1

A Review of Argumentation Based on Deductive Arguments

PHILIPPE BESNARD, ANTHONY HUNTER

ABSTRACT. A deductive argument is a pair where the first item is a set of premises, the second item is a claim, and the premises entail the claim. This can be formalized by assuming a logical language for the premises and the claim, and logical entailment (or consequence relation) for showing that the claim follows from the premises. Examples of logics that can be used include classical logic, modal logic, description logic, temporal logic, and conditional logic. A counterargument for an argument A is an argument B where the claim of B contradicts the premises of A. Different choices of logic, and different choices for the precise definitions of argument and counterargument, give us a range of possibilities for formalizing deductive argumentation. Further options are available to us for choosing the arguments and counterarguments we put into an argument graph. If we are to construct an argument graph based on the arguments that can be constructed from a knowledgebase, then we can be exhaustive in including all arguments and counterarguments that can be constructed from the knowledgebase. But there are other options available to us. These include being selective in the arguments and counterargument we present according to a specified criterion. We consider some of the possibilities in this review and introduce properties and postulates for comparing proposals for deductive argumentation.

1 Introduction

In deductive reasoning, we start with some premises, and we derive a conclusion using one or more inference steps. Each inference step is infallible in the sense that it does not introduce uncertainty. In other words, if we accept the premises are valid, then we should accept that the intermediate conclusion of each inference step is valid, and therefore we should accept that the conclusion is valid. For example, if we accept that Philippe and Tony are having tea together in London is valid, then we should accept that Philippe is not in Toulouse (assuming the background knowledge that London and Toulouse are different places, and that nobody can be in different places at the same time). As another example, if we accept that Philippe and Tony are having an ice cream together in Toulouse is valid, then we should accept that Tony is not in London. Note, however, we do not need to believe or know that the premises are valid to apply deductive reasoning. Rather, deductive reasoning allows us to obtain conclusions that we can accept contingent on the validity of their premises. So for the first example above, the reader might not know whether or not Philippe and Tony are having tea together in London. However, the reader can accept that Philippe is not in Toulouse, contingent on the validity of these premises. Important alternatives to deductive reasoning in argumentation, include inductive reasoning, abductive reasoning, and analogical reasoning.

In this review, we assume that deductive reasoning is formalized by a monotonic logic. Each deductive argument is a pair where the first item is a set of premises that logically entails the second item according to the choice of monotonic logic. So we have a logical language to express the set of premises, and the claim, and we have a logical consequence relation to relate the premises to the claim.

Key benefits of deductive arguments include: (1) Explicit representation of the information used to support the claim of the argument; (2) Explicit representation of the claim of the argument; and (3) A simple and precise connection between the support and claim of the argument via the consequence relation. What a deductive argument does not provide is a specific proof of the claim from the premises. There may be more than one way of proving the claim from the premises, but the argument does not specify which is used. It is therefore indifferent to the proof used.

Deductive argumentation is formalized in terms of deductive arguments and counterarguments, and there are various choices for defining this [Besnard and Hunter, 2008]. Deductive argumentation offers a simple route to instantiating abstract argumentation which we will consider in this review paper. Perhaps the first paper to consider this is by Cayrol who instantiated Dung's proposal with deductive arguments based on classical logic [Cayrol, 1995].

In the rest of this review, we will investigate some of the choices we have for defining arguments and counterarguments, and for how they can be used in modelling argumentation. We will focus on three choices for base logic. These are (1) simple logic (which has a language of literals and rules of the form $\alpha_1 \wedge \ldots \wedge \alpha_n \rightarrow \beta$ where $\alpha_1, \ldots, \alpha_n, \beta$ are literals, and modus ponens is the only proof rule), (2) classical logic (propositional and first-order classical logic), and (3) A conditional logic. Then for instantiating argument graphs (i.e. for specifying what the arguments and attacks are in an argument graph), we will consider descriptive graphs and generated graphs defined informally as follows.

- **Descriptive graphs** Here we assume that the structure of the argument graph is given, and the task is to identify the premises and claim of each argument. Therefore the input is an abstract argument graph, and the output is an instantiated argument graph. This kind of task arises in many situations: For example, if we are listening to a debate, we hear the arguments exchanged, and we can construct the instantiated argument graph to reflect the debate[1].

[1]When we listen to a debate (or similarly, when we read an article that discusses some topic), we use natural language processing to identify arguments and counterarguments in spoken (or written) language. At a high level, this involves determining the words used (i.e.

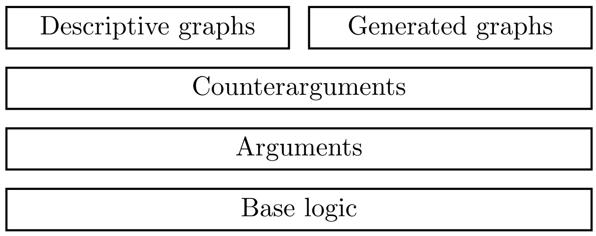


Figure 1. Framework for constructing argument graphs with deductive arguments: For defining a specific argumentation system, there are four levels for the specification: (1) A base logic is required for defining the logical language and the consequence or entailment relation (i.e. what inferences follow from a set of formlulae); (2) A definition of an argument $\langle \Phi, \alpha \rangle$ specified using the base logic (e.g. Φ is consistent, and Φ entails α); (3) A definition of counterargument specified using the base logic (i.e. a definition for when one argument attacks another); and (4) A definition of how the arguments and counterarguments are composed into an argument graph (which is either a descriptive graph or some form of generated graph).

- **Generated graphs** Here we assume that we start with a knowledgebase (i.e. a set of logical formulae), and the task is to generate the arguments and counterarguments (and hence the attacks between arguments). Therefore, the input is a knowledgebase, and the output is an instantiated argument graph. This kind of task also arises in many situations: For example, if we are making a decision based on conflicting information. We have various items of information that we represent by formulae in the knowledgebase, and we construct an instantiated argument graph to reflect the arguments and counterarguments that follow from that information.

For constructing both descriptive graphs and generated graphs, there may be a dynamic aspect to the process. For instance, when constructing descriptive graphs, we may be unsure of the exact structure of the argument graph, and it is only by instantiating individual arguments that we are able to say whether it is attacked or attacks another argument. As another example, when constructing generated graphs, we may be involved in a dialogue, and so through the dialogue, we may obtain further information which allows us to generate further arguments that can be added to the argument graph.

So in order to construct argument graphs with deductive arguments, we need to specify the choice of logic (which we call the base logic) that we use

the verbal description) for each argument and counterargument. This can be thought of as a form of argument mining. Then once the arguments and counterarguments have been identified, the actual logical structure of each argument can be determined from the verbal description.

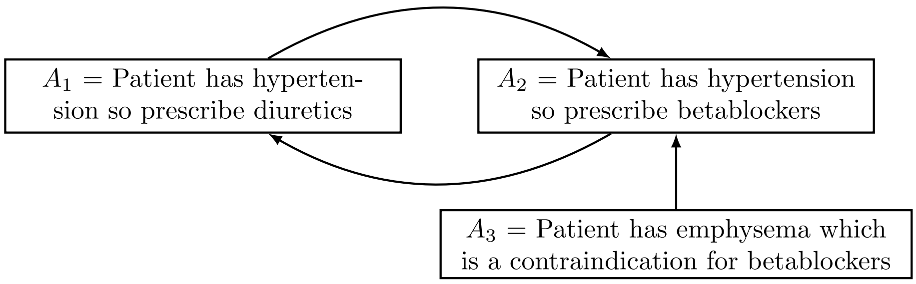


Figure 2. Example of an abstract argument graph which captures a decision making scenario where there are two alternatives for treating a patient, diuretics or betablockers. Since only one treatment should be given for the disorder, each argument attacks the other. There is also a reason to not give betablockers, as the patient has emphysema which is a contraindication for this treatment.

to define arguments and counterarguments, the definition for arguments, the definition for counterarguments, and the definition for instantiating argument graphs. For the latter, we can either produce a descriptive graph or a generated graph. We will explore various options for generated graphs. We summarize the framework for constructing argument graphs with deductive arguments in Figure 1.

We proceed as follows: (Section 2) We briefly review the definitions for abstract argumentation; (Section 3) We review the nature of base logics in argumentation; (Section 4) We consider options for arguments in deductive argumentation; (Section 5) We consider options for counterarguments in deductive argumentation; (Section 6) We consider options for constructing argument graphs instantiated with deductive arguments; (Section 7) We review some properties and postulates for argumentation based on deductive arguments; and (Section 8) We discuss the approach of deductive argumentation and provide suggestions for further reading.

2 Abstract argumentation

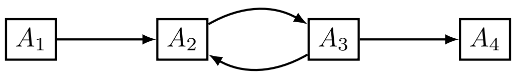
Abstract argumentation, as proposed by [Dung, 1995], provides a good starting point for formalizing argumentation. Dung proposed that a set of arguments and counterarguments could be represented by a directed graph. Each node in the graph denotes an argument and each arc denotes one argument attacking another. So if there is an arc from node A to node B, then A attacks B, or equivalently A is a counterargument to B. See Figure 2 for an example of an abstract argument graph.

An **abstract argument graph** is a pair $(\mathcal{A}, \mathcal{R})$ where $\mathcal{A}$ is a set and $\mathcal{R} \subseteq \mathcal{A} \times \mathcal{A}$. Each element $A \in \mathcal{A}$ is called an **argument** and $(A, B) \in \mathcal{R}$ means that A **attacks** B (accordingly, A is said to be an **attacker** of B) and so A is a **counterargument** for B. A set of arguments $S \subseteq \mathcal{A}$ **attacks** $A_j \in \mathcal{A}$ iff there is an argument $A_i \in S$ such that A_i attacks A_j. Also, S **defends** $A_i \in \mathcal{A}$ iff for each argument $A_j \in \mathcal{A}$, if A_j attacks A_i then S attacks A_j. A set $S \subseteq \mathcal{A}$ of arguments is **conflict-free** iff there are no arguments A_i and A_j in S such that A_i attacks A_j. Let Γ be a conflict-free set of arguments, and let $\mathsf{Defended} : \wp(\mathcal{A}) \rightarrow \wp(\mathcal{A})$ be a function such that $\mathsf{Defended}(\Gamma) = \{A \mid \Gamma \text{ defends } A\}$. We consider the following extensions: (1) Γ is a **complete extension** iff $\Gamma = \mathsf{Defended}(\Gamma)$; (2) Γ is a **grounded extension** iff it is the minimal (w.r.t. set inclusion) complete extension; (3) Γ is a **preferred extension** iff it is a maximal (w.r.t. set inclusion) complete extension; and (4) Γ is a **stable extension** iff it is a preferred extension that attacks every argument that is not in the extension.

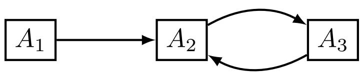
Some argument graphs can be large, and yet we might only be interested in whether some subset of the arguments is in an extension according to some semantics. For this, we introduce the following definitions that lead to the notion of a focal graph.

Definition 2.1 *Let $G = (\mathcal{A}, \mathcal{R})$ be an argument graph. An argument graph $(\mathcal{A}', \mathcal{R}')$ is* **faithful** *with respect to $(\mathcal{A}, \mathcal{R})$ iff $(\mathcal{A}', \mathcal{R}')$ is a subgraph of $(\mathcal{A}, \mathcal{R})$ and for all arguments $A_i, A_j \in \mathcal{A}$, if $A_j \in \mathcal{A}'$ and $(A_i, A_j) \in \mathcal{R}$, then $A_i \in \mathcal{A}'$, and $\mathcal{R}' = \{(A_i, A_j) \mid \mathcal{R} \mid A_i, A_j \in \mathcal{A}'\}$.*

Example 2.2 *Consider the following graph G*



There are three subgraphs that are faithful with respect to G: (1) The graph G ; (2) The subgraph containing just the argument A_1; and (3) The following subgraph. All other subgraphs of G are not faithful.



A faithful subgraph has the same extensions as the graph modulo the arguments in the subgraph. So for every argument A in the subgraph, if A is in the grounded extension in the subgraph, then A is in the grounded extension of the graph, and vice versa. Similarly, for every argument A in the subgraph, if A is in a preferred extension of the subgraph, then A is in a preferred extension of the graph, and vice versa. This follows directly from the directionality criterion of [Baroni and Giacomin, 2007] that says that for a subgraph, arguments in the graph that do not attack any arguments in the subgraph have no affect on the extensions of the subgraph. Therefore, we can ignore the arguments that are not in a faithful subgraph.

Definition 2.3 *Let $\Pi \subseteq \mathcal{A}$ be a set of arguments of interest called the* **focus**. *A graph $(\mathcal{A}', \mathcal{R}')$ is the* **focal graph** *of graph $(\mathcal{A}, \mathcal{R})$ with respect to focus Π iff $(\mathcal{A}', \mathcal{R}')$ is the smallest subgraph of $(\mathcal{A}, \mathcal{R})$ such that $\Pi \subseteq \mathcal{A}'$ and $(\mathcal{A}', \mathcal{R}')$ is faithful with respect to $(\mathcal{A}, \mathcal{R})$.*

Example 2.4 *Continuing Example 2.2, if we let $\Pi = \{A_1, A_2\}$ be the focus, then the third subgraph (i.e. the faithful graph containing A_1, A_2, and A_3) is the focal graph.*

The motivation for finding the focal graph is that given a set of arguments Π as the focus, we want to just have those arguments and any arguments that may affect whether or not any of the arguments in Π are in an extension. By taking the directionality of the arcs into account (i.e. the directionality criteria [Baroni and Giacomin, 2007; Liao *et al.*, 2011]), we can ignore the other arguments.

Even though abstract argumentation provides a clear and precise approach to formalizing aspects of argumentation, the arguments are treated as atomic. There is no formalized content to an argument, and so all arguments are treated as equal. Therefore if we want to understand individual arguments, we need to provide content for them. This leads to the idea of "instantiating" abstract argumentation with deductive arguments. Each deductive argument has some premises from which a claim is derived by deductive reasoning.

3 Base logics

Proposals for logic-based argumentation rely on an underlying logic, which we call a *base logic*, for generating logical arguments and for defining the counterargument relationships (using inference of conflict or existence of inconsistency).

The choice of base logic is an important design decision for a logic-based argumentation system. This then raises the questions of what are the minimal requirements for a base logic and what are the factors that need to be considered for a base logic?

In this paper, we focus on three options for the base logic, namely simple logic, classical logic, and a conditional logic, but other options include modal logic, temporal logic, paraconsistent logic, description logics, and logic programming languages.

Let $\mathcal{L}$ be a language for a logic, and let $\vdash_i$ be the consequence relation for that logic. Therefore, $\vdash_i \subseteq \wp(\mathcal{L}) \times \mathcal{L}$. If α is an atom in $\mathcal{L}$, then α is a **positive literal** in $\mathcal{L}$ and, assuming a negation symbol $\neg$ is available in the language, $\neg\alpha$ is a **negative literal** in $\mathcal{L}$. For a literal β, the **complement** of β is defined as follows: If β is a positive literal, i.e. it is of the form α, then the complement of β is the negative literal $\neg\alpha$, and if β is a negative literal, i.e. it is of the form $\neg\alpha$, then the complement of β is the positive literal α.

The list of properties of a consequence relation given in Table 1 provides a good starting point for considering this question. These properties have been proposed as desirable conditions of a consequence relation. Furthermore, according to Gabbay [Gabbay, 1985] and Makinson [Makinson, 1994], the minimal properties of a consequence relation are reflexivity, monotonicity (or a variant of it) and cut, and the need for each of them can be justified as follows:

- Reflexivity captures the idea of "transparency"; If a formula α is assumed (i.e. $\alpha \in \Delta$), then α can be inferred (i.e $\Delta \vdash_x \alpha$).

- Monotonicity captures the idea of "irreversibility"; Once a formula α is inferred (i.e $\Delta \vdash_x \alpha$), then there is no assumption that can cause α to be withdrawn (i.e. there is no β such that $\Delta \cup \{\beta\} \not\vdash_x \alpha$).

- Cut captures the idea of "equitability" of assumptions and inferences. Once a formula α is inferred (i.e $\Delta \vdash_x \alpha$), it can be used for further reasoning.

These three properties can be seen equivalently in terms of the following three properties based on the consequence closure C_x of a logic x [Makinson, 1994], where $C_x(\Delta) = \{\alpha \mid \Delta \vdash \alpha\}$: (inclusion) $\Delta \subseteq C_x(\Delta)$; (idempotence) $C_x(\Delta) = C_x(C_x(\Delta))$; and (monotony) $C_x(\Delta') \subseteq C_x(\Delta)$ whenever $\Delta' \subseteq \Delta$.

