Annotated Sequent Calculi for Paraconsistent Reasoning and Their Relations to Logical Argumentation

Ofer Arieli[1] , **Kees van Berkel**[2] and **Christian Straßer**[3]
[1]School of Computer Science, Tel-Aviv Academic College, Israel
[2]Institute of Logic and Computation, TU Wien, Austria
[3]Institute for Philosophy II, Ruhr University Bochum, Germany
oarieli@mta.ac.il, kees@logic.at, christian.strasser@rub.de

Abstract

We introduce *annotated sequent calculi*, which are extensions of standard sequent calculi, where sequents are combined with annotations that represent their derivation statuses. Unlike in ordinary calculi, sequents that are derived in annotated calculi may still be retracted in the presence of conflicting sequents, thus inferences are made under stricter conditions. Conflicts in the resulting systems are handled like in adaptive logics and argumentation theory. The outcome is a robust family of proof systems for non-monotonic reasoning with inconsistent information, where revision considerations are fully integrated into the object level of the proofs. These systems are shown to be strongly connected to logical argumentation.

1 Introduction

In this paper we introduce a proof-theoretic approach for non-monotonic reasoning with conflicting information. It consists of analytic sequent calculi [Gentzen, 1934] together with rules for modeling conflicts between sequents, which are similar to attack relations in argumentation theory [Dung, 1995].

The basic idea behind these calculi is to add annotations to sequents,[1] intuitively representing their statuses in a derivation. An annotated sequent is thus an expression of the form $\Gamma \Rightarrow^{[\mathsf{a}]} \Delta$, where $\Gamma \Rightarrow \Delta$ is an ordinary sequent and the superscript $[\mathsf{a}]$ is the annotation of the sequent. Informally, the annotation [i] means that the sequent is introduced (conditionally accepted), but is not yet inferred (finally accepted), because it may be attacked by a counter-sequent. The annotation [e] means that the sequent is eliminated, since it is attacked by an accepted sequent, and the annotation [!] is attached to finally accepted sequents, whose attackers are counter-attacked altogether.

A distinctive property of annotated sequent calculi is their modularity, namely: they can be based on any (propositional) logic with a sound and complete calculus, and any set of attack rules. In this paper, we demonstrate this novel approach by focusing on defeat attacks (and the corresponding reactivation and final acceptance rules described in what follows), mainly due to its commonness and simplicity. We show that the resulting proof systems are paraconsistent (in the sense that a contradictory set of premises does not have an explosive set of finally accepted conclusions), and faithfully represent the semantics of logical argumentation frameworks. The nature of derivations of annotated sequent calculi, and in particular the annotation revision process that proceeds the derivation steps (and that is expressed in the object level of the derivation) indicate that, despite of the relative simplicity of the calculi, they are particularly appropriate for modeling and describing inference processes involving revision of beliefs and defeasible reasoning.

2 Annotated Sequent Calculi

In the sequel, we denote by $\mathcal{L}$ an arbitrary propositional language. Sets of formulas are denoted by $\mathcal{S}, \mathcal{T}$, finite sets of formulas are denoted by Γ, Δ, formulas are denoted by ϕ, ψ, and atomic formulas are denoted by p, q, r (all possibly indexed). The set of (well-formed) formulas of $\mathcal{L}$ is denoted by $\mathsf{WFF}(\mathcal{L})$, and the power set of $\mathsf{WFF}(\mathcal{L})$ by $\wp(\mathsf{WFF}(\mathcal{L}))$.

A *propositional logic* is a pair $\mathfrak{L} = \langle \mathcal{L}, \vdash \rangle$, where $\mathcal{L}$ is a propositional language and $\vdash$ is a consequence relation on $\wp(\mathsf{WFF}(\mathcal{L})) \times \mathsf{WFF}(\mathcal{L})$, satisfying *reflexivity* (if $\phi \in \mathcal{S}$, then $\mathcal{S} \vdash \phi$), *monotonicity* (if $\mathcal{S}' \vdash \phi$ and $\mathcal{S}' \subseteq \mathcal{S}$, then $\mathcal{S} \vdash \phi$), and *transitivity* (if $\mathcal{S} \vdash \phi$ and $\mathcal{S}', \phi \vdash \psi$ then $\mathcal{S}, \mathcal{S}' \vdash \psi$). A logic $\mathfrak{L}$ is assumed to be *non-trivial* ($\mathcal{S} \nvdash \phi$ for some $\mathcal{S} \neq \emptyset$ and ϕ), *structural* (if $\mathcal{S} \vdash \phi$, then $\{\theta(\psi) \mid \psi \in \mathcal{S}\} \vdash \theta(\phi)$ for every substitution θ), and *compact* (if $\mathcal{S} \vdash \phi$ then $\Gamma \vdash \phi$ for some finite $\Gamma \subseteq \mathcal{S}$).

In what follows, we assume that $\mathcal{L}$ contains at least a $\vdash$-negation operator $\neg$, satisfying $p \nvdash \neg p$ and $\neg p \nvdash p$ (for atomic p), and a $\vdash$-conjunction operator $\wedge$, for which $\mathcal{S} \vdash \psi \wedge \phi$ iff $\mathcal{S} \vdash \psi$ and $\mathcal{S} \vdash \phi$.[2] Also, we denote by $\bigwedge\Gamma$ the conjunction of all the formulas in Γ.

Given a logic $\mathfrak{L}$ and a set $\mathcal{S}$ of $\mathcal{L}$-formulas, an $\mathfrak{L}$-*sequent based on* $\mathcal{S}$ [Gentzen, 1934] is a structure of the form $\Gamma \Rightarrow \Delta$, where $\Rightarrow$ is a symbol not in $\mathcal{L}$, and $\Gamma \vdash \bigwedge\Delta$ for some $\Gamma \subseteq \mathcal{S}$.

Definition 1. An *annotated* $\mathfrak{L}$-*sequent* (annotated sequent, for short) is a structure of the form $\Gamma \Rightarrow^{[\mathsf{a}]} \Delta$, where $\Gamma \Rightarrow \Delta$

[1]Note that unlike annotated logics [Abe *et al.*, 2019], we attach annotations to sequents instead of formulas in the language.

[2]By the definition of $\wedge$, we have that $\phi \wedge \psi \vdash \phi$; $\phi \wedge \psi \vdash \psi$, and $\phi, \psi \vdash \phi \wedge \psi$, thus $\mathcal{S}, \phi, \psi \vdash \sigma$ iff $\mathcal{S}, \phi \wedge \psi \vdash \sigma$.

is an $\mathfrak{L}$-sequent and $\mathsf{a} \in \{\mathsf{i}, \mathsf{e}, !\}$. We denote by $s[\mathsf{a}]$ the sequent s whose annotation is $[\mathsf{a}]$ and use $[*]$ when the annotation of the sequent may be arbitrary.

