process introduced here and the one considered in [2]. While the process in [2] 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 [2] 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 [2] 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 [8]): 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 $\mid\!\sim$

In this section we consider some properties of the entailment relations that are induced by dynamic proof systems according to Definition 3.12.

Relations between $\mid\!\sim$ 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 $\mid\!\sim$ 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 $\mid\!\sim$ and $\vdash$ coincide.*

(ii) Another case where $\mid\!\sim$ 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 $\mathcal{S} \mid\!\sim \psi$ iff $\mathcal{S} \vdash \psi$.*

(iii) In general, $\mid\!\sim$ is weaker than $\vdash$.

Proposition 5.3 *If $\mathcal{S} \mid\!\sim \psi$ 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 θ 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 $\mid\!\sim \psi$.*

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

Note 5 Some properties of the base logic are 'inherited' by $\mid\!\sim$. One of them is $\neg$-paraconsistency [7]:

Proposition 5.7 *If $\vdash$ is $\neg$-paraconsistent (that is, there are atoms p, q such that $p, \neg p \not\vdash q$) then so is $\mid\!\sim$.*

Cautious Reflexivity

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

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

Corollary 5.9

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

(ii) *For every atom p it holds that $p \mid\!\sim p$.*

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

Restricted Monotonicity

Clearly, $\mid\!\sim$ is not monotonic. For instance, by Corollary 5.9 $p \mid\!\sim p$ while Example 4.1 shows a case in which $p, \neg p, q \not\mid\!\sim p$. 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 ψ 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 ψ, 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 formulas (in $\mathcal{L}$) is called $\vdash$-*consistent*, if there exists a formula ψ (in $\mathcal{L}$) 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 [11], a finitary propositional logic $\langle \mathcal{L}, \vdash \rangle$ is uniform iff it has a single characteristic matrix (see also [17]). Thus, classical logic as well as many other logics are uniform.

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

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 Γ' contains Γ.

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.

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 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.

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

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

[5] 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.

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

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

[8] 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.

[9] 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.

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

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

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

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

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

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

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

[17] 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.

A Proofs (for the extended version)

Proof of Proposition 3.6. If $\mathsf{Accept}(\mathcal{D})$ is not conflict-free in $\mathcal{AF}(\mathcal{D})$ then there are $s, t \in \mathsf{Accept}(\mathcal{D})$ such that $\langle i, \overline{s}, \mathsf{J}, t\rangle \in \mathcal{D}$ for some $i \in \mathbb{N}$ and some justification J. Since t is accepted, it is not eliminated, and so by the evaluation algorithm s is eliminated, in a contradiction to the assumption that s is also accepted.

Suppose now that $\mathcal{D}$ is coherent. We have to show that $\mathsf{Accept}(\mathcal{D})$ is admissible, complete, and stable (that is, $\mathsf{Accept}(\mathcal{D}) \cup \mathsf{Accept}(\mathcal{D})^+ = \mathsf{Derived}(\mathcal{D})$).

(I) $\mathsf{Accept}(\mathcal{D})$ **is admissible:** Suppose that there is some $s \in \mathsf{Accept}(\mathcal{D})$ that is attacked by t, i.e., there is $T_i = \langle i, \overline{s}, \mathsf{J}, t\rangle \in \mathcal{D}$. Since $s \notin \mathsf{Elim}(\mathcal{D})$ (because it is accepted), t must be in $\mathsf{Elim}(\mathcal{D})$. This means that there is some $j > i$ such that $T_j = \langle j, \overline{t}, \mathsf{J}, r\rangle \in \mathcal{D}$ for some $r \notin \mathsf{Elim}(\mathcal{D})$. It follows that $r \in \mathsf{Attack}(\mathcal{D})$ and since $\mathcal{D}$ is coherent, r is not eliminated later on (i.e., in the remaining j iterations of the evaluation algorithm). Thus, $r \in \mathsf{Accept}(\mathcal{D})$, which means that the attacker (t) of s is attacked by an element (r) of $\mathsf{Accept}(\mathcal{D})$. Thus $s \in \mathsf{Def}(\mathsf{Accept}(\mathcal{D}))$, and so $\mathsf{Accept}(\mathcal{D}) \subseteq \mathsf{Def}(\mathsf{Accept}(\mathcal{D}))$.