Classes of base logics can be identified using properties of the consequence relation, and then argument systems can be developed in terms of them. For instance, to instantiate abstract argumentation, in [Amgoud and Besnard, 2009], the class of Tarskian logics has been used. This is the class defined by inclusion, idempotence, finiteness (i.e. $C_x(\Delta)$ is the union of $C_x(\Gamma)$ for all finite subsets Γ of Δ), absurdity (i.e. $C_x(\{\phi\}) = \mathcal{L}$ for some ϕ in the language $\mathcal{L}$), and coherence (i.e. $C_x(\emptyset) \neq \mathcal{L}$). Classical logic is an example of a Tarskian logic.

4 Arguments

A **deductive argument** is an ordered pair $\langle \Phi, \alpha \rangle$ where $\Phi \vdash_i \alpha$. Φ is the support, or premises, or assumptions of the argument, and α is the claim, or conclusion, of the argument. The definition for a deductive argument only assumes that the premises entail the claim (i.e. $\Phi \vdash_i \alpha$). For an argument $A = \langle \Phi, \alpha \rangle$, the function $\mathsf{Support}(A)$ returns Φ and the function $\mathsf{Claim}(A)$ returns α.

Many proposals have further constraints for an ordered pair $\langle \Phi, \alpha \rangle$ to be an argument. The most commonly assumed constraint is the **consistency constraint**: An argument $\langle \Phi, \alpha \rangle$ satisfies this constraint when Φ is consistent (assuming that the base logic has a notion of consistency). For richer logics, such as classical logic, consistency is often regarded as a desirable property of a deductive argument because claims that are obtained with logics such as classical logic from inconsistent premises are normally useless as illustrated in the next example.

Example 4.1 *If we assume the consistency constraint, then the following are*

Name	Property
Reflexivity	$\Delta \cup \{\alpha\} \vdash_x \alpha$
Literal reflexivity	$\Delta \cup \{\alpha\} \vdash_x \alpha$ if α is a literal
Left logical equivalent	$\Delta \cup \{\beta\} \vdash_x \gamma$ if $\Delta \cup \{\alpha\} \vdash_x \gamma$ and $\vdash \alpha \leftrightarrow \beta$
Right weakening	$\Delta \vdash_x \alpha$ if $\Delta \vdash_x \beta$ and $\vdash \beta \rightarrow \alpha$
And	$\Delta \vdash_x \alpha \wedge \beta$ if $\Delta \vdash_x \alpha$ and $\Delta \vdash_x \beta$
Monotonicity	$\Delta \cup \{\alpha\} \vdash_x \beta$ if $\Delta \vdash_x \beta$
Cut	$\Delta \vdash_x \beta$ if $\Delta \vdash_x \alpha$ and $\Delta \cup \{\alpha\} \vdash_x \beta$
Conditionalization	$\Delta \vdash_x \alpha \rightarrow \beta$ if $\Delta \cup \{\alpha\} \vdash_x \beta$
Deduction	$\Delta \cup \{\alpha\} \vdash_x \beta$ if $\Delta \vdash_x \alpha \rightarrow \beta$
Contraposition	$\Delta \cup \{\alpha\} \vdash_x \beta$ if $\Delta \cup \{\neg\beta\} \vdash_x \neg\alpha$
Or	$\Delta \cup \{\alpha \vee \beta\} \vdash_x \gamma$ if $\Delta \cup \{\alpha\} \vdash_x \gamma$ and $\Delta \cup \{\beta\} \vdash_x \gamma$

Table 1. Some properties of a consequence relation $\vdash_x$ adapted from D. Makinson. General patterns in nonmonotonic reasoning. In D. Gabbay, C. Hog- ger, and J. Robinson, editors, *Handbook of Logic in Artificial Intelligence and Logic Programming*, volume 3, pages 35–110. Oxford University Press, 1994.

not arguments.

$$\langle \{\texttt{study(Sid,logic)}, \neg\texttt{study(Sid,logic)}\},\\ \texttt{study(Sid,logic)} \leftrightarrow \neg\texttt{study(Sid,logic)}\rangle$$

$$\langle \{\texttt{study(Sid,logic)}, \neg\texttt{study(Sid,logic)}\}, \texttt{KingOfFrance(Sid)}\rangle$$

In contrast, for weaker logics (such as paraconsistent logics), it may be desirable to not impose the consistency constraint. With such logics, a credulous approach could be taken so that pros and cons could be obtained from inconsistent premises (as illustrated by the following example).

Example 4.2 *If we assume the base logic is a paraconsistent logic (such as Belnap's four valued logic), and we do not impose the consistent constraint, then the following are arguments.*

$$\langle \{\texttt{study(Sid,logic)} \wedge \neg\texttt{study(Sid,logic)}\}, \texttt{study(Sid,logic)}\rangle$$

$$\langle \{\texttt{study(Sid,logic)} \wedge \neg\texttt{study(Sid,logic)}\}, \neg\texttt{study(Sid,logic)}\rangle$$

Another commonly assumed constraint is the **minimality constraint**: An argument $\langle \Phi, \alpha \rangle$ satisfies this constraint when there is no $\Psi \subset \Phi$ such that $\Psi \vdash \alpha$. Minimality is often regarded as a desirable property of a deductive argument because it eliminates irrelevant premises (as in the following example).

Example 4.3 *If we assume the minimality constraint, then the following is not an argument.*

$$\langle\{\texttt{report}(\texttt{rain}), \texttt{report}(\texttt{rain}) \rightarrow \texttt{carry}(\texttt{umbrella}), \texttt{happy}(\texttt{Sid})\}, \texttt{carry}(\texttt{umbrella})\rangle$$

When we construct a knowledgebase, with simple logic, classical logic, or other base logics, it is possible that some or all of the formulae could be incorrect. For instance, individual formulae may come from different and conflicting sources, they may reflect options that disagree, they may represent uncertain information. A knowledgebase may be inconsistent, and individual formulae may be contradictory. After all, if the knowledge is not inconsistent (i.e. it is consistent), then we will not have counterarguments. We may also include formulae that we know are not always correct. For instance, we may include a formula such as the following that says that a water sample taken from the Mediterranean sea in the summer will be above 15 degrees Celcius. While this may be a useful general rule, it is not always true. For instance, the sample could be taken when there is a period of bad weather, or the sample is taken from a depth of over 500 metres.

$$\forall \texttt{X}, \texttt{Y}.\texttt{watersample}(\texttt{X}) \land \texttt{location}(\texttt{X}, \texttt{Mediterranean}) \\ \land \texttt{season}(\texttt{X}, \texttt{summer}) \land \texttt{termperature}(\texttt{X}, \texttt{Y}) \rightarrow \texttt{Y} > \texttt{15}$$

In the following subsections, we define arguments based on simple logic and on classical logic as the base logic. Alternative base logics include description logic, paraconsistent logic, temporal logic, and conditional logic.

4.1 Arguments based on simple logic

Simple logic is based on a language of literals and simple rules where each **simple rule** is of the form $\alpha_1 \land \ldots \land \alpha_k \rightarrow \beta$ where α_1 to α_k and β are literals. A **simple logic knowledgebase** is a set of literals and simple rules. The consequence relation is modus ponens (i.e. implication elimination) as defined next.

Definition 4.4 *The* **simple consequence relation**, *denoted* $\vdash_s$, *which is the smallest relation satisfying the following condition, and where* Δ *is a simple logic knowledgebase:* $\Delta \vdash_s \beta$ *iff there is an* $\alpha_1 \land \cdots \land \alpha_n \rightarrow \beta \in \Delta$, *and for each* $\alpha_i \in \{\alpha_1, \ldots, \alpha_n\}$, *either* $\alpha_i \in \Delta$ *or* $\Delta \vdash_s \alpha_i$.

Note, the simple consequence relation does not satisfy reflexivity. We could slightly amend the definition so that it does satisfy reflexivity but the above definition will be useful to us later when we consider properties of the argumentation based on this base logic.

Example 4.5 *Let* $\Delta = \{a, b, a \land b \rightarrow c, c \rightarrow \neg d\}$. *Hence,* $\Delta \vdash_s c$ *and* $\Delta \vdash_s \neg d$. *However,* $\Delta \not\vdash_s a$ *and* $\Delta \not\vdash_s b$.

Definition 4.6 *Let Δ be a simple logic knowledgebase. For $\Phi \subseteq \Delta$, and a literal α, $\langle \Phi, \alpha \rangle$ is a* **simple argument** *iff $\Phi \vdash_s \alpha$ and there is no proper subset Φ' of Φ such that $\Phi' \vdash_s \alpha$.*

So each simple argument is minimal but not necessarily consistent (where consistency for a simple logic knowledgebase Δ means that for no atom α does $\Delta \vdash_s \alpha$ and $\Delta \vdash_s \neg\alpha$ hold). We do not impose the consistency constraint in the definition for simple arguments as simple logic is paraconsistent, and therefore can support a credulous view on the arguments that can be generated.

Example 4.7 *Let $\mathtt{p}_1$, $\mathtt{p}_2$, and $\mathtt{p}_3$ be the following formulae. Then the following is a simple argument.*

$$\langle \{\mathtt{p}_1, \mathtt{p}_2, \mathtt{p}_3\}, \mathtt{goodInvestment(BP)} \rangle$$

Note, we use $\mathtt{p}_1$, $\mathtt{p}_2$, and $\mathtt{p}_3$ as labels in order to make the presentation of the premises more concise.

$\mathtt{p}_1 = \mathtt{oilCompany(BP)}$
$\mathtt{p}_2 = \mathtt{goodPerformer(BP)}$
$\mathtt{p}_3 = \mathtt{oilCompany(BP)} \land \mathtt{goodPerformer(BP)}) \rightarrow \mathtt{goodInvestment(BP)}$

Simple logic is a practical choice as a base logic for argumentation. Having a logic with simple rules and modus ponens is useful for applications because the behaviour is quite predictable in the sense that given a knowledgebase it is relatively easy to anticipate the inferences that come from the knowlegebase. Furthermore, it is relatively easy to implement an algorithm for generating the arguments and counterarguments from a knowledgebase. The downside of simple logic as a base logic is that the proof theory is weak. It only incorporates modus ponens (i.e. implication elimination) and so many useful kinds of reasoning (e.g. contrapositive reasoning) are not supported.

4.2 Arguments based on classical logic

Classical logic is appealing as the choice of base logic as it better reflects the richer deductive reasoning often seen in arguments arising in discussions and debates.

We assume the usual propositional and predicate (first-order) languages for classical logic, and the usual the **classical consequence relation**, denoted $\vdash$. A **classical knowledgebase** is a set of classical propositional or predicate formulae.

Definition 4.8 *For a classical knowledgebase Φ, and a classical formula α, $\langle \Phi, \alpha \rangle$ is a* **classical argument** *iff $\Phi \vdash \alpha$ and $\Phi \not\vdash \bot$ and there is no proper subset Φ' of Φ such that $\Phi' \vdash \alpha$.*

So a classical argument satisfies both minimality and consistency. We impose the consistency constraint because we want to avoid the useless inferences that come with inconsistency in classical logic (such as via ex falso quodlibet).

Example 4.9 *The following classical argument uses a universally quantified formula in contrapositive reasoning to obtain the claim about number 77.*

$$\langle\{\forall \mathtt{X}.\mathtt{multipleOfTen}(\mathtt{X}) \to \mathtt{even}(\mathtt{X}), \neg\mathtt{even}(77)\}, \neg\mathtt{multipleOfTen}(77)\rangle$$

Given the central role classical logic has played in philosophy, linguistics, and computer science (software engineering, formal methods, data and knowledge engineering, artificial intelligence, computational linguistics, etc.), we should consider how it can be used in argumentation. Classical propositional logic and classical predicate logic are expressive formalisms which capture more detailed aspects of the world than is possible with restricted formalisms such as simple logic.

4.3 Arguments based on conditional logic

Conditional logics are a valuable alternative to classical logic for knowledge representation and reasoning. They can be used to capture hypothetical statements of the form "If α were true, then β would be true". This done by introducing an extra connective $\Rightarrow$ to extend a classical logic language. Informally, $\alpha \Rightarrow \beta$ is valid when β is true in the possible worlds where α is true. Representing and reasoning with such knowledge in argumentation is valuable because useful arguments exist that refer to fictitious and hypothetical situations (see [Besnard *et al.*, 2013] for some examples).

In this review, we consider the well-known conditional logic MP which can be extended to give many other well-known conditional logics, and we follow the presentation for argumentation given by [Besnard *et al.*, 2013]. The language of conditional logic is that of classical logic extended with the formulae of the form $\alpha \Rightarrow \beta$ where α and β are formulae in the language of conditional logic. The proof theory for the consequence relation $\vdash_c$ of MP is given by classical logic proposition extended by the following axiom schemas and inference rules.

$$\begin{array}{rl}
RCEA & \dfrac{\vdash_c \alpha \leftrightarrow \beta}{\vdash_c (\alpha \Rightarrow \gamma) \leftrightarrow (\beta \Rightarrow \gamma)} \\
RCEC & \dfrac{\vdash_c \alpha \leftrightarrow \beta}{\vdash_c (\gamma \Rightarrow \alpha) \leftrightarrow (\gamma \Rightarrow \beta)} \\
CC & \vdash_c ((\alpha \Rightarrow \beta) \land (\alpha \Rightarrow \gamma)) \to (\alpha \Rightarrow (\beta \land \gamma)) \\
CM & \vdash_c (\alpha \Rightarrow (\beta \land \gamma)) \to ((\alpha \Rightarrow \beta) \land (\alpha \Rightarrow \gamma)) \\
CN & \vdash_c (\alpha \to \top) \\
MP & \vdash_c (\alpha \to \beta) \to (\alpha \to \beta)
\end{array}$$

Using the $\vdash_c$ consequence relation, we can define the notion of an argument with the same constraints as for classical logic arguments. For this, a conditional knowledgebase is a set of formulae of the language of conditional logic.

Definition 4.10 *For a conditional knowledgebase Delta, and a formula of conditional logic α, $\langle \Delta, \alpha \rangle$ is a* **conditional argument** *iff $\Delta \vdash_c \alpha$ and $\Delta \not\vdash_c \bot$ and there is no proper subset Δ' of Δ such that $\Delta' \vdash \alpha$.*

Example 4.11 *Let $\Delta = \{\mathtt{matchIsStruck} \Rightarrow \mathtt{matchLights}, \mathtt{matchIsStruck}\}$. From this knowledgebase, we get the following argument.*

$$\langle \{\mathtt{matchIsStruck} \Rightarrow \mathtt{matchLights}, \mathtt{matchIsStruck}\}, \mathtt{matchLights} \rangle$$

Note, from Δ, we cannot get the following argument.

$$\langle \{\mathtt{matchIsStruck} \wedge \mathtt{matchIsWet} \Rightarrow \mathtt{matchLights}, \mathtt{matchIsStruck}\}, \mathtt{matchLights} \rangle$$

Whereas if we consider $\Delta' = \{\mathtt{matchIsStruck} \rightarrow \mathtt{matchLights}, \mathtt{matchIsStruck}\}$ where we use the classical implication for the formula, we get the classical argument from Δ'.

$$\langle \{\mathtt{matchIsStruck} \wedge \mathtt{matchIsWet} \rightarrow \mathtt{matchLights}, \mathtt{matchIsStruck}\}, \mathtt{matchLights} \rangle$$

This is because from $\mathtt{matchIsStruck} \rightarrow \mathtt{matchLights}$, we can infer the following using classical logic.

$$\mathtt{matchIsStruck} \wedge \mathtt{matchIsWet} \rightarrow \mathtt{matchLights}$$

So the above example illustrates how the proof theory (and the semantics) for conditional logic is more restricted for the $\Rightarrow$ connective than for the $\rightarrow$. This makes it useful for representing and reasoning with knowledge about fictitious and hypothetical situations [Cross and Nute, 2001; Girard, 2006].

5 Counterarguments

A counterargument is an argument that attacks another argument. In deductive argumentation, we define the notion of counterargument in terms of logical contradiction between the claim of the counterargument and the premises of claim of the attacked argument. We explore some of the kinds of counterargument that can be specified for simple logic, classical logic and classical logic.

5.1 Counteraguments based on simple logic

For simple logic, we consider two forms of counterargument. For this, recall that literal α is the complement of literal β if and only if α is an atom and β is $\neg\alpha$ or if β is an atom and α is $\neg\beta$.