Next, we define an annotated sequent calculus $\mathfrak{C}$, extending a sequent calculus $\mathcal{C}$. We suppose that $\mathcal{C}$ is sound and complete (adequate) for $\mathfrak{L}$ (so $\Gamma \Rightarrow \Delta$ is $\mathcal{C}$-provable iff $\Gamma \vdash \bigwedge\Delta$).

Definition 2. An *annotated sequent calculus* $\mathfrak{C}$ consists of the following components:

1. **The axioms and inference rules of** $\mathcal{C}$, where the sequents in the rules' conditions are annotated by $[*]$ and the sequent in the conclusion is annotated by $[\mathsf{i}]$.
2. **Attack rules**. These are inference rules for changing the annotations of an attacked sequent from $[\mathsf{i}]$ to $[\mathsf{e}]$. To avoid arbitrary attacks, these rules are restricted to a set $\mathcal{S}$ of premises, thus the attacking and the attacked sequents must be $\mathcal{S}$-based. For instance, given such an $\mathcal{S}$, the rule Defeat looks as follows: for $\Gamma_1, \Gamma_1', \Gamma_2 \subseteq \mathcal{S}$,

$$\frac{\Gamma_1, \Gamma_1' \Rightarrow^{[\mathsf{i}]} \psi_1 \quad \Gamma_2 \Rightarrow^{[\mathsf{i}]} \psi_2 \quad \psi_2 \Rightarrow^{[*]} \neg \bigwedge\Gamma_1}{\Gamma_1, \Gamma_1' \Rightarrow^{[\mathsf{e}]} \psi_1}$$

   Here, the introduced sequent $\Gamma_2 \Rightarrow \psi_2$ attacks $\Gamma_1, \Gamma_1' \Rightarrow \psi_1$, and so changes the latter's status from 'introduced' to 'eliminated'. The status of the attack condition (rightmost) does not matter, as long as it is logically valid.[3]

   This attack rule is accompanied by a variation, in which $\Gamma_2 \Rightarrow^{[\mathsf{i}]} \psi_2$ is replaced by $\Gamma_2 \Rightarrow^{[!]} \psi_2$, i.e., attacking sequents are either accepted or finally accepted (as defined in Item 5).
3. **Reactivation rules.** These are inference rules for changing the annotations of a sequent that is to be reactivated from $[\mathsf{e}]$ to $[\mathsf{i}]$. Each attack rule has a corresponding reactivation rule. For instance, if Defeat is an attack rule, the following corresponding reactivation rule may be added to the calculus:

$$\frac{\Gamma_1, \Gamma_1' \Rightarrow^{[\mathsf{e}]} \psi_1 \quad \Gamma_2 \Rightarrow^{[\mathsf{e}]} \psi_2 \quad \psi_2 \Rightarrow^{[*]} \neg \bigwedge\Gamma_1}{\Gamma_1, \Gamma_1' \Rightarrow^{[\mathsf{i}]} \psi_1}$$

   This rule indicates that if the sequent in the first condition was attacked (due to the third condition, whose current status does not matter as as long as it is logically valid), but the attacker (the second sequent in the rule's condition) is later counter-attacked, then the originally attacked sequent changes its status from 'eliminated' back to 'introduced' (i.e., conditionally accepted).
4. **Retrospective attack rules.** Unlike attack rules, that allow only introduced attackers, retrospective attack rules allow also eliminated attackers, provided that the attackers can be reactivated.[4] These rules may thus be viewed as pairs of attack and reactivation rules. For instance, for $\Gamma_1, \Gamma_1', \Gamma_2, \Gamma_2', \Gamma_3 \subseteq \mathcal{S}$, one may define:

$$\frac{\Gamma_1, \Gamma_1' \Rightarrow^{[\mathsf{i}]} \psi_1 \quad \Gamma_2, \Gamma_2' \Rightarrow^{[\mathsf{e}]} \psi_2 \quad \psi_2 \Rightarrow^{[*]} \neg \bigwedge\Gamma_1}{\Gamma_1, \Gamma_1' \Rightarrow^{[\mathsf{e}]} \psi_1}$$

(attack rule with eliminated attacker)

$$\vdots$$

$$\frac{\Gamma_2, \Gamma_2' \Rightarrow^{[\mathsf{e}]} \psi_2 \quad \Gamma_3 \Rightarrow^{[\mathsf{e}]} \psi_3 \quad \psi_3 \Rightarrow^{[*]} \neg \bigwedge\Gamma_2}{\Gamma_2, \Gamma_2' \Rightarrow^{[\mathsf{i}]} \psi_2}$$

(the eliminated attacker is reactivated)

   Note that the rules above need not be consecutive in the derivation, but the reactivation of the attacker need to be part of the revision process following the attack rule (see Definition 3 below). A retrospective attack rule is only applicable in case that the revision process leads to a reactivation (see also the examples in Section 3).
5. **Final acceptability rules.** These are rules for assuring inferences of sequents. These rules depend on the attack rules and the sequent whose final acceptability is verified. The rules are relative to a *finite* set $\mathcal{S}$ of premises. For instance, let $\mathsf{Att}(\Gamma) = \{\Delta \subseteq \mathcal{S} \mid \Delta \vdash \neg\bigwedge\Gamma\}$ and suppose that Defeat is the attack rule, then:

$$\frac{\begin{array}{l}\Gamma \Rightarrow^{[\mathsf{i}]} \psi \\ (\forall \Delta \in \mathsf{Att}(\Gamma))\ \Delta \Rightarrow^{[*]} \neg\bigwedge\Gamma \\ (\forall \Delta \in \mathsf{Att}(\Gamma),\ \exists \Sigma \in \mathsf{Att}(\Delta))\ \Sigma \Rightarrow^{[!]} \neg\bigwedge\Delta\end{array}}{\Gamma \Rightarrow^{[!]} \psi}$$

   This final acceptability rule means that $\Gamma \Rightarrow \psi$ is finally accepted if it is conditionally accepted (the first condition of the rule), all of its attackers are produced in the derivation (this is the second condition of the rule),[5] and each such attacker is counter-attacked by a finally accepted sequent (the last condition of the rule).[6]