(II) $\mathsf{Accept}(\mathcal{D})$ **is complete:** Suppose that $s \in \mathsf{Def}(\mathsf{Accept}(\mathcal{D}))$. Then for every proof tuple $T_i = \langle i, \overline{s}, \mathsf{J}, t\rangle \in \mathcal{D}$ there is a proof tuple $T_j = \langle j, \overline{t}, \mathsf{J}, r\rangle \in \mathcal{D}$ and $r \in \mathsf{Accept}(\mathcal{D})$. Now,

- If $j > i$ then since $r \notin \mathsf{Elim}(\mathcal{D})$ we have that $t \in \mathsf{Elim}(\mathcal{D})$, and so s is not eliminated.
- If $i > j$ then either $t \in \mathsf{Elim}(\mathcal{D})$ and again s is not eliminated, or $t \notin \mathsf{Elim}(\mathcal{D})$, thus $t \in \mathsf{Attack}(\mathcal{D})$ (because of T_i) and $t \in \mathsf{Elim}(\mathcal{D})$ (because of T_j), and so $\mathcal{D}$ is not coherent, in contradiction to our assumption.

By the two items above, if s is attacked in $\mathcal{D}$, its attacker must be in $\mathsf{Elim}(\mathcal{D})$, and so $s \in \mathsf{Accept}(\mathcal{D})$. Thus $\mathsf{Def}(\mathsf{Accept}(\mathcal{D})) \subseteq \mathsf{Accept}(\mathcal{D})$ and by admissibility, $\mathsf{Def}(\mathsf{Accept}(\mathcal{D})) = \mathsf{Accept}(\mathcal{D})$.

(II) $\mathsf{Accept}(\mathcal{D})$ **is stable:** In other words, $\mathsf{Accept}(\mathcal{D}) \cup \mathsf{Accept}(\mathcal{D})^+ = \mathsf{Derived}(\mathcal{D})$. Since $\mathsf{Derived}(\mathcal{D}) = \mathsf{Accept}(\mathcal{D}) \cup \mathsf{Elim}(\mathcal{D})$, it is enough to show that $\mathsf{Accept}(\mathcal{D})^+$ coincides with $\mathsf{Elim}(\mathcal{D})$. Indeed,

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

Proof of Proposition 3.7. Let $\mathcal{D}$ be a simple derivation and $\mathcal{E}$ a stable extension of the sequent-based argumentation framework $\mathcal{AF}(\mathcal{D}) = \langle \mathit{Derived}(\mathcal{D}), \mathit{Attack}(\mathcal{D})\rangle$ that is induced by $\mathcal{D}$. Consider a simple derivation $\mathcal{D}'$ which is a concatenation of the following sequences $\mathcal{D}'_1 \oplus \mathcal{D}'_2 \oplus \mathcal{D}'_3$, where $\mathcal{D}'_1$ contains the tuples introducing the sequents in $\mathit{Derived}(\mathcal{D})$, $\mathcal{D}'_3$ 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 $\mathit{Attack}(\mathcal{D})$ (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, $\mathcal{E}^+ = \mathit{Derived}(\mathcal{D}) \setminus \mathcal{E} = \mathit{Derived}(\mathcal{D}') \setminus \mathcal{E}$, and so when the algorithm completes its pass over $\mathcal{D}'_3$ it holds that $\mathsf{Attack} = \mathcal{E}$ and $\mathsf{Elim} = \mathit{Derived}(\mathcal{D}') \setminus \mathcal{E}$. 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 $\mathsf{Accept}(\mathcal{D}') = \mathit{Derived}(\mathcal{D}) \setminus \mathsf{Elim} = \mathcal{E}$. □