Definition 5.1 *For simple arguments A and B, we consider the following type of* **simple attack***:*

- *A is a* **simple undercut** *of B if there is a simple rule $\alpha_1 \wedge \cdots \wedge \alpha_n \rightarrow \beta$ in $\mathsf{Support}(B)$ and there is an $\alpha_i \in \{\alpha_1, \ldots, \alpha_n\}$ such that $\mathsf{Claim}(A)$ is the complement of α_i*

- *A is a* **simple rebut** *of B if* $\mathsf{Claim}(A)$ *is the complement of* $\mathsf{Claim}(B)$

Example 5.2 *The first argument A_1 captures the reasoning that the metro is an efficient form of transport, so one can use it. The second argument A_2 captures the reasoning that there is a strike on the metro, and so the metro is not an efficient form of transport (at least on the day of the strike). A_2 is a simple undercut of A_1.*

$$A_1 = \langle\{\texttt{efficientMetro}, \texttt{efficientMetro} \rightarrow \texttt{useMetro}\}, \texttt{useMetro}\rangle$$
$$A_2 = \langle\{\texttt{strikeMetro}, \texttt{strikeMetro} \rightarrow \neg\texttt{efficientMetro}\}, \neg\texttt{efficientMetro}\rangle$$

Example 5.3 *The first argument A_1 captures the reasoning that the government has a budget deficit, and so the government should cut spending. The second argument A_2 captures the reasoning that the economy is weak, and so the government should not cut spending. The arguments are simple rebuts of each other.*

$$A_1 = \langle\{\texttt{govDeficit}, \texttt{govDeficit} \rightarrow \texttt{cutGovSpend}\}, \texttt{cutGovSpend}\rangle$$
$$A_2 = \langle\{\texttt{weakEconomy}, \texttt{weakEconomy} \rightarrow \neg\texttt{cutGovSpend}\}, \neg\texttt{cutGovSpend}\rangle$$

So in simple logic, a rebut attacks the claim of an argument, and an undercut attacks the premises of the argument (by attacking one of the consequents of one of the rules in the premises).

Simple arguments and counterarguments can be used to model defeasible reasoning. For this, we use simple rules that are normally correct but sometimes are incorrect. For instance, if Sid has the goal of going to work, Sid takes the metro. This is generally true, but sometimes Sid works at home, and so it is no longer true that Sid takes the metro, as we see in the next example.

Example 5.4 *The first argument A_1 captures the general rule that if* `workDay` *holds, then* `metro(Sid)` *holds (denoting that Sid takes the metro). The use of the simple rule in A_1 requires that the assumption* `normal` *holds. This is given as an assumption. The second argument A_2 undercuts the first argument by contradicting the assumption that* `normal` *holds*

$$A_1 = \langle\{\texttt{workDay}, \texttt{normal}, \texttt{workDay} \wedge \texttt{normal} \rightarrow \texttt{metro(Sid)}\}, \texttt{metro(Sid)}\rangle$$
$$A_2 = \langle\{\texttt{workAtHome(Sid)}, \texttt{workAtHome(Sid)} \rightarrow \neg\texttt{normal}\}, \neg\texttt{normal}\rangle$$

Informally, if we start with just argument A_1, then A_1 is undefeated, and so `metro(Sid)` *is an acceptable claim. However, if we then add A_2, then A_1 is a defeated argument and A_2 is an undefeated argument. Hence, if we have A_2, we have to withdraw* `metro(Sid)` *as an acceptable claim.*

So by having appropriate conditions in the antecedent of a simple rule we can disable the rule by generating a counterargument that attacks it. This in effect stops the usage of the simple rule. This means that we have a convention to attack an argument based on the inferences obtained by the simple logic (e.g. as in Example 5.2 and Example 5.3), or on the rules used (e.g. Example 5.4).

This way to disable rules by adding appropriate conditions (as in Example 5.4) is analogous to the use abnormality predicates used in formalisms such as circumscription (see for example [McCarthy, 1980]). We can use the same approach to capture defeasible reasoning in other logics such as classical logic. Note, this does not mean that we turn the base logic into a nonmonotonic logic. Both simple logic and classical logic are monotonic logics. Hence, for a simple logic knowledgebase Δ (and similarly for a classical logic knowledgebase Δ), the set of simple arguments (respectively classical arguments) obtained from Δ is a subset of the set of simple arguments (respectively classical arguments) obtained from $\Delta \cup \{\alpha\}$ where α is a formula not in Δ. But at the level of evaluating arguments and counterarguments, we have non-monotonic defeasible behaviour. For instance in Example 5.2, with just A_1 we have the acceptable claim that `useMetro`, but then when we have also A_2, we have to withdraw this claim. In other words, if the set of simple arguments is $\{A_1\}$, then we can construct an argument graph with just A_1, and by applying Dung's dialectical semantics, there is one extension containing A_1. However, if the set of simple arguments is $\{A_1, A_2\}$, then we can construct an argument graph with A_1 attacked by A_2, and by applying Dung's dialectical semantics, there is one extension containing A_2. This illustrates the fact that the argumentation process is nonmonotonic.

5.2 Counterarguments based on classical logic

Given the expressivity of classical logic (in terms of language and inferences), there are a number of natural ways to define counterarguments.

Definition 5.5 *Let A and B be two classical arguments. We define the following types of* **classical attack**.

- *A is a* **classical defeater** *of B if* $\mathsf{Claim}(A) \vdash \neg \bigwedge \mathsf{Support}(B)$.
- *A is a* **classical direct defeater** *of B if* $\exists \phi \in \mathsf{Support}(B)$ *s.t.* $\mathsf{Claim}(A) \vdash \neg\phi$.
- *A is a* **classical undercut** *of B if* $\exists \Psi \subseteq \mathsf{Support}(B)$ *s.t.* $\mathsf{Claim}(A) \equiv \neg \bigwedge \Psi$.
- *A is a* **classical direct undercut** *of B if* $\exists \phi \in \mathsf{Support}(B)$ *s.t.* $\mathsf{Claim}(A) \equiv \neg\phi$.
- *A is a* **classical canonical undercut** *of B if* $\mathsf{Claim}(A) \equiv \neg \bigwedge \mathsf{Support}(B)$.
- *A is a* **classical rebuttal** *of B if* $\mathsf{Claim}(A) \equiv \neg\mathsf{Claim}(B)$.
- *A is a* **classical defeating rebuttal** *of B if* $\mathsf{Claim}(A) \vdash \neg\mathsf{Claim}(B)$.

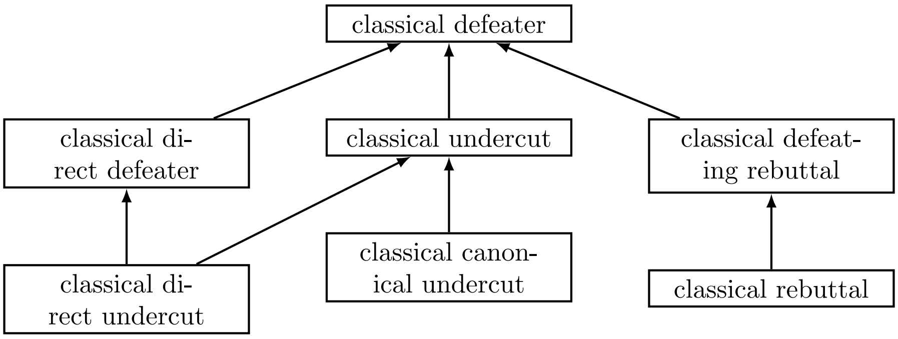


Figure 3. We can represent the containment between the classical attack relations as above where an arrow from R_1 to R_2 indicates that $R_1 \subseteq R_2$.

Note, in the rest of this section, we will drop the term "classical" when we discuss these types of attack.

To illustrate these different notions of classical counterargument, we consider the following examples, and we relate these definitions in Figure 3 where we show that classical defeaters are the most general of these definitions.

Example 5.6 *Let* $\Delta = \{a \vee b, a \leftrightarrow b, c \rightarrow a, \neg a \wedge \neg b, a, b, c, a \rightarrow b, \neg a, \neg b, \neg c\}$

$\langle\{a \vee b, c\}, (a \vee b) \wedge c\rangle$ *is a defeater of* $\langle\{\neg a, \neg b\}, \neg a \wedge \neg b\rangle$
$\langle\{a \vee b, c\}, (a \vee b) \wedge c\rangle$ *is a direct defeater of* $\langle\{\neg a \wedge \neg b\}, \neg a \wedge \neg b\rangle$
$\langle\{\neg a \wedge \neg b\}, \neg(a \wedge b)\rangle$ *is a undercut of* $\langle\{a, b, c\}, a \wedge b \wedge c\rangle$
$\langle\{\neg a \wedge \neg b\}, \neg a\rangle$ *is a direct undercut of* $\langle\{a, b, c\}, a \wedge b \wedge c\rangle$
$\langle\{\neg a \wedge \neg b\}, \neg(a \wedge b \wedge c)\rangle$ *is a canonical undercut of* $\langle\{a, b, c\}, a \wedge b \wedge c\rangle$
$\langle\{a, a \rightarrow b\}, b \vee c\rangle$ *is a rebuttal of* $\langle\{\neg a \wedge \neg b, \neg c\}, \neg(b \vee c)\rangle$
$\langle\{a, a \rightarrow b\}, b\rangle$ *is a defeating rebuttal of* $\langle\{\neg a \wedge \neg b, \neg c\}, \neg(b \vee c)\rangle$

Using simple logic, the definitions for counterarguments against the support of another argument are limited to attacking just one of the items in the support. In contrast, using classical logic, a counterargument can be against more than one item in the support. For example, in Example 5.7, the undercut is not attacking an individual premise but rather saying that two of the premises are incompatible (in this case that the premises `lowCostFly` and `luxuryFly` are incompatible).

Example 5.7 *Consider the following arguments.* A_1 *is attacked by* A_2 *as* A_2 *is an undercut of* A_1 *though it is neither a direct undercut nor a canonical undercut. Essentially, the attack says that the flight cannot be both a low cost* *flight and a luxury flight.*

$A_1 = \langle\{\texttt{lowCostFly}, \texttt{luxFly}, \texttt{lowCostFly} \wedge \texttt{luxFly} \rightarrow \texttt{goodFly}\}, \texttt{goodFly}\rangle$
$A_2 = \langle\{\neg\texttt{lowCostFly} \vee \neg\texttt{luxFly}\}, \neg\texttt{lowCostFly} \vee \neg\texttt{luxFly}\rangle$

Trivially, undercuts are defeaters but it is also quite simple to establish that rebuttals are defeaters. Furthermore, if an argument has defeaters then it has undercuts. It may happen that an argument has defeaters but no rebuttals as illustrated next.

Example 5.8 *Let* $\Delta = \{\neg\texttt{containsGarlic} \wedge \texttt{goodDish}, \neg\texttt{goodDish}\}$. *Then the following argument has at least one defeater but no rebuttal.*

$$\langle\{\neg\texttt{containsGarlic} \wedge \texttt{goodDish}\}, \neg\texttt{containsGarlic}\rangle$$

There are some important differences between rebuttals and undercuts that can be seen in the following examples.

Example 5.9 *Consider the following arguments. The first argument* A_1 *is a direct undercut to the second argument* A_2*, but neither rebuts each other. Furthermore,* A_1 *"agrees" with the claim of* A_2 *since the premises of* A_1 *could be used for an alternative argument with the same claim as* A_2*.*

$A_1 = \langle\{\neg\texttt{containsGarlic} \wedge \neg\texttt{goodDish}\}, \neg\texttt{containsGarlic}\rangle$
$A_2 = \langle\{\texttt{containsGarlic}, \texttt{containsGarlic} \rightarrow \neg\texttt{goodDish}\}, \neg\texttt{goodDish}\rangle$

Example 5.10 *Consider the following arguments. The first argument is a rebuttal of the second argument, but it is not an undercut because the claim of the first argument is not equivalent to the negation of some subset of the premises of the second argument.*

$A_1 = \langle\{\texttt{goodDish}\}, \texttt{goodDish}\rangle$
$A_2 = \langle\{\texttt{containsGarlic}, \texttt{containsGarlic} \rightarrow \neg\texttt{goodDish}\}, \neg\texttt{goodDish}\rangle$

So an undercut for an argument need not be a rebuttal for that argument, and a rebuttal for an argument need not be an undercut for that argument.

Arguments are not necessarily independent. In a sense, some encompass others (possibly up to some form of equivalence), which is the topic we now turn to.

Definition 5.11 *An argument* $\langle\Phi, \alpha\rangle$ *is* **more conservative** *than an argument* $\langle\Psi, \beta\rangle$ *iff* $\Phi \subseteq \Psi$ *and* $\beta \vdash \alpha$.

Example 5.12 $\langle\{a\}, a \vee b\rangle$ *is more conservative than* $\langle\{a, a \rightarrow b\}, b\rangle$.

Roughly speaking, a more conservative argument is more general: It is, so to speak, less demanding on the support and less specific about the claim.

Example 5.13 *Consider the following formulae.*

$$\begin{aligned}
\mathtt{p}_1 &= \mathtt{divisibleByTen(50)} \\
\mathtt{p}_2 &= \forall \mathtt{X}.\mathtt{divisibleByTen(X)} \rightarrow \mathtt{divisibleByTwo(X)} \\
\mathtt{p}_3 &= \forall \mathtt{X}.\mathtt{divisibleByTwo(X)} \rightarrow \mathtt{even(X)}
\end{aligned}$$

Hence, A_1 is an argument with the claim "The number 50 is divisible by 2", and A_2 is an argument with the claim "The number 50 is divisible by 2 and the number 50 is an even number". However, A_1 is more conservative than A_2.

$$\begin{aligned}
A_1 &= \langle \{\mathtt{p}_1, \mathtt{p}_2\}, \mathtt{divisibleByTwo(50)} \rangle \\
A_2 &= \langle \{\mathtt{p}_1, \mathtt{p}_2, \mathtt{p}_3, \}, \mathtt{even(50)} \wedge \mathtt{divisibleByTwo(50)} \rangle
\end{aligned}$$

We can use the notion of "more conservative" to help us identify the most useful counterarguments amongst the potentially large number of counterarguments.

Example 5.14 *Let $\{a, b, c, \neg a \vee \neg b \vee \neg c\}$ be our knowledgebase. Suppose we start with the argument $\langle \{a, b, c\}, a \wedge b \wedge c \rangle$. Now we have numerous undercuts to this argument including the following.*

$$\begin{gathered}
\langle \{b, c, \neg a \vee \neg b \vee \neg c\}, \neg a \rangle \\
\langle \{a, c, \neg a \vee \neg b \vee \neg c\}, \neg b \rangle \\
\langle \{a, b, \neg a \vee \neg b \vee \neg c\}, \neg c \rangle \\
\langle \{c, \neg a \vee \neg b \vee \neg c\}, \neg a \vee \neg b \rangle \\
\langle \{b, \neg a \vee \neg b \vee \neg c\}, \neg a \vee \neg c \rangle \\
\langle \{a, \neg a \vee \neg b \vee \neg c\}, \neg b \vee \neg c \rangle \\
\langle \{\neg a \vee \neg b \vee \neg c\}, \neg a \vee \neg b \vee \neg c \rangle
\end{gathered}$$

All these undercuts say the same thing which is that the set $\{a, b, c\}$ is inconsistent together with the formula $\neg a \vee \neg b \vee \neg c$. As a result, this can be captured by the last undercut listed above. Note this is the maximally conservative undercut amongst the undercuts listed, and moreover it is a canonical undercut. This example therefore illustrates how the canonical undercuts are the undercuts that (in a sense) represent all the other undercuts.

So choosing classical logic as the base logic gives us a wider range of choices for defining attacks. This has advantages if we want to better capture argumentation as arising in natural language, or to more precisely capture counterarguments generated from certain kinds of knowledge. However, it does also mean that we need to be aware of the consequences of our choice of definition for attacks when using a generated approach to instantiating argument graphs (as we will discuss in the next section).

5.3 Counterarguments based on conditional logic

In order to define counterarguments, we follow the proposal in [Besnard *et al.*, 2013] for *conditional contrariety*. For this, we require the notion of an extended literal which is either of the form ϕ or $\neg\phi$ where ϕ is either an atom or a formula of the form $\alpha \Rightarrow \beta$. Then we exploit the fact that any formula of conditional logic can be rewritten using the proof rules into an equivalent set of disjunctive formulae of the form $\alpha_1 \vee \ldots \vee \alpha_n$ where each α_i is an extended literal. In particular, for the next definition, we are interested in the specific situation when a formula can be equivalently represented by a single disjunctive formula.

Definition 5.15 *Let α and β be two formulae of conditional logic such that the disjunctive form of α is $\alpha_1 \vee \ldots \vee \alpha_m$ and the disjunctive form of β is $\beta_1 \vee \ldots \vee \beta_n$. α is the* **contrary** *of β, denoted $\alpha \triangleright \beta$ iff for all $\alpha_i \in \{\alpha_1, \ldots, \alpha_m\}$, and for all $\beta_j \in \{\beta_1, \ldots, \beta_n\}$,*

1. $\{\alpha_i, \beta_j\} \vdash_c \bot$
2. *there exists $\gamma_1, \gamma_2, \delta_1, \delta_2$ in the language such that the following conditions are satisfied*
   (a) $\gamma_1 \vdash_c \delta_1$
   (b) $\delta_1 \not\vdash_c \gamma_1$
   (c) $\gamma_1 \not\vdash_c \gamma_2$
   (d) $\delta_2 \vdash_c \gamma_2$
   (e) $\gamma_2 \vdash_c \delta_2$

   where either $\{\alpha_i, \beta_j\} \vdash_c (\gamma_1 \Rightarrow \gamma_2) \wedge (\delta_1 \Rightarrow \delta_2)$ such that $\alpha \vdash_c \gamma_1 \Rightarrow \gamma_2$, or $\beta_j \vdash_c (\gamma_1 \Rightarrow \gamma_2) \wedge (\delta_1 \rightarrow \delta_2)$.