   Sequents that cannot be Defeat-attacked by any $\mathcal{S}$-based sequent, either since their left-hand side is empty these are tautological sequents), or because $\mathcal{S} \notin \mathsf{Att}(\Gamma)$, are accepted. Thus, when Defeat is the attack rule (or any other premise-attack rule), we also have:

$$\frac{\Rightarrow^{[\mathsf{i}]} \psi}{\Rightarrow^{[!]} \psi} \qquad \frac{\Gamma \Rightarrow^{[\mathsf{i}]} \psi \quad \mathcal{S} \notin \mathsf{Att}(\Gamma)}{\Gamma \Rightarrow^{[!]} \psi}$$

   Further (admissible) final derivability rules may be expressed. For instance, for premise-attack rules (e.g., Defeat), we may have rules expressing that if a sequent is finally derived, so is any sequent with a weaker support:

$$\frac{\Gamma, \Gamma' \Rightarrow^{[!]} \phi \quad \Gamma \Rightarrow^{[\mathsf{i}]} \psi}{\Gamma \Rightarrow^{[!]} \psi}$$

$$\frac{\Gamma_1 \Rightarrow^{[!]} \phi \quad \Gamma_1 \Rightarrow^{[\mathsf{i}]} \bigwedge\Gamma_2 \quad \Gamma_2 \Rightarrow^{[\mathsf{i}]} \psi}{\Gamma_2 \Rightarrow^{[!]} \psi}\ (\Gamma_2 \subseteq \mathcal{S})$$

[3] Undercut is a similar rule, where $\Gamma_1 \Rightarrow^{[*]} \neg\psi_2$ is added to the conditions of Defeat. In Undercut, the conclusion of the attacker is equivalent to the negation of some premises of the attacked sequent. Further sequent-based attack rules can be imported from, e.g., [Arieli and Straßer, 2019], using annotations similar to Defeat.

[4] Intuitively, retrospective attacks will deal with attack cycles.

[5] Alternatively, that all attackers are produced in a derivation, may be verified by a refutation calculus $\mathcal{R}$ (where $\Gamma \not\Rightarrow \psi$ is $\mathcal{R}$-provable iff $\Gamma \nvdash \psi$), like those in [Bonatti and Olivetti, 2002] (for classical logic) and [Pkhakadze and Tompits, 2020] (for L3).

[6] By monotonicity, instead of $\mathsf{Att}(\Gamma)$, it suffices to refer in the rule to $\mathsf{MinAtt}(\Gamma) = \{\Delta \in \mathsf{Att}(\Gamma) \mid (\forall \Delta' \subsetneq \Delta)\ \Delta' \notin \mathsf{Att}(\Gamma)\}$.

Definition 3. An $\mathcal{S}$-based *derivation* $\mathcal{D}$ of an annotated calculus is a sequence of tuples. Each tuple T contains an index (the tuple's order in the derivation), the derived annotated sequent (the tuple's sequent), the derivation rule that is applied (the tuple's rule), and the indexes of the tuples whose sequents serve as the conditions of the tuple's rule. The tuples' rules are those described in Definition 2, and some of them (like the attack rules) are $\mathcal{S}$-dependent. After each extension of the derivation with a tuple with an attack or a retrospective attack rule, an *annotation revision process* is initiated, and the derivation is extended with new attack or reactivation rules for updating the sequent annotations when necessary.

The annotation revision process. Let $p(s)$ be the plain sequent (without annotation) of the annotated sequent s. Also, if $s[\mathsf{a1}]$ is derived and at no later point $s[\mathsf{a2}]$ is derived, we say that the most updated status of s is $[\mathsf{a1}]$. Now, when a tuple T with a [retrospective] attack on s is introduced in the derivation, we let $\mathsf{RevSeq} = \{p(s)\}$[7] and the derivation sequence is traversed backwards, starting from the tuple before T.

• Suppose that during the traversal a tuple T of a [retrospective] attack is encountered, with attacker s_1 and attacked s_2, such that $p(s_1) \in \mathsf{RevSeq}$ while $p(s_2) \notin \mathsf{RevSeq}$.[8] Then:

- If the most updated status of s_1 is [e] (i.e., $p(s_1)$ was (counter-)attacked during the revision), T should be "rolled-back" by means of a reactivation rule for $p(s_2)$.[9] As a consequence, the most updated status of $p(s_2)$ in the derivation becomes [i].
- If the most updated status of s_1 is [i] (i.e,. $p(s_1)$ was reactivated during the revision), but still the most updated status of the attacked sequent $p(s_2)$ is also [i], T should be "reinstated" by means of an attack in which $p(s_1)$ re-attacks $p(s_2)$. As a consequence, the most updated status of $p(s_2)$ in the derivation becomes [e].

In both cases, $p(s_2)$ is added to RevSeq.

• Suppose that during the traversal a reactivating tuple T is encountered, where the condition of the reactivation is s_1 and the reactivated sequent is s_2, such that $p(s_1) \in \mathsf{RevSeq}$ and $p(s_2) \notin \mathsf{RevSeq}$. If the most updated status of s_1 is [i] (i.e, $p(s_1)$ has been reactivated during the traversal), T should be "rolled-back" and the original attack is reinstated in the derivation by re-applying the original attack rule on $p(s_2)$, using a new tuple T'. As a consequence, the most updated status of $p(s_2)$ becomes [e] and $p(s_2)$ is added to RevSeq.

Definition 4. Let $s[\mathsf{i}]$ be an attacking sequent of a derivation tuple T. Then T is *coherent*, if at the end of the revision process following the introduction of T, the most updated status of s is still [i].

A derivation is *coherent* if all its tuples are coherent, and there is no tuple such that the most updated statuses of both its attacking and attacked sequents are in $\{[!], [\mathsf{i}]\}$.

[7]The set RevSeq consists of the sequents whose annotation are revised during the traversal.

[8]That is, the status of the attacker has been modified, but the status of the attacked sequent has not been modified yet.

[9]Reactivation rules are applied only during a revision process.

Definition 5. For an annotated calculus $\mathfrak{C}$, we write $\mathcal{S} \mid\sim_{\mathfrak{C}} \psi$ (henceforth, $\mathfrak{C}$ is omitted) if there is an $\mathcal{S}$-based $\mathfrak{C}$-derivation $\mathcal{D}$ and a subset $\Gamma \subseteq \mathcal{S}$, such that $\Gamma \Rightarrow^{[!]} \psi$ is derived in $\mathcal{D}$ (i.e., the sequent $\Gamma \Rightarrow \psi$ is *finally accepted* in $\mathcal{D}$).