Proof of Proposition 3.11. Suppose that s is finally derived in $\mathcal{D}$ but it is not finally derived in some extension $\mathcal{D}'$ of $\mathcal{D}$. This means that there is some extension $\mathcal{D}''$ of $\mathcal{D}'$ in which $s \in \mathsf{Elim}(\mathcal{D}'')$. Since $\mathcal{D}''$ is also an extension of $\mathcal{D}$, we get a contradiction to the final derivability of s in $\mathcal{D}$. □

Proof of Proposition 5.1. If there are no attack rules, dynamic derivations are in fact standard $\mathfrak{C}$-proofs in which every derived sequent is finally derived. Thus, $\mathcal{S} \mid\!\sim_{\mathfrak{S}} \psi$ iff there is a derivation of $\Gamma \Rightarrow \psi$ in $\mathfrak{C}$ for some finite $\Gamma \subseteq \mathcal{S}$. Since $\mathfrak{C}$ is sound and complete for $\mathfrak{L}$, the latter is a necessary and sufficient condition for $\mathcal{S} \vdash \psi$. □

Proof of Proposition 5.2. If there are no attacks between arguments in $\mathsf{Arg}_{\mathfrak{L}}(\mathcal{S})$ then no attack rule in $\mathfrak{A}$ is applicable, and so the proof is similar to that of Proposition 5.1. □

Proof of Proposition 5.3. If $\mathcal{S} \mid\!\sim \psi$ then there is an $\mathcal{S}$-based dynamic derivation for $\mathfrak{S}$, in which $\Gamma \Rightarrow \psi$ is finally derived for some finite $\Gamma \subseteq \mathcal{S}$. In particular, there is a proof in $\mathfrak{C}$ for $\Gamma \Rightarrow \psi$. Since $\mathfrak{C}$ is complete for $\mathfrak{L}$, this implies that $\Gamma \vdash \psi$, and by the monotonicity of $\mathfrak{L}$ we have that $\mathcal{S} \vdash \psi$. □

Proof of Proposition 5.5. If $\vdash \psi$ then the sequent $\Rightarrow \psi$ is provable in $\mathfrak{C}$. Since there are only theorem-preserving rules in $\mathfrak{A}$, this sequent cannot be attacked, and so any $\mathfrak{C}$-proof of $\Rightarrow \psi$ is also a dynamic derivation for $\mathfrak{S}$, in which $\Rightarrow \psi$ is finally derived. □

Proof of Corollary 5.6. By Propositions 5.3 and 5.5. □

Proof of Proposition 5.7. Since $\vdash$ is paraconsistent, $p, \neg p \nvdash q$. Thus, by Proposition 5.3, $p, \neg p \not\mid\!\sim q$. □

Proof of Proposition 5.8. This is a direct corollary of Proposition 5.2 and the fact that $\mathcal{S} \vdash \psi$ for every $\psi \in \mathcal{S}$ (since $\vdash$ is reflexive). □

Proof of Proposition 5.12. Suppose that $\mathcal{S}_1 \mid\!\sim \psi$. Then there is a dynamic derivation $\mathcal{D}^{\psi}_{\mathcal{S}_1}$, in which for some finite subset Γ of $\mathcal{S}_1$ the sequent $\Gamma \Rightarrow \psi$ is derived, (say, in step i of the derivation) and there is no dynamic derivation for $\mathfrak{S}$ that is based on $\mathcal{S}_1$, which extends $\mathcal{D}^{\psi}_{\mathcal{S}_1}$, and in which $\Gamma \Rightarrow \psi$ is attacked. Suppose for a contradiction that $\mathcal{S}_1, \mathcal{S}_2 \not\mid\!\sim \psi$. Since $\mathcal{D}^{\psi}_{\mathcal{S}_1}$ may be viewed as a dynamic derivation for $\mathfrak{S}$ that is based on $\mathcal{S}_1 \cup \mathcal{S}_2$, there must be a dynamic derivation $\mathcal{D}^{\psi}_{\mathcal{S}_1 \cup \mathcal{S}_2}$ for $\mathfrak{S}$ that is based on $\mathcal{S}_1 \cup \mathcal{S}_2$, and which extends $D^{\psi}_{\mathcal{S}_1}$ so that $\Gamma \Rightarrow \psi$ is attacked. It follows that for some $j > i$ there is a tuple $T_j = \langle j, \Gamma \not\Rightarrow \psi, J_j, \Theta \Rightarrow \phi \rangle$ in $\mathcal{D}^{\psi}_{\mathcal{S}_1 \cup \mathcal{S}_2}$