In the above definition, the first condition subsumes "classical contradiction", and the second condition captures situations where two rules conflict as illustated by the following examples.

- $a \wedge (a \wedge c \Rightarrow f)$ is a contrary of $a \rightarrow (a \Rightarrow f)$
- $a \wedge (a \wedge c \Rightarrow f)$ is a contrary of $\neg a \vee c \Rightarrow f$.
- $\neg a \wedge b$ and $a \wedge (a \wedge c \rightarrow b \vee \neg d)$ are the contrary of each other.

As another illustration of rules for which the second condition of the above definition applies is the following where the second rule is the contrary of the first. The intuition of the example is that the second rule "corrects" the circumstances under which John will go to watch the match at the stadium.

- `matchTonight` $\Rightarrow$ `JohnGoesToTheStadium`
- `matchTonight` $\wedge$ `JohnHasEnoughMoney` $\Rightarrow$ `JohnGoesToTheStadium`

Example 5.16 *Let* $\Delta = \{a \Rightarrow b, a \vee d \Rightarrow b \wedge c, a \Rightarrow c\}$*. Let* α *be* $a \Rightarrow b \wedge c$*. Note that* $\Delta \vdash_c \alpha$*. However,* $\alpha \rhd \Delta$ *because* $\alpha \rhd a \vee d \Rightarrow b \wedge c$*.*

The notion of contrary is extended to sets of conditional formulae Φ so that $\alpha \rhd \Phi$ holds iff there exists a $\beta \in \Phi$ such that $\Phi \vdash_c \beta$ and $\alpha \rhd \beta$ holds.

Example 5.17 *Let* $\Delta = \{a \Rightarrow b, a \vee d \Rightarrow b \wedge c, a \Rightarrow c\}$*. Let* α *be* $a \Rightarrow b \wedge c$*. Note that* $\Delta \vdash_c \alpha$*. However,* $\alpha \rhd \Delta$ *because* $\alpha \rhd a \vee d \Rightarrow b \wedge c$*.*

Now we can extend the definitions of counterargument given for classical logic. We just give two options to illustrate the space of possibilities.

Definition 5.18 *Let* $\langle \Phi, \alpha \rangle$ *and* $\Psi, \beta\rangle$ *be conditional logic arguments.*

- $\langle \Psi, \beta \rangle$ *is a* **conditional rebuttal** *for* $\langle \Phi, \alpha \rangle$ *iff* $\beta \rhd \alpha$*.*
- $\langle \Psi, \beta \rangle$ *is a* **conditional defeater** *for* $\langle \Phi, \alpha \rangle$ *iff* $\beta \rhd \Phi$*.*

Example 5.19 *Below, the second argument is a conditional rebuttal for the first argument.*

$$\begin{array}{l}\langle\{\texttt{matchTonight} \Rightarrow \texttt{JohnGoesToTheStadium}\}, \\ \qquad \texttt{matchTonight} \Rightarrow \texttt{JohnGoesToTheStadium}\rangle \\ \langle\{\texttt{matchTonight} \wedge \texttt{JohnHasEnoughMoney} \Rightarrow \texttt{JohnGoesToTheStadium}\}, \\ \qquad \texttt{matchTonight} \wedge \texttt{JohnHasEnoughMoney} \Rightarrow \texttt{JohnGoesToTheStadium}\rangle\end{array}$$

Example 5.20 *Some conditional defeaters for* $\langle\{a \vee \neg d \Rightarrow b \wedge c, f \vee \neg b, b\}, f \wedge (a \vee \neg d \Rightarrow b \wedge c)\rangle$ *are listed below.*

$$\begin{array}{l}\langle\{\neg b\}, \neg b\rangle \\ \langle\{\neg b\}, \neg(\neg b \rightarrow b)\rangle \\ \langle\{\neg b, \neg a \rightarrow b\}, \neg b \wedge a\rangle \\ \langle\{e \wedge \neg d \Rightarrow b \wedge c\}, e \wedge \neg d \Rightarrow b \wedge c\rangle \\ \langle\{a \Rightarrow b, a \Rightarrow c\}, a \Rightarrow b \wedge c\rangle \\ \langle\{a \Rightarrow b, a \Rightarrow c\}, \neg\neg(a \Rightarrow b \wedge c)\rangle \\ \langle\{a \Rightarrow b, a \Rightarrow c\}, (a \Rightarrow b) \wedge (a \Rightarrow c)\rangle\end{array}$$

Example 5.21 *Below, the second argument is a conditional defeater for the first argument.*

$$\begin{array}{l}\langle\{\texttt{matchTonight}, \texttt{matchTonight} \Rightarrow \texttt{JohnGoesToTheStadium}\}, \\ \qquad \texttt{JohnGoesToTheStadium}\rangle \\ \langle\{\texttt{matchTonight} \wedge \texttt{JohnHasEnoughMoney} \Rightarrow \texttt{JohnGoesToTheStadium}\}, \\ \qquad \texttt{matchTonight} \wedge \texttt{JohnHasEnoughMoney} \Rightarrow \texttt{JohnGoesToTheStadium}\rangle\end{array}$$

By using conditional logic as a base logic, we have a range of options for more effective modelling complex real-world scenarios. Whilst many conditional logics extend classical logic, the implication introduced is normally more restricted than the strict implication used in classical logic. This means that many knowledge modelling situations, such as for non-monotonic reasoning, can be better captured by conditional logics (such as [Delgrande, 1987; Kraus *et al.*, 1990; Arló-Costa and Shapiro, 1992]).

6 Argument graphs

We now investigate options for instantiating argument graphs. We start with descriptive argument graphs, and then turn to generated argument graphs, using simple logic, classical logic, and conditional logic.

6.1 Descriptive graphs

For the descriptive approach to argument graphs, we assume that we have some abstract argument graph as the input, together with some informal description of each argument. For instance, when we listen to a debate on the radio, we may identify a number of arguments and counterarguments, and for each of these we may be able to write a brief text summary. So if we then want to understand this argumentation in more detail, we may choose to instantiate each argument with a deductive argument. So for this task we choose the appropriate logical formulae for the premises and claim for each argument (compatible with the choice of base logic). Examples of descriptive graphs are given in Figure 4 using simple logic, and in Example 6.1 and Figure 5 using classical logic.

Example 6.1 *Consider the following argument graph where A_1 is "The flight is low cost and luxury, therefore it is a good flight", and A_2 is "A flight cannot be both low cost and luxury".*

$$\boxed{A_1} \longleftarrow \boxed{A_2}$$

For this, we instantiate the arguments in the above abstract argument graph to give the following descriptive graph. So in the descriptive graph below, A_2 is a classical undercut to A_1.

$$\boxed{A_1 = \langle\{\texttt{lowCostFly}, \texttt{luxFly}, \texttt{lowCostFly} \wedge \texttt{luxFly} \rightarrow \texttt{goodFly}\}, \texttt{goodFly}\rangle}$$

$$\boxed{A_2 = \langle\{\neg(\texttt{lowCostFly} \wedge \texttt{luxFly})\}, \neg\texttt{lowCostFly} \vee \neg\texttt{luxFly}\rangle}$$

So for the approach of descriptive graphs, we do not assume that there is an automated process that constructs the graphs. Rather the emphasis is on having a formalization that is a good representation of the argumentation. This is so that we can formally analyze the descriptive graph, perhaps as part of some sense-making or decision-making process. Nonetheless, we can envisage that in the medium term natural language processing technology will be able to parse the text or speech (for instance in a discussion paper or in a debate) in order to automatically identify the premises and claim of each argument and counterargument.

Since we are primarily interested in representational and analytical issues when we use descriptive graphs, a richer logic such as classical logic is a more appealing formalism than a weaker base logic such as simple logic. Given a set of real-world arguments, it is often easier to model them using deductive arguments with classical logic as the base logic than a "rule-based logic" like simple logic as the base logic. For instance, in Example 6.1, the undercut does not claim that the flight is not low cost, and it does not claim that it is not luxury. It only claims that the flight cannot be both low cost *and* luxury. It is natural to represent this exclusion using disjunction.

As another example of the utility of classical logic as base logic, consider the importance of quantifiers in knowledge which require a richer language such as classical logic for reasoning with them. Moreover, if we consider that many arguments are presented in natural language (spoken or written), and that formalizing natural language often calls for a richer formalism such as classical logic (or even richer), it is valuable to harness classical logic in formalizations of deductive argumentation.

Conditional logics are also important formalisms for capturing some of the subtleties of natural language as they can reflect hypothetical statements, and can often provide a better representation of non-monotonic statements. We give an example of descriptive graph using conditional logic in Figure 6.

6.2 Generated graphs based on simple logic

Given a knowledgebase Δ, we can generate an argument graph $G = (\mathcal{A}, \mathcal{R})$ where $\mathcal{A}$ is the set of simple arguments obtained from Δ as follows and $\mathcal{R} \subseteq \mathcal{A} \times \mathcal{A}$ is simple undercut.

Definition 6.2 *Let Δ be a simple logic knowledgebase. A* **simple exhaustive graph** *for Δ is an argument graph $G = (\mathcal{A}, \mathcal{R})$ where $\mathcal{A}$ is $\mathsf{Arguments}_s(\Delta)$ and $\mathcal{R}$ is $\mathsf{Attacks}_s(\Delta)$ defined as follows*

$$\mathsf{Args}_s(\Delta) = \{\langle\Phi, \alpha\rangle \mid \Phi \subseteq \Delta \textit{ and } \langle\Phi, \alpha\rangle \textit{ is a simple argument } \}$$
$$\mathsf{Attacks}_s(\Delta) = \{(A, B) \mid A, B \in \mathsf{Args}_s(\Delta) \textit{ and } A \textit{ is a simple undercut of } B\}$$

This is an exhaustive approach to constructing an argument graph from a knowledgebase since all the simple arguments and all the simple undercuts are in the argument graph. We give an example of such an argument graph in Figure 7.

Simple exhaustive graphs provide a direct and useful way to instantiate argument graphs. There are various ways the definitions can be adapted, such as defining the attacks to be the union of the simple undercuts and the simple rebuts.

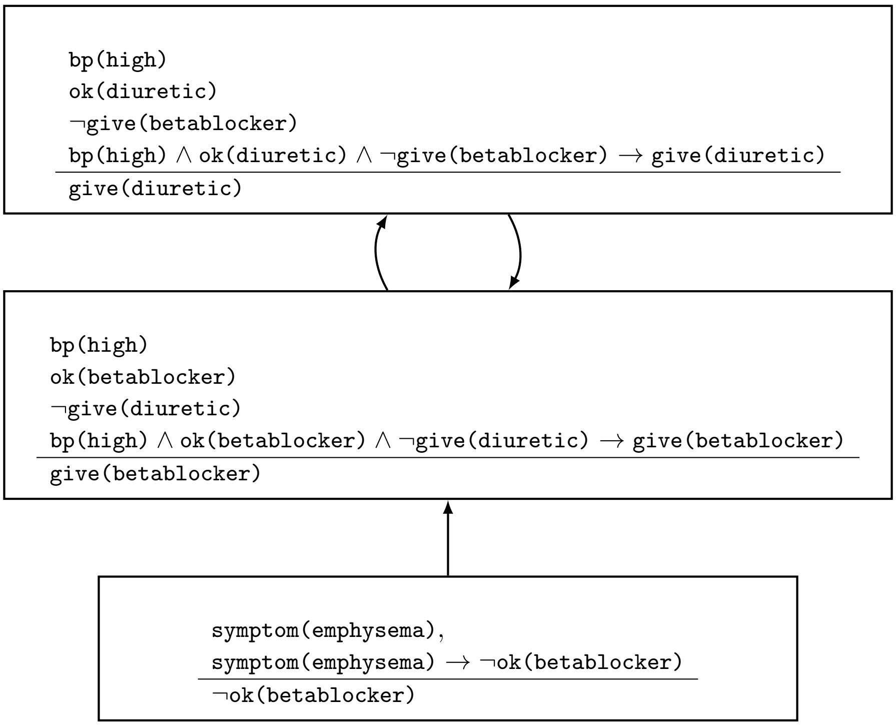


Figure 4. A descriptive graph representation of the abstract argument graph in Figure 2 using simple logic. The atom `bp(high)` denotes that the patient has high blood pressure. Each attack is a simple undercut by one argument on another. For the first argument, the premises include the assumptions `ok(diuretic)` and ¬`give(betablocker)` in order to apply its simple rule. Similarly, for the second argument, the premises include the assumptions `ok(betablocker)` and ¬`give(diuretic)` in order to apply its simple rule.

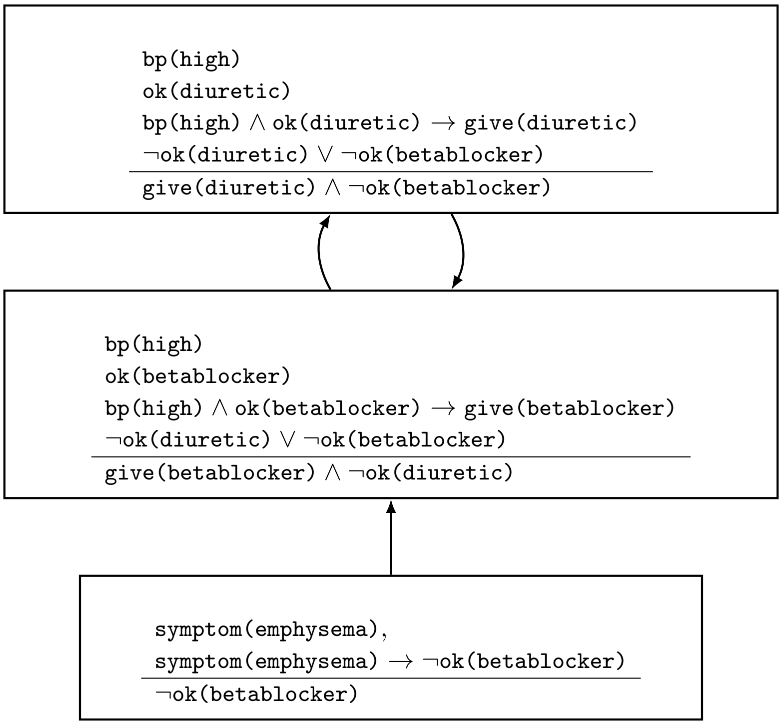


Figure 5. A descriptive graph representation of the abstract argument graph in Figure 2 using classical logic. The atom `bp(high)` denotes that the patient has high blood pressure. The top two arguments rebut each other (i.e. the attack is classical defeating rebut). For this, each argument has an integrity constraint in the premises that says that it is not ok to give both betablocker and diuretic. So the top argument is attacked on the premise `ok(diuretic)` and the middle argument is attacked on the premise `ok(betablocker)`. So we are using the `ok` predicate as a normality condition for the rule to be applied (as suggested in Section 5.1).

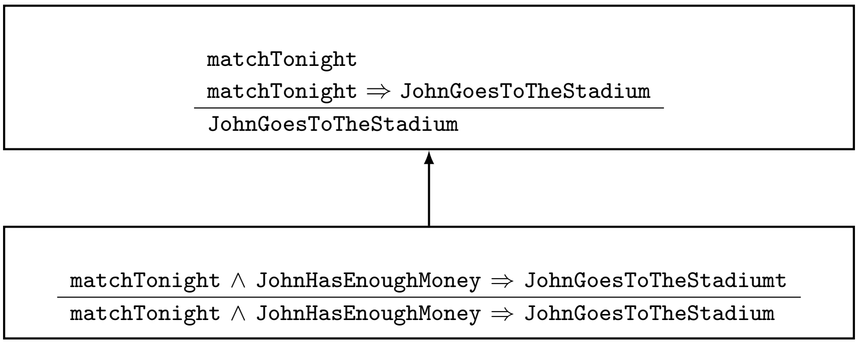


Figure 6. A descriptive graph that captures two arguments. The first argument says John will go to the stadium because there is a match tonight. The second corrects the first argument by correcting the circumstances under which John will go to watch the match at the stadium.