3 Some Examples and Properties

Below are some illustrations of derivations by annotated sequent calculi and some basic properties of the induced entailment relation, $\mid\sim$. In all of the examples we use the proof system in Definition 2 with Defeat as the sole attack rule (and the corresponding reactivation and final acceptance rules).[10]

Example 1. Consider the set of assumptions $\mathcal{S} = \{p, \neg p, q\}$ and let $\mathfrak{L}$ be classical logic. To see that $q \Rightarrow q$ is finally accepted from $\mathcal{S}$ (i.e., there is a derivation of $q \Rightarrow^{[!]} q$), note that $q \Rightarrow^{[\mathsf{i}]} q$ and $\Rightarrow^{[\mathsf{i}]} p \vee \neg p$ are derivable (e.g., by using Gentzen's LK). Moreover, by one of the final acceptability rules, $\Rightarrow p \vee \neg p$ is finally accepted (since its left-hand side is empty). Now, the $\mathcal{S}$-based attackers of $q \Rightarrow q$ are $p, \neg p \Rightarrow \neg q$ and $p, \neg p, q \Rightarrow \neg q$ (thus $\mathsf{Att}(q) = \{\{p, \neg p\}, \{p, \neg p, q\}\}$). These attackers are also derivable, but $\Rightarrow^{[!]} p \vee \neg p$ attacks both of them. For instance, we have

$$\frac{p, \neg p \Rightarrow^{[\mathsf{i}]} \neg q \qquad \Rightarrow^{[!]} p \vee \neg p \qquad p \vee \neg p \Rightarrow^{[\mathsf{i}]} \neg(p \wedge \neg p)}{p, \neg p \Rightarrow^{[\mathsf{e}]} \neg q}$$

By the final acceptability rule $q \Rightarrow^{[!]} q$ is derived, thus $\mathcal{S} \mid\sim q$.

The situation regarding the other formulas in $\mathcal{S}$ is different, and we have that $\mathcal{S} \not\mid\sim p$ and $\mathcal{S} \not\mid\sim \neg p$. Indeed, while both $p \Rightarrow^{[\mathsf{i}]} p$ and $\neg p \Rightarrow^{[\mathsf{i}]} \neg p$ are derivable, none of them is finally accepted. This may be explained by the fact that each one of them attacks the other, causing a revision of their statuses, and so there are no finally accepted sequents that can eliminate at once all of the attackers. Indeed, after

$$\frac{\neg p \Rightarrow^{[\mathsf{i}]} \neg p \qquad p \Rightarrow^{[\mathsf{i}]} p \qquad p \Rightarrow^{[\mathsf{i}]} \neg\neg p}{\neg p \Rightarrow^{[\mathsf{e}]} \neg p}$$

$p \Rightarrow p$ is accepted and $\neg p \Rightarrow \neg p$ is eliminated. By extending the derivation with the retrospective attack

$$\frac{p \Rightarrow^{[\mathsf{i}]} p \qquad \neg p \Rightarrow^{[\mathsf{e}]} \neg p \qquad \neg p \Rightarrow^{[\mathsf{i}]} \neg p}{p \Rightarrow^{[\mathsf{e}]} p}$$

$$\frac{\neg p \Rightarrow^{[\mathsf{e}]} \neg p \qquad p \Rightarrow^{[\mathsf{e}]} p \qquad p \Rightarrow^{[\mathsf{i}]} \neg\neg p}{\neg p \Rightarrow^{[\mathsf{i}]} \neg p}$$

the situation is reversed, and now $\neg p \Rightarrow \neg p$ is accepted while $p \Rightarrow p$ is eliminated. Another application of the retrospective attack rule, this time when $p \Rightarrow^{[\mathsf{e}]} p$ retrospectively attacks $\neg p \Rightarrow^{[\mathsf{i}]} \neg p$, reverses their statuses again, and so forth.

Example 2. The previous example demonstrates, e.g., a cyclic attack of size two ($p \Rightarrow p$ and $\neg p \Rightarrow \neg p$ attack each other). Figure 1 shows a cycle of size four. The left-hand side of the figure, Stage I, represents a 3-chain of attacks. In the middle of the figure (Stage II), the chain turns into a 4-cycle.

[10]To preserve readability of the examples, in what follows we only consider the sequents of tuples occurring in a derivation (complete tuples may be easily reconstructed).

Note that in order to close the cycle, a retrospective attack is needed. After the revision process (denoted by the dashed arrows) s_2 and s_4 are accepted, while s_1 and s_3 are eliminated. These statuses may be reversed by an application of another retrospective attack, as shown in Stage III of the figure. As a result, s_1 and s_3 are accepted while s_2 and s_4 are eliminated. The last two steps may be repeated interchangeably, revising each time the statuses of the sequents involved in the cycle of attacks. It follows that each of the four sequents is exposed to repeated attacks, and so none of them is finally accepted.

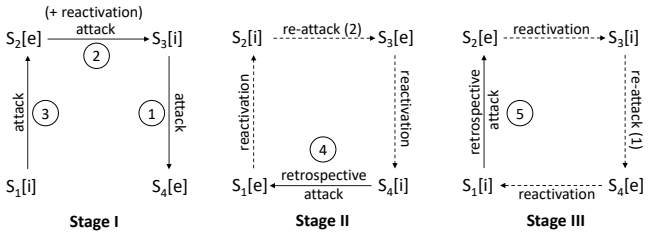


Figure 1: Derivations for an attack cycle of length four (Example 2). The numbers represent the order of the derivation steps. Strict arrows are applications of attack rules and dashed arrows denote rules applied in the corresponding annotation revision process.

A similar analysis holds for any even-length cycle of attacks, thus derivation tuples in even-length cycles are *coherent* (Definition 4). In contrast, derivations with an odd-length cyclic attack must be incoherent. See, e.g., derivation step 3 in Figure 2 Stage II, for a 3-cycle. Indeed, for 'closing' such a cycle with an attack rule the attacker must be eliminated at the end of the revision process (this may be verified by induction on the length of the cycle). Moreover, if a retrospective attack is initiated in such cases, the attacker cannot be reactivated.

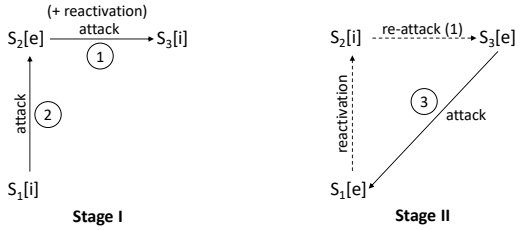


Figure 2: Derivations for an attack cycle of length three (Example 2).