that is obtained by an application of Undercut for some finite subset Θ of $\mathcal{S}_1 \cup \mathcal{S}_2$, such that

1. $\Theta \Rightarrow \phi$ is $\mathfrak{C}$-provable from $\mathcal{S}_1 \cup \mathcal{S}_2$, and
2. $\phi \Rightarrow \neg\bigwedge\Gamma'$ and $\neg\bigwedge\Gamma' \Rightarrow \phi$ are $\mathfrak{C}$-provable for some $\Gamma' \subseteq \Gamma$.

The two items above, together with the completeness of $\mathfrak{C}$, imply that $\Theta \vdash \phi$ and $\phi \vdash \neg\bigwedge\Gamma'$. By the transitivity of $\vdash$, then,

($\star$) $\Theta \vdash \neg\bigwedge\Gamma'$.

Now, $\mathcal{S}_2$ is a $\vdash$-consistent set of formulas that is irrelevant to $\mathcal{S}_1 \cup \{\neg\bigwedge\Gamma'\}$ (This follows from the fact that $\mathsf{Atoms}(\Gamma') \subseteq \mathsf{Atoms}(\Gamma) \subseteq \mathsf{Atoms}(\mathcal{S}_1)$ and since $\mathsf{Atoms}(\mathcal{S}_1) \cap \mathsf{Atoms}(\mathcal{S}_2) = \emptyset$). In particular, then, $\Theta \cap \mathcal{S}_2$ is (a $\vdash$-consistent set of formulas that is) irrelevant to $\mathcal{S}_1 \cup \{\neg\bigwedge\Gamma'\}$, and so $\Theta \cap \mathcal{S}_2$ is irrelevant to $(\Theta \setminus \mathcal{S}_2) \cup \{\neg\bigwedge\Gamma'\}$. By the uniformity of $\mathfrak{L}$ we thus infer from ($\star$) that $\Theta \setminus \mathcal{S}_2 \vdash \neg\bigwedge\Gamma'$. By Item (2) and the completeness of $\mathfrak{C}$ again, $\neg\bigwedge\Gamma' \vdash \phi$. By transitivity, we have $\Theta \setminus \mathcal{S}_2 \vdash \phi$, and since $\Theta \setminus \mathcal{S}_2 \subseteq \mathcal{S}_1$ we get that:

3. $(\Theta \setminus \mathcal{S}_2) \Rightarrow \phi$ is $\mathfrak{C}$-provable from $\mathcal{S}_1$.

Now, Items (2) and (3) above imply that there is a dynamic derivation $\mathcal{D}$ for $\mathfrak{S}$ that is based on $\mathcal{S}_1$ and which extends $\mathcal{D}^{\psi}_{\mathcal{S}_1}$ by $\mathfrak{C}$-proofs of $\phi \Rightarrow \neg\bigwedge\Gamma'$, $\neg\bigwedge\Gamma' \Rightarrow \phi$, and $(\Theta \setminus \mathcal{S}_2) \Rightarrow \phi$.[17] Moreover, $\Gamma \not\Rightarrow \psi$ is derivable (in a derivation that is based on $\mathcal{S}_1$) by an application of Undercut using the last three sequents, but this contradicts the assumption that $\Gamma \Rightarrow \psi$ is finally derived in $\mathcal{D}^{\psi}_{\mathcal{S}_1}$. □

Note 10 To see that the property considered in Note 8 is indeed needed for the proof above, we show below that when Undercut is replaced by Canonical Undercut (who does *not* have the property of Note 8), the corresponding revised version of Proposition 5.12 cease to hold.