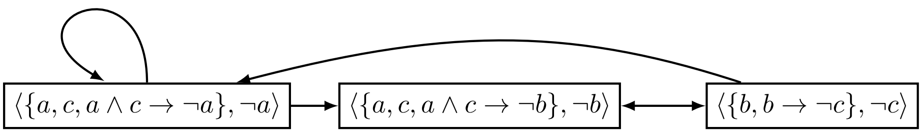


Figure 7. An exhaustive simple logic argument graph where $\Delta = \{a, b, c, a \wedge c \rightarrow \neg a, b \rightarrow \neg c, a \wedge c \rightarrow \neg b\}$. Note, that this exhaustive graph contains a self cycle, and an odd length cycle.

6.3 Generated graphs based on classical logic

In this section, we consider generated graphs for classical logic. We start with the classical exhaustive graphs which are the same as the simple exhaustive graphs except we use classical arguments and attacks. We show that whilst this provides a comprehensive presentation of the information, its utility is limited for various reasons. We then show that by introducing further information, we can address these shortcomings. To illustrate this, we consider a version of classical exhaustive graphs augmented with preference information. This is just one possibility for introducing extra information into the construction process.

6.3.1 Classical exhaustive graphs

Given a knowledgebase Δ, we can generate an argument graph $G = (\mathcal{A}, \mathcal{R})$ where A is the set of arguments obtained from Δ as follows and $\mathcal{R} \subseteq \mathcal{A} \times \mathcal{A}$ is one of the definitions for classical attack.

Definition 6.3 *Let Δ be a classical logic knowledgebase. A* **classical exhaustive graph** *is an argument graph $G = (\mathcal{A}, \mathcal{R})$ where $\mathcal{A}$ is $\mathsf{Arguments}_c(\Delta)$ and $\mathcal{R}$ is $\mathsf{Attacks}_c^X(\Delta))$ defined as follows where X is one of the attacks given in Definition 5.5 such as defeater, direct undercut, or rebuttal.*

$$\mathsf{Arguments}_c(\Delta) = \{\langle \Phi, \alpha \rangle \mid \Phi \subseteq \Delta \textit{ and } \langle \Phi, \alpha \rangle \textit{ is a classical argument } \}$$
$$\mathsf{Attacks}_c^X(\Delta) = \{(A, B) \in \mathsf{Arguments}_c(\Delta) \times \mathsf{Arguments}_c(\Delta) \mid A \textit{ is } X \textit{ of } B\}$$

This is a straightforward approach to constructing an argument graph from a knowledgebase since all the classical arguments and all the attacks (according to the chosen definition of attack) are in the argument graph as illustrated in Figure 8. However, if we use this exhaustive definition, we obtain infinite graphs, even if we use a knowledgebase with few formulae. For instance, if we have an argument $\langle \{a\}, a \rangle$, we also have arguments such as $\langle \{a\}, a \vee a \rangle$, $\langle \{a\}, a \vee a \vee a \rangle$, etc, as well as $\langle \{a\}, \neg\neg a \rangle$, etc.

Even though the graph is infinite, we can present a finite representation of it, by just presenting a representative of each class of structurally equivalent arguments (as considered in [Amgoud *et al.*, 2011]), where we say that two arguments A_i and A_j are **structurally equivalent** in $G = (\mathcal{A}, \mathcal{R})$ when the following conditions are satisfied: (1) if A_k attacks A_i, then A_k attacks A_j; (2) if A_k attacks A_j, then A_k attacks A_i; (3) if A_i attacks A_k, then A_j attacks A_k; and (4) if A_i attacks A_k, then A_j attacks A_k. For example, in Figure 8, the argument A_4 is a representative for $\langle \{b\}, b \rangle$, $\langle \{b\}, a \vee b \rangle$, $\langle \{b\}, \neg a \vee b \rangle$, etc.

We can also ameliorate the complexity of classical exhaustive graphs by presenting a focal graph (as discussed in Section 2). We illustrate this in Figure 9.

To conclude our discussion of classical exhaustive graphs, the definitions ensure that all the ways that the knowledge can be used to generate classical arguments and classical counterarguments (modulo the choice of attack relation) are laid out. This may involve many arguments being presented. This

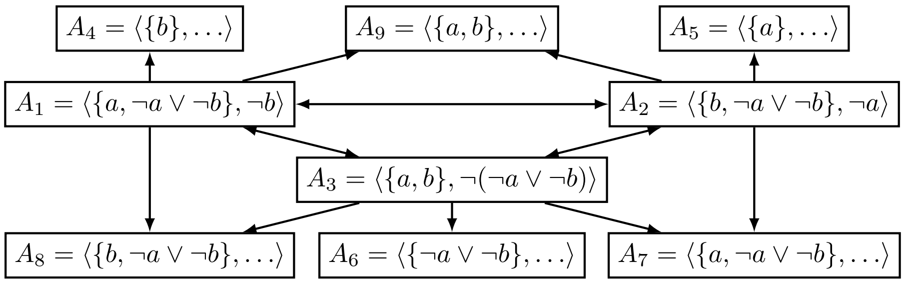


Figure 8. An exhaustive classical logic argument graph where $\Delta = \{a, b, \neg a \vee \neg b\}$ and the attack is direct undercut. Note, argument A_4 represents all arguments with a claim that is implied by b, argument A_5 represents all arguments with a claim that is implied by a, argument A_6 represents all arguments with a claim that is implied by $\neg a \vee \neg b$, argument A_7 represents all arguments with a claim that is implied by $a \wedge \neg b$ except $\neg b$ or any claim implied by a or any claim implied by $\neg a \vee \neg b$, argument A_8 represents all arguments with a claim that is implied by $\neg a \wedge b$ except $\neg a$ or any claim implied by b or any claim implied by $\neg a \vee \neg b$, and argument A_9 represents all arguments with a claim that is implied by $a \wedge b$ except $\neg(\neg a \vee \neg b)$ or any claim implied by a or any claim implied by b.

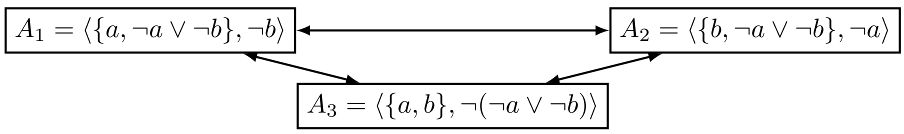


Figure 9. Focal graph formed from Figure 8 where focus is $\{A_1, A_2\}$.

can be addressed by the generation process discriminating between the arguments (and/or the attacks) based on extra information about the arguments and/or information about the audience. There are many ways that this can be done. In the next section, we consider a simple proposal for augmenting the generation process with preferences over arguments.

6.3.2 Preferential exhaustive graphs

One of the first proposals for capturing the idea of preferences in constructing argument graphs was preference-based argumentation frameworks (PAF) by [Amgoud and Cayrol, 2002]. This generalizes Dung's definition for an argument graph by introducing a preference relation over arguments that in effect causes an attack to be ignored when the attacked argument is preferred over the attacker. So in PAF, we assume a preference relation over arguments, denoted $\preceq$, as well as a set of arguments $\mathcal{A}$ and an attack relation $\mathcal{R}$. From this, we need to define a defeat relation $\mathcal{D}$ as follows, and then $(\mathcal{A}, \mathcal{D})$ is used as the argument graph, instead of $(\mathcal{A}, \mathcal{R})$, with Dung's usual definitions for extensions.

$$\mathcal{D} = \{(A_i, A_j) \in \mathcal{R} \mid (A_j, A_i) \not\in \preceq\}$$

So with this definition for defeat, extensions for a preference-based argument graph $(\mathcal{A}, \mathcal{R}, \preceq)$ can be obtained as follows: For S denoting complete, preferred, stable or grounded semantics, $\Gamma \subseteq \mathcal{A}$, Γ is an extension of $(\mathcal{A}, \mathcal{R}, \preceq)$ w.r.t. semantics S iff Γ is an extension of $(\mathcal{A}, \mathcal{D})$ w.r.t. semantics S.

We now revise the definition for classical exhaustive graphs to give the following definition for preferential exhaustive graphs.

Definition 6.4 *Let Δ be a classical logic knowledgebase. A* **preferential exhaustive graph** *is an argument graph* $(\mathsf{Arguments}_c(\Delta), \mathsf{Attacks}^X_{c,\preceq}(\Delta))$ *defined as follows where X is one of the attacks given in Definition 5.5 such as defeater, direct undercut, or rebuttal.*

$$\mathsf{Arguments}_c(\Delta) = \{\langle\Phi, \alpha\rangle \mid \Phi \subseteq \Delta \;\&\; \langle\Phi, \alpha\rangle \textit{ is a classical argument }\}$$
$$\mathsf{Attacks}^X_{c,\preceq}(\Delta) = \{(A, B) \mid A, B \in \mathsf{Arguments}_c(\Delta) \;\&\; A \textit{ is } X \textit{ of } B \;\&\; (B, A) \not\in \preceq\}$$

We give an illustration of a preferential exhaustive graph in Figure 10, and we give an illustration of a focal graph obtained from a preferential exhaustive graph in Example 6.5.

Example 6.5 *This example concerns two possible treatments for glaucoma caused by raised pressure in the eye. The first is a prostoglandin analogue (PGA) and the second is a betablocker (BB). Let Δ contain the following six formulae. The atom $\mathtt{p}_1$ is the fact that the patient has glaucoma, the atom $\mathtt{p}_2$ is the assumption that it is ok to give PGA, and the atom $\mathtt{p}_3$ is the assumption that it is ok to give BB. Each implicational formula (i.e. $\mathtt{p}_4$ and $\mathtt{p}_5$) captures the knowledge that if a patient has glaucoma, and it is ok to give a particular*

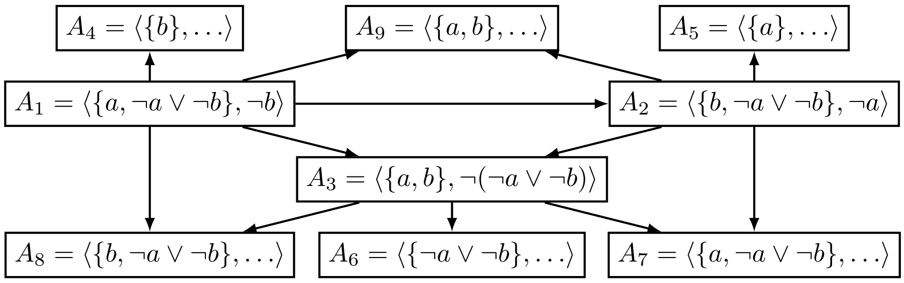


Figure 10. The preferential exhaustive graph where the knowledgesbase is $\Delta = \{a, b, \neg a \vee \neg b\}$ and the attack is direct undercut. This is the same knowledgebase and attack relation as in Figure 8. For the preference relation, A_1 is preferred to all other arguments, A_2 is preferred to all other arguments apart from A_1, and the remaining arguments are equally preferred. So for all i such that $i \neq 1$, $A_1 \prec A_i$, and for all i such that $i \neq 1$ and $i \neq 2$, $A_2 \prec A_i$. This results in three attacks in Figure 8 not appearing in this graph. The dropped attacks are A_2 on A_1, A_3 on A_1, and A_3 on A_1.

drug, then give that drug. Formula $\mathtt{p}_6$ *is an integrity constraint that ensures that only one treatment is given for the condition.*

$$\begin{array}{ll}
\mathtt{p}_1 = \mathtt{glaucoma} & \mathtt{p}_4 = \mathtt{glaucoma} \wedge \mathtt{ok(PGA)} \rightarrow \mathtt{give(PGA)} \\
\mathtt{p}_2 = \mathtt{ok(PGA)} & \mathtt{p}_5 = \mathtt{glaucoma} \wedge \mathtt{ok(BB)} \rightarrow \mathtt{give(BB)} \\
\mathtt{p}_3 = \mathtt{ok(BB)} & \mathtt{p}_6 = \neg\mathtt{ok(PGA)} \vee \neg\mathtt{ok(BB)}
\end{array}$$

There are numerous arguments that can be constructed from this set of formulae such as the following.

$$\begin{array}{ll}
A_1 = \langle\{\mathtt{p}_1, \mathtt{p}_2, \mathtt{p}_4, \mathtt{p}_6\}, \mathtt{give(PGA)} \wedge \neg\mathtt{ok(BB)}\rangle & A_5 = \langle\{\mathtt{p}_1, \mathtt{p}_2, \mathtt{p}_4\}, \mathtt{give(PGA)}\rangle \\
A_2 = \langle\{\mathtt{p}_1, \mathtt{p}_3, \mathtt{p}_5, \mathtt{p}_6\}, \mathtt{give(BB)} \wedge \neg\mathtt{ok(PGA)}\rangle & A_6 = \langle\{\mathtt{p}_1, \mathtt{p}_3, \mathtt{p}_5\}, \mathtt{give(BB)}\rangle \\
A_3 = \langle\{\mathtt{p}_2, \mathtt{p}_3\}, \mathtt{ok(PGA)} \wedge \mathtt{ok(BB)}\rangle & A_7 = \langle\{\mathtt{p}_2, \mathtt{p}_6\}, \neg\mathtt{ok(BB)}\rangle \\
A_4 = \langle\{\mathtt{p}_6\}, \neg\mathtt{ok(PGA)} \vee \neg\mathtt{ok(BB)}\rangle & A_8 = \langle\{\mathtt{p}_3, \mathtt{p}_6\}, \neg\mathtt{ok(PGA)}\rangle
\end{array}$$

Let $\mathsf{Arguments}_c(\Delta)$ *be the set of all classical arguments that can be constructed from* Δ*, and let the preference relation* $\preceq$ *be such that* $A_i \preceq A_1$ *and* $A_i \preceq A_2$ *for all* i *such that* $i \neq 1$ *and* $i \neq 2$*. Furthermore, let* $\Pi = \{A_1, A_2\}$ *be the focus (i.e. the arguments of interest). In other words, we know that each of these two arguments in the focus contains all the information we are interested in (i.e. we want to determine the options for treatment taking into account the integrity constraint). This would give us the following focal graph using the classical direct defeater definition for attack.*

$$
\begin{tikzcd}
\boxed{A_1 = \langle \{\mathtt{p}_1, \mathtt{p}_2, \mathtt{p}_4, \mathtt{p}_6\}, \mathtt{give(PGA)} \land \neg \mathtt{ok(BB)} \rangle} \arrow[d, bend left] \\
\boxed{A_2 = \langle \{\mathtt{p}_1, \mathtt{p}_3, \mathtt{p}_5, \mathtt{p}_6\}, \mathtt{give(BB)} \land \neg \mathtt{ok(PGA)} \rangle} \arrow[u, bend left]
\end{tikzcd}
$$

By taking this focal graph, we have ignored arguments such as A_3 to A_8 which do not affect the dialectical status of A_1 or A_2 given this preference relation.

Using preferences is a general approach. There is no restriction on what preference relation we use over arguments, and there are various natural interpretations for this ranking such as capturing belief for arguments (where the belief in the argument can be based on the belief for the premises and/or claim), and capturing the relative number of votes for arguments (where a group of voters will vote for or against each argument), etc.

To conclude, by introducing preferences over arguments, we can reduce the number of attacks that occur. Using preferences over arguments is a form of meta-information, and with the definition for preference-based argumentation (as defined by [Amgoud and Cayrol, 2002]), it supports selectivity in generating argument graphs that discriminates between arguments and thereby between attacks. With this definition more practical argument graphs can be constructed than with the definition for classical exhaustive graphs.

7 Properties of deductive arguments

In this section, we consider some properties of argumentation based on deductive arguments. Out focus is predominantly on classical logic. We consider postulates for counterarguments (i.e. for the attack relation), postulates for extensions, and properties of the structure of generated graphs.

7.1 Counterargument properties

We consider some desirable properties of attack relations in the form of postulates and classify several well-known attack relations from the literature with regards to the satisfaction of these postulates. We define these postulates in terms of function D where $D(A, B) = \top$ holds iff A attacks B. Different definitions of counterargument, give is different definitions of attack. So for example, if A is a defeater of B, then this denoted by $D_D(A, B) = \top$.

In Table 2, we review postulates relevant to attack functions. From now on, A, B, C and their primed versions will stand for arguments. We explain them as follows:

- (D0) This is a classic syntax-independence requirement: the syntax of the components of two arguments should not play a role in deciding whether there is an attack between those arguments;

- (D1) This mandates that if an argument attacks another, then it must be that the claim of the former is inconsistent with the support of the

Name	Property
D0	if $A \equiv A', B \equiv B'$ then $D(A, B) = D(A', B')$
D1	if $D(A, B) = \top$ then $\{\mathsf{Claim}(A)\} \cup \mathsf{Support}(B) \vdash \bot$
D2	if $D(A, B) = \top$ and $\mathsf{Claim}(C) \equiv \mathsf{Claim}(A)$ then $D(C, B) = \top$
D2i	if $D(A, B) = \top$ and $\mathsf{Claim}(C) \vdash \mathsf{Claim}(A)$ then $D(C, B) = \top$
D3	if $D(A, B) = \top$ and $\mathsf{support}(B) = \mathsf{support}(C)$ then $D(A, C) = \top$
D3i	if $D(A, B) = \top$ and $\mathsf{support}(B) \subseteq \mathsf{support}(C)$ then $D(A, C) = \top$
D4	if $\mathsf{Arcs}(G) = \emptyset$ then $\mathsf{MinIncon}(\Delta) = \emptyset$

Table 2. Postulates for an attack relation D. We denote an attack relation from A to B as holding when $D(A, B) = \top$.

latter. This requirement reflects a fundamental assumption in logical argumentation, namely that if two arguments are logically consistent there cannot be any attack between them.