Example 3. Consider a logic whose negation $\neg$ does not respect double-negation introduction (i.e., $p \not\vdash \neg\neg p$), and suppose again that Defeat is the only attack rule. Let $\mathcal{S} = \{p, \neg p, \neg\neg p, \neg\neg\neg p\}$. We denote by $\neg_i p$ the formula in which p is preceded by i negations (in particular, $\neg_0 p$ is p) and by $s_i[\mathsf{a}]$ the annotated sequent $\neg_i p \Rightarrow^{[\mathsf{a}]} \neg_i p$ for $\mathsf{a} \in \{\mathsf{i}, \mathsf{e}, !\}$. By reflexivity, $s_i[\mathsf{i}]$ is derivable for every $0 \leq i \leq 3$. Now, consider the following derivation (D and R denote Defeat and Reactivation, respectively).

$$\dfrac{s_0[\mathsf{i}] \qquad \dfrac{\dfrac{s_1[\mathsf{i}] \quad s_2[\mathsf{i}] \quad s_2[\mathsf{i}]}{s_1[\mathsf{e}]}\,\mathsf{D} \quad \dfrac{s_2[\mathsf{i}] \quad s_3[\mathsf{i}] \quad s_3[\mathsf{i}]}{s_2[\mathsf{e}]}\,\mathsf{D} \quad s_2[\mathsf{e}]}{s_1[\mathsf{i}]}\,\mathsf{R} \qquad s_1[\mathsf{i}]}{s_0[\mathsf{e}]}\,\mathsf{D}$$

At this point, the derived sequents and their most updated statuses are $s_0[\mathsf{e}]$, $s_1[\mathsf{i}]$, $s_2[\mathsf{e}]$ and $s_3[\mathsf{i}]$. Moreover, this is a kind of a 'steady state', in which all the sequents that are conditionally accepted (and only them) can in fact be finally accepted (note that all their $\mathcal{S}$-based attackers are derived). Indeed, since $\mathcal{S} \notin \mathsf{Att}(\neg_3 p)$, we have that $\mathsf{Att}(\neg_3 p) = \emptyset$ (i.e., s_3 cannot be attacked by any $\mathcal{S}$-based sequent), and so by the final acceptability rule we can derive $s_3[!]$. In turn, $s_3[!]$ attacks s_2 , the single attacker of s_1, thus $s_1[!]$ is derived. Finally, $s_1[!]$ attacks s_0 , and so the latter cannot be finally accepted. It follows, then, that $\mathcal{S} \mid\sim \neg p$ and $\mathcal{S} \mid\sim \neg\neg\neg p$.

Now, suppose that $\neg_4 p$ is added to the set of assertions: i.e., $\mathcal{S}' = \{p, \neg p, \neg\neg p, \neg\neg\neg p, \neg\neg\neg\neg p\}$. Then $s_4[\mathsf{i}]$ is derived, and so the previous derivation may be extended as follows:

$$\dfrac{s_0[\mathsf{e}] \qquad \dfrac{\dfrac{s_1[\mathsf{i}] \qquad \dfrac{\dfrac{s_2[\mathsf{e}] \qquad \dfrac{s_3[\mathsf{i}] \quad s_4[\mathsf{i}] \quad s_4[\mathsf{i}]}{s_3[\mathsf{e}]}\,\mathsf{D} \quad s_3[\mathsf{e}]}{s_2[\mathsf{i}]}\,\mathsf{R} \quad s_2[\mathsf{i}]}{s_1[\mathsf{e}]}\,\mathsf{D} \quad s_1[\mathsf{e}]}{s_0[\mathsf{i}]}\,\mathsf{R}$$

The derived sequents and their most updated statuses are now: $s_0[\mathsf{i}]$, $s_1[\mathsf{e}]$, $s_2[\mathsf{i}]$, $s_3[\mathsf{e}]$ and $s_4[\mathsf{i}]$. Again, for similar reasons as before, all the sequents that are conditionally accepted (and only them) are also finally accepted, so this time we conclude that $\mathcal{S}' \mid\sim p$ and $\mathcal{S}' \mid\sim \neg\neg p$ and $\mathcal{S}' \mid\sim \neg\neg\neg\neg p$.

An alternative representation of the derivation in this example is presented in Figure 3.

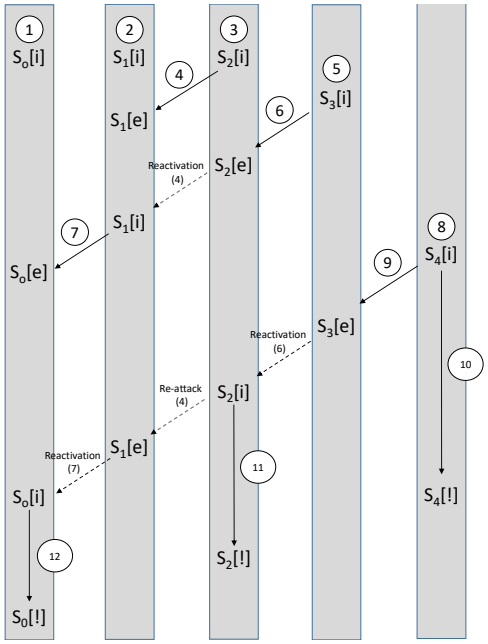


Figure 3: The derivation of Example 3 for $\mathcal{S}'$ (progressing along the vertical axis according to the circled numbers, which represent derivation steps). The solid arrows are applications of attack rules and the dashed arrows are the rules applied in the corresponding annotation revision process. The gray rectangles highlight the status changing of each sequent.

Next, we consider some basic properties of the entailment relation $\mid\!\sim$ that is induced by annotated calculi. We start with two observations about derivations in annotated calculi.

Proposition 1. *Let $\mathcal{D}$ be a derivation (Definition 3).*

a) *For a fixed set of assumptions, a finally accepted sequent cannot be eliminated.*[11]

b) *Suppose that $\mathcal{D}$ is coherent. Then at the end of a revision process (following a derivation step) no sequent attacks another sequent and is eliminated at the same time.*

Proof. Item (a) holds simply because a finally accepted sequent cannot be attacked. For item (b) suppose that s_1 attacks s_2. This may happen in one of the following two cases:

1. The attack is part of a derivation step. If this step is an application of an attack rule, then the status of s_1 is [i] (or [!]), and thus by coherence (or by the first item, in case that s_1 is finally derived), s_1 cannot be eliminated. The other possibility is that a retrospective attack rule is applied, in which case s_1 must be reactivated, turning its status back to [i] (and it cannot be modified again in the revision process, since s_1 is in RevSeq).
2. The attack is reinstated as part of the annotation revision process. This may happen only if s_1 was reactivated earlier in the revision process, so its status was changed to [i] (and cannot be changed again during the revision).