Indeed, let $\mathfrak{L} = \langle \mathcal{L}, \vdash \rangle$ be a logic with a $\vdash$-conjunction $\wedge$ and a $\vdash$-negation $\neg$, satisfying the following rules:

$$\text{(A.1)} \qquad \frac{\Gamma \Rightarrow \neg\psi, \neg\phi}{\Gamma \Rightarrow \neg(\psi \wedge \phi)} \qquad \frac{\Gamma \Rightarrow \neg(\psi \wedge \phi)}{\Gamma \Rightarrow \neg\psi, \neg\phi}.$$

We denote by $\mid\!\sim$ the entailment relation (according to Definition 3.12) for an argumentation setting whose base logic is $\mathfrak{L}$ and whose sole attack rule is C-Ucut.

Let now $\mathcal{S}_1 = \{p_1, \neg p_1, \neg\neg p_1\}$ and $\mathcal{S}_2 = \{p_i \mid i \geq 2\}$. Clearly, $\mathcal{S}_2$ is $\vdash$-consistent and irrelevant to $\mathcal{S}_1$. Yet, we show that: (a) $\mathcal{S}_1 \mid\!\sim p_1$ while: (b) $\mathcal{S}_1, \mathcal{S}_2 \not\mid\!\sim p_1$.

A derivation for proving Claim (a) may be the following:

1.	$p_1 \Rightarrow p_1$	Axiom	
2.	$\neg p_1 \Rightarrow \neg p_1$	Axiom	
3.	$\neg\neg p_1 \Rightarrow \neg\neg p_1$	Axiom	
4.	$\neg p_1 \not\Rightarrow \neg p_1$	C-Ucut, 3, 3, 3, 2	$\neg\neg p_1 \Rightarrow \neg\neg p_1$

[17] To see that the extension $\mathcal{D}$ is a valid derivation note that $\mathcal{D}$ extends a valid derivation only by introducing proof tuples whose derived sequents are not eliminated. Indeed, (the proofs of) $\phi \Rightarrow \neg\bigwedge\Gamma'$ and $\neg\bigwedge\Gamma' \Rightarrow \phi$ can be added to $\mathcal{D}^{\psi}_{\mathcal{S}_1}$, otherwise the attack in T_j would not be applicable (see Item 2), and (the proof of) $(\Theta \setminus \mathcal{S}_2) \Rightarrow \phi$ can be added otherwise according to Note 8 the sequent $\Theta \Rightarrow \phi$ would be Ucut-attacked as well, and again T_j would not be applicable, in contrast to our assumption.

The only potential attacker of Tuple 1 is Tuple 2, but the latter is eliminated and there is no way to attack its attacker, Tuple 3. Thus $p_1 \Rightarrow p_1$ is finally derived here.

To see Claim (b), note that once $\mathcal{S}_2$ is available, we can for instance extend the previous derivation by:

5.	$\neg p_1, p_2 \Rightarrow \neg p_1$	Weakening, 2	
6.	$p_1 \not\Rightarrow p_1$	C-Ucut, 5, 2, 2, 1	$\neg p_1, p_2 \Rightarrow \neg p_1$ [18]

Tuple 1 can still be defended by eliminating $\neg p_1, p_2 \Rightarrow \neg p_1$ (using the rules A.1 above), but then it may be re-attacked, e.g., by $\neg p_1, p_3 \Rightarrow \neg p_1$ (a weakening of Tuple 2), and so on. It follows that Tuple 1 is never finally derived. A similar argument applies to other ways of deriving p_1, such as by using $p_1, p_2 \Rightarrow p_1$.

Proof of Note 9. Similar to that of Proposition 5.12. Using the notation of that proof, in case of Direct Undercut the set Γ' should be replaced by a formula $\gamma \in \Gamma$ and in the case of Rebuttal Γ' is replaced by ψ'. □

[18] Note that with Ucut the extension of Lines 1–5 by Line 6 would not be allowed, since Tuple 5 is Ucut-attacked by Tuple 3.