- (D2) This imposes a certain fairness restrictions on existing attacks by requiring that all arguments that have equivalent claims with that of A should attack B.
- (D2i) is a strengthening of D2. It requires that any argument with a stronger claim than A, i.e., one that logically entails that of A, should also attack anything A attacks
- (D3) This requires that if A attacks B then all arguments with the same support with that of B should also be attacked by A.
- (D3i) is a strengthening of D3 proposed by Amgoud and Besnard [Amgoud and Besnard, 2009]. It mandates that any argument whose support is a superset of that of B, and thus is stronger than that of B, should also be attacked by A.
- (D4) This postulate can be read as follows: if we restrict D on the arguments that can be generated from Δ and find that no two such arguments attack each other, then it must be that Δ itself is consistent (hence it has no minimal inconsistent subsets).

Given the postulates in Table 2, we can classify the notions of counterargument given for classical logic. This classification is given in Table 3,

If we then impose further constraints as listed in Table 4, such as constraints on the claim of the attacker D1i or D1ii, and D5 to D5iii, and constraints forcing the existence of attacks D6 to D6iii, then we obtain the following proposition provides characterization results for the classical attack relations. This means we have alternative definitions for our attack relation that are specified entirely in terms of a set of properties. For the proofs, see [Gorogiannis and Hunter, 2011].

	D_D	D_{DD}	D_U	D_{DU}	D_{CU}	D_R	D_{DR}
D0	Yes	Yes	Yes	Yes	Yes	Yes	Yes
D1	Yes	Yes	Yes	Yes	Yes	Yes	Yes
D2	Yes	Yes	Yes	Yes	Yes	Yes	Yes
D2i	Yes	Yes	No	No	No	No	Yes
D3	Yes	Yes	Yes	Yes	Yes	No	No
D3i	Yes	Yes	Yes	Yes	No	No	No
D4	Yes	Yes	Yes	Yes	Yes	Yes	Yes

Table 3. Postulates satsified by attack functions

Name	Property
D1i	if $D(A,B) = \top$ then $\exists\phi \in \mathsf{Support}(B)$ s.t. $\mathsf{Claim}(A) \vdash \neg\phi$
D1ii	if $D(A,B) = \top$ then $\mathsf{Claim}(A) \vdash \neg\mathsf{Claim}(B)$
D5	if $D(A,B) = \top$ then $\neg\mathsf{Claim}(A) \vdash \bigwedge \mathsf{Support}(B)$
D5i	if $D(A,B) = \top$ then $\exists\phi \in \mathsf{Support}(B)$ s.t. $\neg\mathsf{Claim}(A) \vdash \phi$
D5ii	if $D(A,B) = \top$ then $\neg\mathsf{Claim}(A) \vdash \mathsf{Claim}(B)$
D5iii	if $D(A,B) = \top$ then $\exists X \subseteq \mathsf{Support}(B)$ s.t. $\neg\mathsf{Claim}(A) \equiv \bigwedge X$
D6	if $\{\mathsf{Claim}(A)\} \cup \mathsf{Support}(B) \vdash \bot$ then there exists C s.t. $\mathsf{Claim}(A) \vdash \mathsf{Claim}(C)$ and $D(C,B) = \top$
D6i	if $\exists\phi \in \mathsf{Support}(B)$ s.t. $\mathsf{Claim}(A) \vdash \neg\phi$ then there exists C s.t. $\mathsf{Claim}(A) \vdash \mathsf{Claim}(C)$ and $D(C,B) = \top$
D6ii	if $\mathsf{Claim}(A) \vdash \neg\mathsf{Claim}(B)$ then there exists C s.t. $\mathsf{Claim}(A) \vdash \mathsf{Claim}(C)$ and $D(C,B) = \top$
D6iii	if $\exists X \subseteq \mathsf{Support}(B)$ s.t. $\mathsf{Claim}(A) \equiv \neg\bigwedge X$ then $D(A,B) = \top$

Table 4. Further constraints on the attack relation.

Proposition 7.1 *Let D be an attack relation.*

- *$D = D_D$ is a defeater relation iff D satisfies D1, D2i and D6*
- *$D = D_{DD}$ is a direct defeater relation iff D satisfies D1i, D2i and D6i*
- *$D = D_{DR}$ is a defeating rebuttal relation iff D satisfies D1ii, D2i and D6ii*
- *$D = D_{CU}$ is a canonical undercut relation iff D satisfies D1, D2, D5 and D6*
- *$D = D_{DU}$ is a direct undercut relation iff D satisfies D1i, D2, D5i and D6i*
- *$D = D_R$ is a rebuttal relation iff D satisfies D1ii, D2, D5ii and D6ii*
- *$D = D_U$ is an undercut relation iff D satisfies D5iii and D6iii*

Since classical logic offers a variety of different options for defining a counterargument (i.e. an attack relation), it is helpful to characterize the options in terms of postulates. Furthermore, these postulates can be used or adapted for classifying and characterizing attack relations for a variety of base logics.

7.2 Extension properties

Various postulates have been proposed for classical exhaustive graphs (e.g. [Gorogiannis and Hunter, 2011]). Some of these are concerned with consistency of the set of premises (or set of claims) obtained from the arguments in an extension according to one of Dung's dialectical semantics. In the rest of this subsection, we restrict consideration to classical logic arguments, though the postulates can be adapted for other base logics.

To consider extension properties, we will introduce some subsidiary definitions. We start with sceptical acceptance of arguments and credulous acceptance of arguments defined as follows where G is an argument graph and $Y \in \{\mathsf{pr}, \mathsf{gr}, \mathsf{st}\}$ is a dialectical semantics (where pr denotes preferred, gr denotes grounded, and st denotes stable). and $\mathsf{Extensions}_Y(G)$ is the set of extensions obtained according to Y. If $\mathsf{Extensions}_Y(G) = \emptyset$, then $\mathsf{Sceptical}_Y(G) = \mathsf{Credulous}_Y(G) = \emptyset$, otherwise

$$\mathsf{Sceptical}_Y(G) = \bigcap_{S \in \mathsf{Extension}_Y(G)} S$$

$$\mathsf{Credulous}_Y(G) = \bigcup_{S \in \mathsf{Extension}_Y(G)} S$$

So, for example, we will say that an argument A in $\mathsf{Nodes}(G)$ is sceptically accepted in the preferred semantics if $A \in \mathsf{Sceptical}_{\mathsf{pr}}(G)$. Clearly, we have the following observations where $Y \in \{\mathsf{pr}, \mathsf{gr}, \mathsf{st}\}$

- $\mathsf{Sceptical}_Y(G) \subseteq \mathsf{Credulous}_Y(G)$
- $\mathsf{Credulous}_{gr}(G) = \mathsf{Sceptical}_{gr}(G)$

The definition of an exhaustive argument graph takes an attack function D and a knowledgebase Δ in order to produce the argument graph. Such an argument graph can be evaluated with choices for dialectical semantics (preferred, grounded, etc) and acceptability criteria (sceptical or credulous).

Since the arguments are logical, we can evaluate the logical properties of the extensions. We will review the free postulate, non-free postulate, and consistency postulates, in the rest of this section. For this, we require the following definition for the free formulae which is the set of formulae not in any minimal inconsistent subset of Δ.

$$\mathsf{Free}(\Delta) = \{\alpha \in \Delta \mid \alpha \notin \bigcup_{\Gamma \in \mathsf{MinIncon}(\Delta)} \Gamma\}$$

where

$$\mathsf{MinIncon}(\Delta) = \{\Gamma \subseteq \Delta \mid \Gamma \vdash \bot \text{ and for all } \Gamma' \subset \Gamma, \Gamma' \not\vdash \bot\}$$

We identify the free arguments in a graph (i.e. the arguments with no premises involved in a minimal inconsistent subset of the knowledgebase) and the non-free arguments in a graph (i.e. the arguments with one or more premise involved in a minimal inconsistent subset of the knowledgebase) as follows.

$$\mathsf{FreeArguments}(G) = \{A \in \mathsf{Nodes}(G) \mid \mathsf{Support}(A) \subseteq \mathsf{Free}(\Delta)\}$$

$$\mathsf{NonFreeArguments}(G) = \{A \in \mathsf{Nodes}(G) \mid \mathsf{Support}(A) \not\subseteq \mathsf{Free}(\Delta)\}$$

Our first extension-based postulate is the free postulate (defined next) states that the free arguments are sceptical arguments (i.e. in all extensions of the graph). This encodes our expectation that since free arguments are uncontroversial, they should be in every extension.

Definition 7.2 *For a descriptive or generated argument graph G based on classical logic arguments, the* **free postulate** *is defined as follows, where $Y \in \{\mathsf{pr}, \mathsf{gr}, \mathsf{st}\}$.*

$$\mathsf{FreeArguments}(G) \subseteq \mathsf{Sceptical}_Y(G)$$

For the proof of the following proposition, see [Gorogiannis and Hunter, 2011].

Proposition 7.3 *If the attack relation D satisfies $D1$, then D satisfies the free postulate.*

All the attack functions considered for classical logic in this paper satisfy $D1$, and therefore satisfy the free postulate. Therefore, for all semantics considered, all extensions of G contain all free arguments.

Next, we define the non-free postulate. This states that there exists a knowledgebase that is inconsistent and for which some arguments are *not* credulously accepted.

Definition 7.4 *For a descriptive or generated argument graph G based on classical logic arguments, the* **non-free postulate** *is defined as follows, where* $Y \in \{\mathsf{pr}, \mathsf{gr}, \mathsf{st}\}$.

$$\begin{array}{l} \exists \Delta \ s.t. \ \bigcup_{A \in \mathsf{Nodes}(G)} \mathsf{Support}(A) \subseteq \Delta \\ \qquad\qquad and \ \mathsf{Nonfree}(G) \neq \emptyset \ and \ \mathsf{Credulous}_X(G) \neq \mathsf{Nodes}(G) \end{array}$$

Failure means that if $\mathsf{Support}(A) \subseteq \Delta$, then A is credulously accepted. So for any argument that can be formed from a knowledgebase, there is a preferred extension that contains that argument. So if it does fail for an attack function D and a dialectical semantics Y, then this indicates that the combination of D and Y is, in a sense, very credulous. For the proof of the following proposition, see [Gorogiannis and Hunter, 2011].

Proposition 7.5 *We consider the non-free postulate with respect to the semantics where the attack relation is undercut, direct undercut, canonical undercut, or rebuttal.*

- *For stable, preferred, and complete extensions, the non-free postulate is not satisfied.*
- *For grounded extensions, the non-free postulate is satisfied.*

Finally, we consider the consistency postulates. These postulates are variations of the requirement that certain arguments' supports or claims must be consistent together. The expectation is that once an extension is obtained, then the arguments contained in it present a somehow consistent set of assumptions. Applying this restriction to the supports of the arguments or to their claims, and to the sceptically accepted set of arguments or to all extensions individually, yields the versions of this principle listed below.

Definition 7.6 *For a descriptive or generated argument graph G based on classical logic arguments, the* **consistency postulates** *are defined as follows,*

Attack	CN1	CN1'	CN2	CN2'
Direct undercut	Yes	Yes	Yes	Yes
Direct defeat	Yes	Yes	Yes	Yes
Canonical undercut	Yes	Yes	Yes	Yes
Rebut	No	No	No	No

Table 5. Satisfaction of consistency postulates for grounded semantics

Attack	CN1	CN1'	CN2	CN2'
Direct undercut	Yes	Yes	Yes	Yes
Direct defeat	Yes	Yes	Yes	Yes
Canonical undercut	Yes	Yes	No	No
Rebut	No	No	No	No

Table 6. Satisfaction of consistency postulates for preferred and complete semantics

where $Y \in \{\mathsf{pr}, \mathsf{gr}, \mathsf{st}\}$.

$$(CN1) \quad \bigcup_{A \in \mathsf{sceptical}_Y(G)} \mathsf{Support}(A) \not\vdash \bot$$

$$(CN2) \bigcup_{A \in S} \mathsf{Support}(A) \not\vdash \bot, \textit{for all } S \in \mathsf{Extension}_Y(G)$$

$$(CN1') \quad \bigcup_{A \in \mathsf{Sceptical}_Y(G)} \mathsf{Claim}(A) \not\vdash \bot$$

$$(CN2') \bigcup_{A \in S} \mathsf{Claim}(A) \not\vdash \bot, \ \textit{for all } S \in \mathsf{Extension}_Y(G)$$

The reason we provide all four versions of the consistency postulates is that it is not yet clear whether one form of the postulate is more appropriate than others. For example, consistency postulates similar to CN1' and CN2' have been proposed in [Caminada and Amgoud, 2005] in the context of rule-based argumentation systems and versions of CN1 and CN2 have been proposed in [Amgoud and Besnard, 2009] for classical logics. It should be clear that CN2 entails CN1, CN2' entails CN1', CN1 entails CN1' and CN2 entails CN2'.

We summarize which attack relations satisfy which of the four consistency postulates in Table 7.2 for grounded semantics and in Table 7.2 for preferred, stable, and complete semantics. These results show that for some attack relation (e.g. rebuttal), the consistent extension property is not guaranteed (as in Example 7.7) whereas for other choices of attack relation (e.g. direct undercut), the consistent extension property is guaranteed. We illustrate a consistent set of premises obtained from arguments in a preferred extension in Example 7.8.

Example 7.7 *Let $\Delta = \{a \wedge b, \neg a \wedge c\}$. For the reviewed semantics for rebut, the following are arguments in any extension: $A_1 = \langle \{a \wedge b\}, b\rangle$ and $A_2 = \langle \{\neg a \wedge c\}, c\rangle$. Clearly $\{a \wedge b, \neg a \wedge c\} \vdash \bot$. Hence, the consistent extensions property fails for rebut.*

Example 7.8 *Consider the argument graph given in Figure 8. There are three preferred extensions $\{A_1, A_5, A_6, A_7\}$, $\{A_2, A_4, A_6, A_8\}$, and $\{A_3, A_4, A_5, A_9\}$. In each case, the union of the premises is consistent. For instance, for the first extension,*

$$\mathsf{Support}(A_1) \cup \mathsf{Support}(A_5) \cup \mathsf{Support}(A_6) \cup \mathsf{Support}(A_7) \not\vdash \bot$$

Example 7.9 *Consider the argument graph given in Figure 9. There are three preferred extensions $\{A_1\}$, $\{A_2\}$, and $\{A_3\}$. In each case, the union of the premises is consistent.*

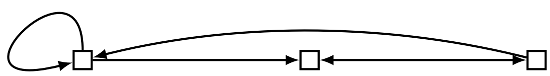
The failure of the consistency postulates with some attack relations is an issue that may be interpreted as a weakness of the attack relation or of the specific semantics, and perhaps raises the need for alternatives to be identified. Another response is that it is not the attack relation and dialectical semantics that should be responsible for ensuring that all the premises used in the winning arguments are consistent together. Rather, it could be argued that checking that the premises used are consistent together should be the responsibility of something external to the defeat relation and dialectical semantics, and so knowing whether the consistent extensions property holds or not influences what external mechanisms are required for checking. Furthermore, checking consistency of premises of sets of arguments may be part of the graph construction process. For instance, in Garcia and Simari's proposal for dialectical trees [García and Simari, 2004], there are constraints on what arguments can be added to the tree based on consistency with the premises of other arguments in the tree.

7.3 Structural properties

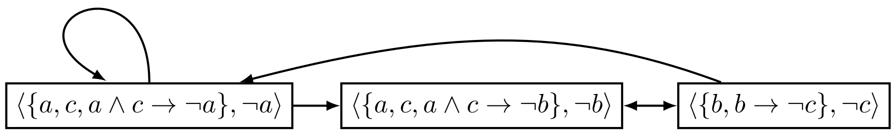
Simple logic has the property that for any argument graph, there is a knowledgebase that can be used to generate it: Let (N, E) be a directed graph (i.e. N is a set of nodes, and E is a set of edges between nodes), then there is a simple logic knowledgebase Δ such that the generated argument graph $(\mathsf{Arguments}_s(\Delta), \mathsf{Attacks}_s(\Delta))$ is isomorphic to (N, E). So simple exhaustive graphs are said to be **constructively complete** for graphs.