In both cases s_1 is not eliminated. □

Proposition 2. *If $\mathcal{S}$ is $\vdash$-consistent (i.e., $\mathcal{S} \nvdash \neg\psi$ for every $\psi \in \mathcal{S}$), then $\mathcal{S} \vdash \psi$ iff $\mathcal{S} \mid\!\sim \psi$.*[12]

Proof. If $\mathcal{S}$ is $\vdash$-consistent, no attack rule is applied, thus no sequent is eliminated (and so no reactivation or retrospective attack is applied either). It follows that in this case a derivation consists only of rules of $\mathcal{C}$ and the final acceptability rules. Moreover, for every $\Gamma \subseteq \mathcal{S}$ it holds that $\mathsf{Att}(\Gamma) = \emptyset$, thus any derived sequent is also finally accepted (and, ofcourse, any finally accepted sequent must be derived). It follows that $\mathcal{S} \vdash \psi$ iff $\mathcal{S} \Rightarrow^{[\mathsf{i}]} \psi$ is derived, iff $\mathcal{S} \Rightarrow \psi$ is finally accepted in that derivation, iff $\mathcal{S} \mid\!\sim \psi$. □

Proposition 3. *If $\vdash$ is paraconsistent (i.e., $p, \neg p \nvdash q$) or contrapositive (i.e., if $\Gamma \vdash \psi$ then $\Gamma, \neg\psi \vdash \neg\gamma$ for every $\gamma \in \Gamma$), then $\mid\!\sim$ is paraconsistent .*

Proof. If $\vdash$ is paraconsistent, then $p, \neg p \Rightarrow q$ is not derivable. Since neither $p \Rightarrow q$ nor $\neg p \Rightarrow q$ is derivable (the logic is not trivial), there is no derivable $\{p, \neg p\}$-based argument whose conclusion is q. Thus, no such sequent is finally derived, and so $p, \neg p \not\mid\!\sim q$. Suppose then that $\vdash$ is not paraconsistent. Then $p, \neg p \Rightarrow$ is derivable, and so by contraposition $\Rightarrow \neg(p \wedge \neg p)$ is also derivable. The latter is finally accepted (by the final acceptability rule), and moreover $\{p, \neg p\} \in \mathsf{Att}(q)$. Thus, even if $p, \neg p \Rightarrow q$ is derived, the last condition in the final acceptability rule is not met. So, $\Gamma \Rightarrow q$ is not finally derived for $\Gamma \subseteq \{p, \neg p\}$, thus $p, \neg p \not\mid\!\sim q$ in this case as well. □

[11] In other words, there is no extension of $\mathcal{D}$ (by further derivation steps and revisions), based on the same set of assumptions, in which the annotation of a sequent changes from [!] to [e]. This resembles the notion of final derivability in adaptive logics [Batens, 2007; Straßer, 2014] and in [Arieli and Straßer, 2019].

[12] Recall, $\vdash$ is the consequence relation of the base logic verifying sequents, and $\mid\!\sim$ is the entailment relation given by final acceptance.

4 Relations to Logical Argumentation

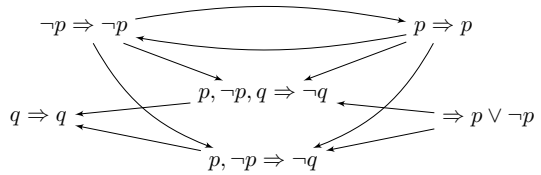
Annotated calculi, and in particular the attack rules, are inspired by similar concepts in formal argumentation [Dung, 1995; Baroni *et al.*, 2018; Gabbay *et al.*, 2021]. In this section, we show some relations between the two formalisms.

Definition 6. Let $\mathfrak{C}$ be an annotated sequent calculus and let $\mathcal{D}$ be an $\mathcal{S}$-based $\mathfrak{C}$-derivation. Then:

- $\mathsf{Derived}(\mathcal{D})$ is the set of $\mathcal{S}$-based sequents s s.t. $s[\mathsf{i}] \in \mathcal{D}$;
- $\mathsf{Accept}(\mathcal{D})$ is the set of sequents s in $\mathsf{Derived}(\mathcal{D})$ such that their most updated status is [i] or [!];
- $\mathsf{Final}(\mathcal{D})$ is the set of sequents s in $\mathsf{Derived}(\mathcal{D})$ such that $s[!] \in \mathcal{D}$;
- $\mathsf{Attack}(\mathcal{D})$ is the set of pairs (s_1, s_2) such that s_1 attacks or retrospectively attacks s_2 in $\mathcal{D}$ ($s_1, s_2 \in \mathsf{Derived}(\mathcal{D})$).

$\mathcal{AF}(\mathcal{D}) = \langle \mathsf{Derived}(\mathcal{D}), \mathsf{Attack}(\mathcal{D}) \rangle$ is called the *(sequent-based) argumentation framework that is induced by $\mathcal{D}$.*

Example 4. The possible attacks among the sequents in Example 1 and the argumentation framework induced by the corresponding derivation are the following:



Definition 6 is a logic-based representation of argumentation frameworks, which according to Dung [1995] are pairs $\mathcal{AF} = \langle \mathsf{Args}, \mathsf{Attack} \rangle$, where Args is a denumerable set of elements, called arguments, and Attack is a relation on $\mathsf{Args} \times \mathsf{Args}$, whose instances are called attacks.

Given a framework $\mathcal{AF}$, a key issue in its understanding is the question what combinations of arguments can collectively be accepted in $\mathcal{AF}$. This is determined as follows:

Definition 7. Let $\mathcal{AF} = \langle \mathsf{Args}, \mathsf{Attack} \rangle$ be an argumentation framework, and let $\mathcal{E} \subseteq \mathsf{Args}$.

- $\mathcal{E}$ *attacks* an argument A if there is an argument $B \in \mathcal{E}$ that attacks A (i.e., $(B, A) \in \mathsf{Attack}$). The set of arguments that are attacked by $\mathcal{E}$ is denoted by $\mathcal{E}^+$.
- $\mathcal{E}$ *defends* A if $\mathcal{E}$ attacks every argument that attacks A.
- $\mathcal{E}$ is *conflict-free* if it does not attack any of its elements (i.e., $\mathcal{E}^+ \cap \mathcal{E} = \emptyset$), $\mathcal{E}$ is *admissible* if it is conflict-free and defends all of its elements, and $\mathcal{E}$ is *complete* if it is admissible and contains all the arguments that it defends.
- $\mathcal{E}$ is a *stable extension* of $\mathcal{AF}$ if it is conflict-free and $\mathcal{E} \cup \mathcal{E}^+ = \mathsf{Args}$. $\mathcal{E}$ is a *grounded extension* of $\mathcal{AF}$ if it is $\subseteq$-minimal among the complete extensions of $\mathcal{AF}$.

The next results show the close relations between derivations in annotated calculi (Definition 2) and stable, respectively grounded semantics of argumentation frameworks.

Proposition 4. *If $\mathcal{D}$ is coherent, then* $\mathsf{Accept}(\mathcal{D})$ *is a stable extension of* $\mathcal{AF}(\mathcal{D})$.

Outline of proof. We show that $\mathsf{Accept}(\mathcal{D})$ is conflict-free in $\mathcal{AF}(\mathcal{D})$ and attacks any eliminated sequent. The former follows from the coherence of $\mathcal{D}$. To see the latter, let r be an eliminated sequent. If r is [retrospectively] attacked by s in a derivation step, then by coherence the status of s is [i], thus $s \in \mathsf{Accept}(\mathcal{D})$. If r is attacked by s during the status revision process, then it is easy to verify that the statuses of r and s must interchange. Since r is eliminated, $s \in \mathsf{Accept}(\mathcal{D})$. □

Note 1. The coherence requirement in Proposition 4 is necessary. Consider the derivation $\mathcal{D}$ in Figure 2. Step 3 in Stage II is not coherent, as the attacker s_3 is eliminated. Thus, $\mathsf{Accept}(\mathcal{D}) = \{s_2\}$, but $\mathcal{AF}(\mathcal{D})$ has no stable extensions.

Example 5. In Example 4, if the last [retrospective] attack of $p \Rightarrow p$ on $\neg p \Rightarrow \neg p$ is performed after the last [retrospective] attack of $\neg p \Rightarrow \neg p$ on $p \Rightarrow p$, then for the corresponding derivation $\mathcal{D}$ we have $\mathsf{Accept}(\mathcal{D}) = \{\Rightarrow p \vee \neg p, q \Rightarrow q, p \Rightarrow p\}$. This is indeed a stable extension of $\mathcal{AF}(\mathcal{D})$. If the mutual attacks of $p \Rightarrow p$ and $\neg p \Rightarrow \neg p$ are performed in a reversed order, then $\mathsf{Accept}(\mathcal{D}) = \{\Rightarrow p \vee \neg p,\ q \Rightarrow q,\ \neg p \Rightarrow \neg p\}$, which again is a stable extension of $\mathcal{AF}(\mathcal{D})$.

Definition 8. An $\mathcal{S}$-derivation $\mathcal{D}$ is *saturated* if $\mathsf{Final}(\mathcal{D})$ is exhaustive in $\mathcal{D}$: i.e., the final acceptability rules are applied to every derived sequent in $\mathcal{D}$ to which it can be applied.[13]

Proposition 5. *If $\mathcal{D}$ is saturated, then* $\mathsf{Final}(\mathcal{D})$ *is the (unique) grounded extension $\mathcal{E} \subseteq \mathsf{Derived}(\mathcal{D})$ of $\mathcal{AF}(\mathcal{D})$.*

Proof. We show that $\mathsf{Final}(\mathcal{D}) = \mathcal{E}$. Let $\langle s_1[!], \ldots, s_n[!] \rangle$ be the ordered set of all [!]-annotated sequents derived in $\mathcal{D}$, in the order in which they occur in $\mathcal{D}$.

"$\subseteq$". We prove inductively that $s_i[!] \in \mathcal{E}$ for $i = 1, \ldots, n$. *Base case*. Since $s_1[!]$ is derived by an application of a final applicability rule and it is the first sequent in $\mathcal{D}$ with this property, it has no attackers. Since $\mathcal{E}$ is complete, $s_1 \in \mathcal{E}$. *Inductive step*. Suppose the sequent $s_{i+1}[!]$ was derived by a final acceptability rule calling upon the [!]-annotated sequents $s_{j_1}[!], \ldots, s_{j_m}[!]$. Then $j_1, \ldots, j_m < i+1$ and by the inductive hypothesis, $s_{j_1}, \ldots, s_{j_m} \in \mathcal{E}$. Also, the sequents $s_{j_1}, \ldots, s_{j_m}$ attack all attackers of s_{i+1}, so s_{i+1} is defended by $\mathcal{E}$ and by the completeness of $\mathcal{E}$, $s_{i+1} \in \mathcal{E}$.

"$\supseteq$". We show that $\mathsf{Final}(\mathcal{D})$ is complete in $\mathcal{AF}(\mathcal{D})$. Since $\mathcal{E}$ is $\subseteq$-minimal complete, then $\mathcal{E} \subseteq \mathsf{Final}(\mathcal{D})$. We show conflict-freeness inductively by showing that for each $i = 1, \ldots, n$ there is no $k \leq i$ such that s_k attacks s_i. *Base case*. Trivial, since s_1 doesn't have attackers. *Inductive step*. Suppose there is a $k \leq i$ s.t. s_k attacks s_{i+1}. By the final acceptability rule there is a $j \leq i$ s.t. s_j attacks s_k, which contradicts the inductive hypothesis. Suppose now that s_{i+1} attacks itself. By the final acceptability rule there is a $k \leq i$ such that s_k attacks s_{i+1} which we have already excluded.

For admissibility, suppose that $s \in \mathsf{Derived}(\mathcal{D}) \setminus \mathsf{Final}(\mathcal{D})$ attacks some s_i. Then by the application of the final acceptability rule that produces $s_i[!] \in \mathcal{D}$, there is a $k < i$ such that s_k attacks s. For completeness, let $s \in \mathsf{Derived}(\mathcal{D})$ be defended by $\mathsf{Final}(\mathcal{D})$. Then we can apply final applicability rule to derive $s[!]$. Since $\mathcal{D}$ is saturated, $s[!] \in \mathsf{Final}(\mathcal{D})$. □

[13] For a finite $\mathcal{S}$ this is a decidable property.

Proposition 6. *For a derivation $\mathcal{D}$, let $\mathcal{S} = \bigcup\{\Delta \mid \Delta \Rightarrow \phi \in \mathcal{D}\}$. Let $\mathcal{AF}(\mathcal{S}) = \langle \mathsf{Args}(\mathcal{S}), \mathsf{Attack} \rangle$ where $\mathsf{Args}(\mathcal{S}) = \{\Gamma \Rightarrow \psi \mid \Gamma \subseteq \mathcal{S}$ and $\Gamma \vdash \psi\}$ and $\mathsf{Attack} = \{(s,t) \mid s = \Delta \Rightarrow \phi, t = \Gamma \Rightarrow \psi \in \mathsf{Args}$ and $\phi \vdash \neg\bigwedge\Gamma\}$. Let $\mathcal{E}$ be the grounded extension of $\mathcal{AF}(\mathcal{S})$. For every $s \in \mathcal{E}$, there is an $\mathcal{S}$-based derivation $\mathcal{D}'$ of $s[!]$ without attack rules.*