To show that simple exhaustive graphs are constructively complete for graphs, we can use a coding scheme for the premises so that each argument is based on a single simple rule where the antecedent is a conjunction of one or more positive literals, and each consequent is a negative literal unique to that simple rule (i.e. it is an identifier for that rule and therefore for that argument). If we want one argument to attack another, and the attacking argument has the consequent $\neg\alpha$, then the attacked argument needs to have the positive literal α in the antecedent of its simple rule. The restriction of each rule to only have positive literals as conditions in the antecedent, and a negative literal as its consequent, means that the rules cannot be chained. This ensures that the premises of each argument has only one simple rule. We illustrate this in the following example.

Example 7.10 *Consider the following directed graph (N, E). Note, that it includes a self-attack, bidirectional attacks and uni-directional attacks.*



Let $\Delta = \{a, b, c, a \wedge c \rightarrow \neg a, a \wedge c \rightarrow \neg b, b \rightarrow \neg c\}$. From this we can construct the following exhaustive argument graph which is isomorphic to the above directed graph. Note that each argument is identified by a single simple rule.



In contrast to simple logic, the definition for classical exhaustive graphs (i.e. classical logic, with any of the definitions for counterarguments), is not constructively complete for graphs. Since the premises of a classical argument are consistent, by definition, it is not possible for a classical argument to attack itself using the definitions for attack given earlier. But, there are many other graphs for which there is no classical logic knowledgebase that can be used to generate a classical exhaustive graph that is isomorphic to it. To illustrate this failure, we consider in Example 7.11 the problem of constructing a component with two arguments attacking each other. Note, this is not a pathological example as there are many graphs that contain a small number of nodes and that cannot be generated as a classical exhaustive graph.

Example 7.11 *Let $\Delta = \{a, \neg a\}$ be a classical logic knowledgebase. Hence, there are two classical arguments A_1 and A_2 that are direct undercuts of each other. Plus, there is the representative A_4 for arguments with a claim that is strictly weaker than a (i.e. the claim b is such that $\{a\} \vdash b$ and $\{b\} \not\vdash \{a\}$),* *and there is the representative A_3 for arguments with a claim that is strictly weaker than $\neg a$ (i.e. the claim b is such that $\{\neg a\} \vdash b$ and $\{b\} \nvdash \{\neg a\}$).*

$$\begin{tikzcd} \boxed{A_3 = \langle \{\neg a\}, \ldots \rangle} & \boxed{A_1 = \langle \{a\}, a \rangle} \arrow[l] \arrow[r, bend left] & \boxed{A_2 = \langle \{\neg a\}, \neg a \rangle} \arrow[l, bend left] \arrow[r] & \boxed{A_4 = \langle \{a\}, \ldots \rangle} \end{tikzcd}$$

Given a set of directed graphs Φ we can define further properties. The set of directed graphs can be based on well-known definitions such as the set of bipartite graphs, the set of acyclic graphs, or the set of trees, or it can be defined in a domain specific way. A deductive argumentation system (i.e. a base logic, a definition for arguments, a definition for counterarguments, and a definition for a generated graphs) can then be evaluated with respect to Φ. We give these properties where we consider systems defined in terms of definitions for $\texttt{Arguments}(\Delta)$ (i.e. the definition for arguments that can be obtained from a knowledgebase Δ) and $\texttt{Attacks}(\Delta)$ (i.e. the definition for attacks that can be obtained from a knowledgebase Δ) with exhaustive graphs.

- A system $(\texttt{Arguments}, \texttt{Attacks})$ **constructively covers** Φ iff for all $G \in X$, there is a Δ and there is an $A \in \texttt{Arguments}(\Delta)$, such that $(\texttt{Arguments}(\Delta), \texttt{Attacks}(\Delta)) = G$.
- A system $(\texttt{Arguments}, \texttt{Attacks})$ is **constructively covered** by Φ iff for all Δ and for all $A \in \texttt{Arguments}(\Delta)$, if $(\texttt{Arguments}(\Delta), \texttt{Attacks}(\Delta)) = G$ then $G \in \Phi$.
- A system $(\texttt{Arguments}, \texttt{Attacks})$ is **constructively complete** for Φ iff $(\texttt{Arguments}, \texttt{Attacks})$ constructively covers Φ and $(\texttt{Arguments}, \texttt{Attacks})$ is constructively covered by Φ

The more general the class of graphs that a logical argument system can cover, the wider the range of argumentation situations the logical argument systems can capture. If one of these properties holds for a class of graphs, then it can be described as a kind of structural property of the system. If it fails then, it means that there are situations that cannot be captured by the system. This is, however, not necessarily bad news. In fact, it is known that the computational complexity of evaluating argumentation frameworks can be decreased if the class of graphs is restricted, for instance to acyclic, bipartite or symmetric graphs or to graphs which have certain parameters (like treewidth) fixed (see for example [Coste-Marquis *et al.*, 2005; Dunne, 2007; Dvorák *et al.*, 2012b; Dvorák *et al.*, 2012a]).

7.4 Discussion of properties

In this section, we have considered a range of properties of argumentation based on deductive arguments. For some choices of attack relation, there is a question of consistency (which may be an issue if no further consistency checking is undertaken). Also, the definition for classical exhaustive graphs is not constructively complete for graphs (which means that many argument graphs cannot be generated as classical exhaustive graphs). Perhaps more problematical is that even for small knowledgebases, the classical exhaustive graphs that are generated are complex. Because of the richness of classical logic, the knowledge can be in different combinations to create many arguments. Whilst, we can ameliorate this problem by presenting argument graphs using a representative for a class of structurally equivalent arguments, and by using focal graphs, the graphs can still be large. What is evident from this is that there needs to be more selectivity in the process of generating argument graphs. The generation process needs to discriminate between the arguments (and/or the attacks) based on extra information about the arguments and/or information about the audience. There are many ways that this can be done.

8 Further reading

We provide further reading on formalization of deductive arguments and counterarguments, properties of exhaustive graphs, the importance of selectivity in generating argument graphs, and on automated reasoning.

8.1 Deductive arguments and counterarguments

There have been a number of proposals for deductive arguments using classical propositional logic [Cayrol, 1995; Besnard and Hunter, 2001; Amgoud and Cayrol, 2002; Gorogiannis and Hunter, 2011], classical predicate logic [Besnard and Hunter, 2005], description logic [Black *et al.*, 2009; Moguillansky *et al.*, 2010; Zhang *et al.*, 2010; Zhang and Lin, 2013], temporal logic [Mann and Hunter, 2008], simple (defeasible) logic [Governatori *et al.*, 2004; Hunter, 2010], conditional logic [Besnard *et al.*, 2013], and probabilistic logic [Haenni, 1998; Haenni, 2001; Hunter, 2013].

There has also been progress in understanding the nature of classical logic in computational models of argument. Various types of counterarguments have been proposed including rebuttals [Pollock, 1987; Pollock, 1992], direct undercuts [Elvang-Gøransson *et al.*, 1993; Elvang-Gøransson and Hunter, 1995; Cayrol, 1995], and undercuts and canonical undercuts [Besnard and Hunter, 2001]. In most proposals for deductive argumentation, an argument A is a counterargument to an argument B when the claim of A is inconsistent with the support of B. It is possible to generalize this with alternative notions of counterargument. For instance, with some common description logics, there is not an explicit negation symbol. In the proposal for argumentation with description logics, [Black *et al.*, 2009] used the description logic notion of *incoherence* to define the notion of counterargument: A set of formulae in a description logic is incoherent when there is no set of assertions (i.e. ground literals) that would be consistent with the formulae. Using this, an argument A is a counterargument to an argument B when the claim of A together with the support of B is incoherent.

Meta-arguments for deductive argumentation was proposed by [Wooldridge *et al.*, 2005], and the investigation of the representation of argument schemes in deductive argumentation was first proposed by [Hunter, 2008].

8.2 Properties of exhaustive argument graphs

In order to investigate how Dung's notion of abstract argumentation can be instantiated with classical logic, [Cayrol, 1995] presents results concerning stable extensions of argument graphs where the nodes are classical logic arguments, and the attacks are direct undercuts. As well as being the first paper to propose instantiating abstract argument graphs with classical arguments, it also showed how the premises in the arguments in the stable extension correspond to maximal consistent subsets of the knowledgebase, when the attack relation is direct undercut.

Insights into the options for instantiating abstract argumentation with classical logic can be based on postulates. [Amgoud and Besnard, 2009] have proposed a consistency condition and they examine special cases of knowledge bases and symmetric attack relations and whether consistency is satisfied in this context. Then [Amgoud and Besnard, 2010] extend this analysis by showing correspondences between the maximal consistent subsets of a knowledgebase and the maximal conflict-free sets of arguments.

Given the wide range of options for attack in classical logic, [Gorogiannis and Hunter, 2011] propose a series of desirable properties of attack relations to classify and characterize attack relations for classical logic. Furthermore, they present postulates regarding the logical content of extensions of argument graphs that may be constructed with classical logic, and a systematic study is presented of the status of these postulates in the context of the various combinations of attack relations and extension semantics.

Use of the notion of generated graphs then raises the question of whether for a specific logical argument system S, and for any graph G, there is a knowledgebase such that S generates G. If it holds, then it can be described as a kind of "structural" property of the system [Hunter and Woltran, 2013]. If it fails then, it means that there are situations that cannot be captured by the system. The approach of simple exhaustive graphs is constructively complete for graphs, whereas the approach of classical exhaustive graphs is not.

Preferences have been introduced into classical logic argumentation, and used to instantiate abstract argumentation with preferences by [Amgoud and Cayrol, 2002]. Amgoud and Vesic [Amgoud and Vesic, 2010] have shown how preferences can be defined so as to equate inconsistency handling in argumentation with inconsistency handling using Brewka's preferred sub-theories [Brewka, 1989].

8.3 Importance of selectivity in deductive argumentation

Some of the issues raised with classical exhaustive graphs (i.e. the lack of structural completeness, the failure of consistent extensions property for some choices of attack relation, and the correspondences with maximally consistent subsets of the knowledgebase) suggest that often we need a more sophisticated way of constructing argument graphs. In other words, to reflect any abstract argument graph in a logical argument system based on a richer logic, we need to be selective in the choice of arguments and counterarguments from those that can be generated from the knowledgebase. Furthermore, this is not just for theoretical interest. Practical argumentation often seems to use richer logics such as classical logic, and often the arguments and counterarguments considered are not exhaustive. Therefore, we need to better understand how the arguments are selected. For example, suppose agent 1 posits $A_1 = \langle\{b, b \rightarrow a\}, a\rangle$, and agent 2 then posits $A_2 = \langle\{c, c \rightarrow \neg b\}, \neg b\rangle$. It would be reasonable for this dialogue to stop at this point (since further arguments are only re-expressing the same conflict, and so, in a sense, they would be redundant) even though there are further arguments that can be constructed from the public knowledge such as $A_3 = \langle\{b, c \rightarrow \neg b\}, \neg c\rangle$. So in terms of constructing the constellation of arguments and counterarguments from the knowledge, we need to know what are the underlying principles for selecting arguments.

Selectivity in argumentation is an important and as yet under-developed topic [Besnard and Hunter, 2008]. Two key dimensions are selectivity based on object-level information and selectivity based on meta-level information.

- **Selectivity based on object-level information** In argumentation, object-level information is the information in the premises and claims of the arguments. So if these are generated by deductive reasoning from a knowledgebase, then the object-level information is the information in the knowledgebase. Selectivity based on object-level information is concerned with having a more concise presentation of arguments and counterarguments in an argument graph without changing the outcome of the argumentation. For instance, a more concise presentation can be obtained by removing structurally equivalent arguments or by using focal graphs (as discussed in Section 6.3.1).

- **Selectivity based on meta-level information** In argumentation, meta-level information is the information about the arguments and counterarguments (e.g. certainty and sources of the premises in arguments) and information about the participants or audience of the argumentation (e.g. the goals, beliefs, or biases of the audience). Selectivity based on meta-level information is concerned with generating an argument graph using the meta-level information according to sound principles. By using this extra information, a different argument graph may be obtained than would be obtained without the extra information. For instance, with a preference relation over arguments which is a form of meta-level information, preference-based argumentation offers a principled way of generating an argument graph that has potentially fewer attacks between arguments than obtained with the classical exhaustive argument graph (as discussed in Section 6.3.2).

Various kinds of meta-level information can be considered for argumentation including preferences over arguments, weights on arguments, weights on attacks, a probability distribution over models of the language of the deductive argumentation, etc. The need for meta-level information also calls for better modeling of the audience, of what they believe, of what they regard as important for their own goals, etc, are important features of selectivity (see for example [Hunter, 2004b; Hunter, 2004a]). Consider a journalist writing a magazine article on current affairs. There are many arguments and counterarguments that could be included, but the writer is selective. Selectivity may be based on what the likely reader already believes and what s/he may find interesting. Or, consider a lawyer in court, again there may be many arguments and counterarguments, that could be used, but only some will be used. Selection will in part be based on what could be believed by the jury, and convince them to take the side of that lawyer. Or, consider a politician giving a speech to an audience of potential voters. Here, the politician will select arguments based on what will be of more interest to the audience. For instance, if the audience is composed of older citizens, there may be more arguments concerning healthcare, whereas if the audience is composed of younger citizens, there may be more arguments concerning job opportunities. So whilst selectivity is clearly important in real-world argumentation, we need principled ways of bringing selectivity into structured argumentation such as that based on deductive argumentation.

8.4 Automated reasoning for deductive argumentation

For argumentation, it is computationally challenging to generate arguments from a knowledgebase with the minimality constraints using classical logic. If we consider the problem as an abduction problem, where we seek the existence of a minimal subset of a set of formulae that implies the consequent, then the problem is in the second level of the polynomial hierarchy [Eiter and Gottlob, 1995]. The difficult nature of argumentation has been underlined by studies concerning the complexity of finding individual arguments [Parsons *et al.*, 2003], the complexity of some decision problems concerning the instantiation of argument graphs with classical logic arguments and the direct undercut attack relation [Wooldridge *et al.*, 2006], and the complexity of finding argument trees [Hirsch and Gorogiannis, 2009]. Encodation of these tasks as quantified Boolean formulae also indicate that development of algorithms is a difficult challenge [Besnard *et al.*, 2009], and Post's framework, has been used to give a breakdown of where complexity lies in logic-based argumentation [Creignou *et al.*, 2011].

Despite the computational complexity results, there has been progress in developing algorithms for constructing arguments and counterarguments. One approach has been to adapt the idea of connection graphs to enable us to find arguments. A connection graph [Kowalski., 1975; Kowalski., 1979] is a graph where a clause is represented by a node and an arc (ϕ, ψ) denotes that there is a disjunct in ϕ with its complement being a disjunct in ψ. Essentially this graph is manipulated to obtain a proof by contradiction. Furthermore, finding this set of formulae can substantially reduce the number of formulae that need to be considered for finding proofs for a claim, and therefore for finding arguments and canonical undercuts. Versions for full propositional logic, and for a subset of first-order logic, have been developed and implemented [Efstathiou and Hunter, 2011].

Another approach for algorithms for generating arguments and counterarguments (canonical undercuts) has been given in a proposal that is based on a SAT solver [Besnard *et al.*, 2010]. This approach is based on standard SAT technology and it is also based on finding proofs by contradiction.

8.5 Handling enthymemes

Real arguments (i.e. those presented by people in general) are normally enthymemes. We can consider two types which we will refer to as *implicit support enthymemes* and *implicit claim enthymemes*. An implicit support enthymeme only explicitly represents some of the premises for entailing its claim. An implicit claim enthymeme not only misses some of the premises for entailing its claim, but also does not explicitly represent its claim. So if Γ is the set of premises explicitly given for an implicit support enthymeme, and α is the claim, then Γ does not entail α, but there are some implicitly assumable premises Γ' such that $\Gamma \cup \Gamma'$ is a minimal consistent set of formulae that entails α. For example, for a claim that *you need an umbrella today*, a husband may give his wife the premise *the weather report predicts rain*. Clearly, the premise does not entail the claim, but it is easy for the wife to identify the common knowledge used by the husband (i.e. *if the weather report predicts rain, then you need an umbrella today*) in order to reconstruct the intended argument correctly.

If we want to build agents that can understand real arguments coming from humans, they need to identify the missing premises and missing claims with some reliability. And if we want to build agents that can generate real arguments for humans, they need to identify the premises and claims that can be missed without causing undue confusion. Clearly, deciding how to construct or deconstruct enthymemes is difficult, and proposals for logic-based formalisations of the process remain underdeveloped.

In [Hunter, 2007; Black and Hunter, 2008], we introduced a way for each agent in a dialogue to have information about what it can use as shared knowledge, and then a proponent can use this information to remove redundant premises from an intended argument (creating an implicit support enthymeme), and a recipient can use this information to identify the necessary premises in order to recover the intended argument. Then in [Black and Hunter, 2012], this proposal was extended by allowing each agent to also have a representation of information requirements. These are formulae that the agent would like to receive arguments about. So for example if an agent asks a question, it is making an explicit declaration of an information requirement. By introducing the notion of information requirements, we can formalise a key idea from relevance theory that the relevance of an utterance depends on maximising cognitive effect and minimising cognitive effort. This allows proponents to construct both implicit support and implicit claim enthymemes that are relevant for the intended recipient, and the recipient can deconstruct such enthymemes by using relevance criteria to overcome some of the ambiguities that normally arise when trying to understand enthymemes. This has been further developed for persuasion dialogues [Dupin de Saint-Cyr, 2011].

9 Discussion

Deductive argumentation is an appealing approach to instantiation of abstract argumentation. A deductive argument has all the premises explicitly in the support of the argument, and the claim is derived by the consequence relation of the base logic. Since established and well-understood logics can be used as a base logic, the semantics and proof theory for the individual arguments is inherited from the base logic. This is important if we want to harness developments in knowledge representation and in computational linguistics for specialized logics. So for example, if we want to represent a natural language argument as a deductive argument, we can use an appropriate logic from computational linguistics to represent the information.

The approach is also flexible since different constraints can put on an argument (e.g. consistency, minimality, etc) and on the definition for counterargument (e.g. defeater, undercut, direct undercut, canonical undercut, rebuttal, etc). Furthermore, we can use the approach for descriptive graphs and for generated graphs. Over the past few years, there has been substantial interest in bipolar argumentation (see for example [Cayrol and Lagasquie-Schiex, 2005; Oren and Norman, 2008; Nouioua and Risch, 2011]). In future work, we would like to generalize the use of deductive arguments for bipolar argumentation.

Acknowledgements

The authors are very grateful to Henry Prakken for some very interesting and valuable discussions on the nature of structural argumentation. The authors are also very grategul to the anonymous reviewers for their comprehensive and insightful feedback. This research was partly supported by EPSRC grant EP/N008294/1.

BIBLIOGRAPHY

[Amgoud and Besnard, 2009] L. Amgoud and Ph. Besnard. Bridging the gap between abstract argumentation systems and logic. In *Proceedings of the 3rd International Conference on Scalable Uncertainty Management (SUM'09)*, volume 5785 of *Lecture Notes in Computer Science*, pages 12–27. Springer, 2009.

[Amgoud and Besnard, 2010] L. Amgoud and Ph. Besnard. A formal analysis of logic-based argumentation systems. In *Proceedings of the 4th International Conference on Scalable Uncertainty Management (SUM'10)*, volume 6379 of *Lecture Notes in Computer Science*, pages 42–55. Springer, 2010.

[Amgoud and Cayrol, 2002] L. Amgoud and C. Cayrol. A reasoning model based on the production of acceptable arguments. *Annals of Mathematics and Artificial Intelligence*, 34:197–215, 2002.

[Amgoud and Vesic, 2010] L. Amgoud and S. Vesic. Handling inconsistency with preference-based argumentation. In *Proceedings of the 4th International Conference on Scalable*

Uncertainty Management (SUM'10), volume 6379 of *Lecture Notes in Computer Science*, pages 56–69. Springer, 2010.

[Amgoud *et al.*, 2011] L. Amgoud, Ph. Besnard, and S. Vesic. Identifying the core of logic-based argumentation systems. In *Proceedings of the IEEE International Conference on Tools with Artificial Intelligence, (ICTAI'11)*, pages 633–636. IEEE Press, 2011.

[Arló-Costa and Shapiro, 1992] H. Arló-Costa and S. Shapiro. Maps between conditional logic and non-monotonic logic. In *Proceedings of the 3rd International Conference on Principles of Knowledge Representation and Reasoning (KR'92)*, page 553565. Morgan Kaufmann, 1992.

[Baroni and Giacomin, 2007] P. Baroni and M. Giacomin. On principle-based evaluation of extension-based argumentation semantics. *Artificial Intelligence*, 171:675–700, 2007.

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

[Besnard and Hunter, 2005] Ph Besnard and A Hunter. Practical first-order argumentation. In *Proceedings of the 20th American National Conference on Artificial Intelligence (AAAI'05)*, pages 590–595. MIT Press, 2005.

[Besnard and Hunter, 2008] Ph. Besnard and A Hunter. *Elements of Argumentation.* MIT Press, 2008.

[Besnard *et al.*, 2009] Ph. Besnard, A. Hunter, and S. Woltran. Encoding deductive argumentation in quantified boolean formulae. *Artificial Intelligence*, 173(15):1406–1423, 2009.

[Besnard *et al.*, 2010] Ph. Besnard, E. Gregoire, C. Piette, and B. Raddaoui. Mus-based generation of arguments and counter-arguments. In *Proceedings of the 11th IEEE International Conference on Information Reuse and Integration (IRI'10)*, pages 239–244. IEEE Press, 2010.

[Besnard *et al.*, 2013] Ph. Besnard, E. Gregoire, and B. Raddaoui. A conditional logic-based argumentation framework. In *Proceedings of the 7th International Conference on Scalable Uncertainty Management (SUM'13)*, volume 7958 of *Lecture Notes in Computer Science*, pages 44–56. Springer, 2013.

[Black and Hunter, 2008] E. Black and A. Hunter. Using enthymemes in an inquiry dialogue system. In *Proceedings of the 7th International Conference on Autonomous Agents and Multiagent Systems (AAMAS'08)*, pages 437–444. IFAAMAS, 2008.

[Black and Hunter, 2012] E Black and A Hunter. A relevance-theoretic framework for constructing and deconstructing enthymemes. *Journal of Logic and Computation.*, 22(1):55–78, 2012.

[Black *et al.*, 2009] E. Black, A Hunter, and J Pan. An argument-based approach to using multiple ontologies. In *Proceedings of the 3rd International Conference on Scalable Uncertainty Management (SUM'09)*, volume 5785 of *Lecture Notes in Computer Science*, pages 68–79. Springer, 2009.

[Brewka, 1989] G. Brewka. Preferred subtheories: An extended logical framework for default reasoning. In *Proceedings of the 11th International Joint Conference on Artificial Intelligence (IJCAI'89)*, pages 1043 – 1048. Morgan Kaufmann, 1989.

[Caminada and Amgoud, 2005] M. Caminada and L. Amgoud. An axiomatic account of formal argumentation. In *Proceedings of the 20th National Conference on Artificial Intelligence (AAAI'05)*, pages 608–613. AAAI Press, 2005.

[Cayrol and Lagasquie-Schiex, 2005] C. Cayrol and M-C Lagasquie-Schiex. On the acceptability of arguments in bipolar argumentation frameworks. In *Proceedings of the 8th Symbolic and Quantitative Approaches to Reasoning and Uncertainty (ECSQARU'05)*, volume 3571 of *LNCS*, pages 378–389. Springer, 2005.

[Cayrol, 1995] C. Cayrol. On the relation between argumentation and non-monotonic coherence-based entailment. In *Proceedings of the 14th International Joint Conference on Artificial Intelligence (IJCAI'95)*, pages 1443–1448, 1995.

[Coste-Marquis *et al.*, 2005] S Coste-Marquis, C Devred, and P Marquis. Symmetric argumentation frameworks. In *Proceedings of the 8th European Conference on Symbolic and Quantitative Approaches to Reasoning with Uncertainty (ECSQARU'05)*, volume 3571 of *Lecture Notes in Computer Science*, pages 317–328. Springer, 2005.

[Creignou *et al.*, 2011] N. Creignou, J. Schmidt, M. Thomas, and S. Woltran. Complexity of logic-based argumentation in Post's framework. *Argument & Computation*, 2(2-3):107–129, 2011.

[Cross and Nute, 2001] C. Cross and D. Nute. Conditional logic. In D. Gabbay, editor, *Handbook of Philosophical Logic*, volume IV. D. Reidel, 2001.

[Delgrande, 1987] J. Delgrande. A first-order logic for prototypical properties. *Artificial Intelligence*, 33:105–130, 1987.

[Dung, 1995] P. 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.

[Dunne, 2007] P Dunne. Computational properties of argument systems satisfying graph-theoretic constraints. *Artificial Intelligence*, 171(10-15):701–729, 2007.

[Dupin de Saint-Cyr, 2011] F. Dupin de Saint-Cyr. Handling enthymemes in time-limited persuasion dialogs. In *Proceedings of the 5th International Conference on Scalable Uncertainty Management (SUM'11)*, volume 6929 of *Lecture Notes in Computer Science*, pages 149–162. Springer, 2011.

[Dvorák *et al.*, 2012a] W. Dvorák, R. Pichler, and S. Woltran. Towards fixed-parameter tractable algorithms for abstract argumentation. *Artificial Intelligence*, 186:1–37, 2012.

[Dvorák *et al.*, 2012b] W. Dvorák, S. Szeider, and S. Woltran. Abstract argumentation via monadic second order logic. In *Proceedings of the 6th International Conference on Scalable Uncertainty Management (SUM'12)*, volume 7520 of *LNCS*, pages 85–98. Springer, 2012.

[Efstathiou and Hunter, 2011] V. Efstathiou and A. Hunter. Algorithms for generating arguments and counterarguments in propositional logic. *International Journal of Approximate Reasoning*, 52:672–704., 2011.

[Eiter and Gottlob, 1995] T. Eiter and G. Gottlob. The complexity of logic-based abduction. *Journal of the ACM*, 42(1):3–42, 1995.

[Elvang-Gøransson and Hunter, 1995] M. Elvang-Gøransson and A. Hunter. Argumentative logics: Reasoning with classically inconsistent information. *Data & Knowledge Engineering*, 16(2):125–145, 1995.

[Elvang-Gøransson *et al.*, 1993] M. Elvang-Gøransson, P. Krause, and J. Fox. Acceptability of arguments as 'logical uncertainty'. In *Proceedings of the 2nd European Conference on Symbolic and Quantitative Approaches to Reasoning and Uncertainty (ECSQARU'93)*, volume 747 of *Lecture Notes in Computer Science*, pages 85–90. Springer, 1993.

[Gabbay, 1985] D. Gabbay. Theoretical foundations for nonmonotonic reasoning in expert systems. In K. Apt, editor, *Logic and Models of Concurrent Systems*. Springer, 1985.

[García and Simari, 2004] A. García and G. Simari. Defeasible logic programming: An argumentative approach. *Theory and Practice of Logic Programming*, 4:95–138, 2004.

[Girard, 2006] P. Girard. From onions to broccoli: Generalizing lewis' counterfactual logic. *Journal of Applied Non-Classical Logic*, 17(2):213 – 229, 2006.

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

[Governatori *et al.*, 2004] G. Governatori, M. Maher, G. Antoniou, and D. Billington. Argumentation semantics for defeasible logic. *Journal of Logic and Computation*, 14(5):675–702, 2004.

[Haenni, 1998] R. Haenni. Modelling uncertainty with propositional assumptions-based systems. In *Applications of Uncertainty Formalisms*, volume 1455 of *Lecture Notes in Computer Science*, pages 446–470. Springer, 1998.

[Haenni, 2001] R. Haenni. Cost-bounded argumentation. *International Journal of Approximate Reasoning*, 26(2):101–127, 2001.

[Hirsch and Gorogiannis, 2009] R. Hirsch and N. Gorogiannis. The complexity of the warranted formula problem in propositional argumentation. *Journal of Logic and Computation*, 20(2):481–499, 2009.

[Hunter and Woltran, 2013] A Hunter and S Woltran. Structural properties for deductive argument systems. In *Proceedings of the 12th European Conference on Symbolic and Quantitative Approaches to Reasoning and Uncertainty (ECSQARU'13)*, volume 7958 of *Lecture Notes in Computer Science*, pages 278–289. Springer, 2013.

[Hunter, 2004a] A. Hunter. Making argumentation more believable. In *Proceedings of the 19th National Conference on Artificial Intelligence (AAAI'04)*, pages 269–274. MIT Press, 2004.

[Hunter, 2004b] A. Hunter. Towards higher impact argumentation. In *Proceedings of the 19th National Conference on Artificial Intelligence (AAAI'04)*, pages 275–280. MIT Press, 2004.

[Hunter, 2007] A. Hunter. Real arguments are approximate arguments. In *Proceedings of the 22nd AAAI Conference on Artificial Intelligence (AAAI'07)*, pages 66–71. MIT Press, 2007.

[Hunter, 2008] A Hunter. Reasoning about the appropriateness of proponents for arguments. In *Proceedings of the 23rd AAAI Conference on Artificial Intelligence (AAAI'08)*, pages 89–94. MIT Press, 2008.

[Hunter, 2010] A. Hunter. Base logics in argumentation. In *Proceedings of the 3rd International Conference on Computational Models of Argument (COMMA'10)*, volume 216 of *Frontiers in Artificial Intelligence and Applications*, pages 275–286. IOS Press, 2010.

[Hunter, 2013] A Hunter. A probabilistic approach to modelling uncertain logical arguments. *International Journal of Approximate Reasoning*, 54(1):47–81, 2013.

[Kowalski., 1975] R. Kowalski. A proof procedure using connection graphs. *Journal of the ACM*, 22:572–595, 1975.

[Kowalski., 1979] R. Kowalski. *Logic for Problem Solving.* North-Holland Publishing, 1979.

[Kraus *et al.*, 1990] S. Kraus, D. Lehmann, and M. Magidor. Non-monotonic reasoning, preferential models and cumulative logics. *Artificial Intelligence*, 44:167–207, 1990.

[Liao *et al.*, 2011] B. Liao, L. Jin, and RC. Koons. Dynamics of argumentation systems: A division-based method. *Artificial Intelligence*, 175(11):1790–1814, 2011.

[Makinson, 1994] D. Makinson. General patterns in nonmonotonic reasoning. In D. Gabbay, C. Hogger, and J. Robinson, editors, *Handbook of Logic in Artificial Intelligence and Logic Programming*, volume 3, pages 35–110. Oxford University Press, 1994.

[Mann and Hunter, 2008] N. Mann and A. Hunter. Argumentation using temporal knowledge. In *Proceedings of the 2nd Conference on Computational Models of Argument (COMMA'08)*, volume 172 of *Frontiers in Artificial Intelligence and Applications*, pages 204–215. IOS Press, 2008.

[McCarthy, 1980] J. McCarthy. Circumscription: A form of non-monotonic reasoning. *Artificial Intelligence*, 13(1-2):23–79, 1980.

[Moguillansky *et al.*, 2010] M. Moguillansky, R. Wassermann, and M. Falappa. An argumentation machinery to reason over inconsistent ontologies. In *Advances in Artificial Intelligence (IBERAMIA 2010)*, volume 6433 of *LNCS*, pages 100–109. Springer, 2010.

[Nouioua and Risch, 2011] F. Nouioua and V. Risch. Argumentation frameworks with necessities. In *Proceedings of the 5th International Conference on Scalable Uncertainty Mangement (SUM'11)*, volume Lecture Notes in Computer Science, pages 163–176. Springer, 2011.

[Oren and Norman, 2008] N. Oren and T. Norman. Semantics for evidence-based argumentation. In *Proceedings of the 2nd International Conference Computational Models of Argument (COMMA'08)*, pages 276–284. IOS Press, 2008.

[Parsons *et al.*, 2003] S. Parsons, M. Wooldridge, and L. Amgoud. Properties and complexity of some formal inter-agent dialogues. *Journal of Logic and Computation*, 13(3):347–376, 2003.

[Pollock, 1987] J.L. Pollock. Defeasible reasoning. *Cognitive Science*, 11(4):481–518, 1987.

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

[Wooldridge *et al.*, 2005] M. Wooldridge, P. McBurney, and S. Parsons. On the meta-logic of arguments. In *Argumentation in Multi-agent Systems*, volume 4049 of *Lecture Notes in Computer Science*, pages 42–56. Springer, 2005.

[Wooldridge *et al.*, 2006] M. Wooldridge, P. Dunne, and S. Parsons. On the complexity of linking deductive and abstract argument systems. In *Proceedings of the 21st National Conference on Artificial Intelligence (AAAI'06)*, pages 299–304. AAAI Press, 2006.

[Zhang and Lin, 2013] X. Zhang and Z. Lin. An argumentation framework for description logic ontology reasoning and management. *Journal of Intelligent Information Systems*, 40(3):375–403, 2013.

[Zhang *et al.*, 2010] X. Zhang, Z. Zhang, D. Xu, and Z. Lin. Argumentation-based reasoning with inconsistent knowledge bases. In *Advances in Artificial Intelligence*, volume 6085 of *Lecture Notes in Computer Science*, pages 87–99. Springer, 2010.

Philippe Besnard
IRIT
Université Paul Sabatier
F-31062, Toulouse, France
Email: philippe.besnard@irit.fr

Anthony Hunter
Department of Computer Science
University College London
London, WC1E 6BT, UK
Email: anthony.hunter@ucl.ac.uk