Proof. We note that for each $r \in \mathsf{Args}(\mathcal{S})$ there is an $\mathcal{S}$-based derivation $\mathcal{D}_r$ concluding $r[i]$. Since $\mathcal{D}$ is finite, $\mathcal{E} = \bigcup_{i \geq 1} \mathcal{E}_i$ where $\mathcal{E}_0 = \emptyset$ and $\mathcal{E}_{i+1} = \{s \in \mathsf{Args} \mid \mathcal{E}_i$ defends $s\}$ [Dung, 1995]. Let $s \in \mathcal{E}$. We prove by induction on i, that for each $s \in \mathcal{E}_i$ $(i = 1, \ldots, n)$ there is an $\mathcal{S}$-based derivation $\mathcal{D}'_s$ of $s[!]$ without attacks. *Base case.* Since $s \in \mathcal{E}_1$, s has no attackers. Let $\mathcal{D}'_s$ be the extension of $\mathcal{D}_s$ by an application of the final acceptibility rule that concludes $s[!]$. *Inductive step*. Let $s \in \mathcal{E}_{i+1}$. For each attacker r of s there is an $s' \in \bigcup_{j=1}^{i} \mathcal{E}_j$ that attacks r. By the IH there are derivations $\mathcal{D}'_{s'}$ for each such defending s'. We obtain $\mathcal{D}'_s$ by concatenating the proofs $\mathcal{D}_r$ of each attacker, the proofs $\mathcal{D}'_{s'}$ for each defender, and applying the final acceptability rule to conclude $s[!]$. □

5 Related Work and Conclusion

Annotated versions of sequent calculi have been introduced in, e.g., [Došen, 1985; Indrzejczak, 1997]. However, those calculi significantly deviate from ours and have different motivations and purposes (we discuss this in future work).

There are a variety of reasoning methods for abstract and structured (or, more specifically, logical) argumentation frameworks. We refer to [Cerutti *et al.*, 2017] and [Besnard *et al.*, 2020] for two extensive surveys on this subject. Most of the approaches mentioned in these review papers are based on CSP/SAT/ASP/QBF-solvers, and as such, the reasoning engine is encapsulated in the solvers and/or limited to specific base logics. Our approach extends standard sequent calculi using the same notions (sequents, inference rules, etc), augmented with primary concepts from argumentation theory for a better handling of conflicts among the sequents. Some other notions, such as final acceptability are borrowed from proof systems for adaptive logics [Batens, 2007; Straßer, 2014]. Altogether, this results in a modular paraconsistent method for reasoning with arguments based on different underlying logics and attack rules.

A different approach with a similar motivation is presented in [Arieli and Straßer, 2019]. The formalism introduced in this paper has some advantages: First, in our case all inference rules may be chained, so in particular conclusions of attack rules may serve as conditions of other rules. Second, rules for final acceptability are based on the available derivation and not on its potential extensions, as in [Arieli and Straßer, 2019], or in adaptive logics. Last, by including sequent annotations, the machinery for updating the statuses of derived sequents is included in the derivation itself and does not require an external evaluation procedure. By allowing inference rules to reason about the acceptability statuses of arguments, our approach integrates meta-argumentative reasoning [Jakobovits and Vermeir, 1999; Boella *et al.*, 2009].

Acknowledgements

Ofer Arieli is partially supported by the Israel Science Foundation (Grant No. 550/19). Kees van Berkel is supported by the projects WWTF MA16-028 and FWF W1255-N23.

References

[Abe *et al.*, 2019] Jair Minoro Abe, Kazumi Nakamatsu, and João Inácio da Silva Filho. Three decades of paraconsistent annotated logics: a review paper on some applications. In *Proceedings of the 23rd International Conference Knowledge-Based and Intelligent Information & Engineering Systems (KES-2019)*, volume 159 of *Procedia Computer Science*, pages 1175–1181. Elsevier, 2019.

[Arieli and Straßer, 2019] Ofer Arieli and Christian Straßer. Logical argumentation by dynamic proof systems. *Theoretical Computer Science*, 781:63–91, 2019.

[Baroni *et al.*, 2018] Pietro Baroni, Dov Gabbay, Massimiliano Giacomin, and Leon van der Torre, editors. *Handbook of Formal Argumentation*, volume I. College Publications, 2018.

[Batens, 2007] Diderik Batens. A universal logic approach to adaptive logics. *Logica universalis*, 1(1):221–242, 2007.

[Besnard *et al.*, 2020] Philippe Besnard, Claudette Cayrol, and Marie-Christine Lagasquie-Schiex. Logical theories and abstract argumentation: A survey of existing works. *Journal of Argument and Computation*, 11(1-2):41–102, 2020.

[Boella *et al.*, 2009] Guido Boella, Dov Gabbay, Leendert van der Torre, and Serena Villata. Meta-argumentation modelling I: Methodology and techniques. *Studia Logica*, 93(2–3):297–355, 2009.

[Bonatti and Olivetti, 2002] Piero Andrea Bonatti and Nicola Olivetti. Sequent calculi for propositional nonmonotonic logics. *ACM Transactions on Computational Logic*, 3(2):226–278, 2002.

[Cerutti *et al.*, 2017] Federico Cerutti, Sarah Alice Gaggl, Matthias Thimm, and Johannes Peter Wallner. Foundations of implementations for formal argumentation. *Journal of Applied Logics-IfCoLog Journal of Logics and their Applications*, 4(8):2623–2706, 2017.

[Došen, 1985] Kosta Došen. Sequent-systems for modal logic. *Journal of Symbolic Logic*, 50(1):149–168, 1985.

[Dung, 1995] Phan Minh Dung. On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games. *Artificial Intelligence*, 77(2):321–357, 1995.

[Gabbay *et al.*, 2021] Dov Gabbay, Massimiliano Giacomin, Guillermo R. Simari, and Matthias Thimm, editors. *Handbook of Formal Argumentation*, volume II. College Publications, 2021.

[Gentzen, 1934] Gerhard Gentzen. Untersuchungen über das logische Schließen I, II. *Mathematische Zeitschrift*, 39:176–210, 405–431, 1934.

[Indrzejczak, 1997] Andrzej Indrzejczak. Generalised sequent calculus for propositional modal logics. *Logica Trianguli*, 1:15–31, 1997.

[Jakobovits and Vermeir, 1999] Hadassa Jakobovits and Dirk Vermeir. Robust semantics for argumentation frameworks. *Journal of Logic and Computation*, 9(2):215–261, 1999.

[Pkhakadze and Tompits, 2020] Sopo Pkhakadze and Hans Tompits. Sequent-type calculi for three-valued and disjunctive default logic. *Axioms*, 9(3), 2020.

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