

Compilation of Logical Arguments

Leila Amgoud¹ and Dragan Doder²

¹CNRS - IRIT, France

²Paul Sabatier University - IRIT, France

{Leila.Amgoud, Dragan.Doder}@irit.fr

Abstract

Several argument-based logics have been defined for handling inconsistency in propositional knowledge bases. We show that they may miss intuitive consequences, and discuss two sources of this drawback: the definition of logical argument i) may prevent formulas from being justified, and ii) may allow irrelevant information in argument's support. We circumvent these two issues by considering a general definition of argument and compiling each argument. A compilation amounts to forgetting in an argument's support any irrelevant variable. This operation returns zero, one or several concise arguments, which we then use in an instance of Dung's abstract framework. We show that the resulting logic satisfies existing rationality postulates, namely consistency and closure under deduction. Furthermore, it is more productive than the existing argument-based and coherence-based logics.

1 Introduction

Handling inconsistency in propositional knowledge bases (KBs) has been studied in AI for a long time. Several two-level logics have been defined: They start with classical propositional logic and define on top of it a non-monotonic logic that infers non-trivial consequences from an inconsistent KB. At least two families of such logics can be distinguished: *coherence-based* logics and *argument-based* logics. Initiated in [Rescher and Manor, 1970] and further developed in [Benferhat *et al.*, 1993; Cayrol and Lagasquie-Schiev, 1995; Batens, 2003], the former compute the set of all maximal (for set inclusion) consistent subbases (MCSs) of a KB, then apply one of the following inference mechanisms for drawing consequences from the KB: *free mechanism* infers any formula that follows logically from the intersection of all MCSs; *universal mechanism* infers any formula that follows from each of the MCSs, *argued mechanism* draws any formula that follows from at least one MCS while its negation does not follow from any MCS; *existential mechanism* infers formulas that follow from at least one MCS.

Argument-based logics follow another process. They justify every *candidate* consequence of a KB by *arguments*,

identify possible conflicts between arguments, evaluate arguments using extension semantics from [Dung, 1995], and finally keep among the candidate consequences those supported by arguments belonging to (some or all) extensions. Examples of such logics are those defined in [Cayrol, 1995; Besnard and Hunter, 2001; Gorogiannis and Hunter, 2011; Amgoud and Besnard, 2013; Vesic, 2013; Arieli *et al.*, 2018]. It has been shown that the logics proposed in [Cayrol, 1995; Amgoud and Besnard, 2013; Vesic, 2013] coincide with the coherence-based logic that uses the universal inference mechanism, and three logics defined in [Arieli *et al.*, 2018] coincide with the coherence-based logics that use respectively the free mechanism, argued mechanism and existential one. Thus, argument-based logics suffer from the same limits as the coherence-based ones. Namely, they may miss intuitive consequences. For instance, all the existing argument-based and coherence-based logics will miss the consequence $q \wedge t$ from the KB $\{p \wedge q, \neg p \wedge t\}$ while $q \wedge t$ is not concerned by the inconsistency in the base. Furthermore, the argument-based logics that boil down to the existential or argued inference violate two key properties: *consistency* and *closure under deduction* of the set of consequences [Amgoud, 2014].

In this paper, we propose a novel argument-based logic that circumvents all the above limits. We start by showing that existing argument-based logics may miss intuitive consequences due to the definition of logical argument. The latter i) may prevent formulas from being justified, and ii) may allow irrelevant information in the argument's support, leading thus to unexpected attacks on the argument. Then, we solve these two issues by considering a very general notion of argument and compiling each such argument by *forgetting* in its support irrelevant variables. The result of this operation is zero, one, or several arguments that enjoy desirable properties including conciseness and consistency of their supports. Finally, we use those arguments in an instance of Dung's framework, and show that the new defined logic satisfies existing rationality postulates, namely consistency and closure under deduction. Furthermore, it is more productive than the existing argument-based and coherence-based logics.

The paper is organized as follows: Section 2 recalls some notions, Section 3 discusses limits of existing logics. Section 4 shows the sources of those limits, Section 5 introduces compilation, and Section 6 introduces a new argument-based logic and discusses its properties. Section 7 concludes.

2 Background

Logical preliminaries: Throughout the paper, we consider classical propositional logic (\mathcal{F}, \vdash) , where \mathcal{F} is a propositional language built up from a finite set \mathcal{P} of variables, the two Boolean constants \top (true) and \perp (false), and the usual connectives $(\neg, \vee, \wedge, \rightarrow, \leftrightarrow)$, and \vdash is the consequence relation of the logic. Two formulas $\phi, \psi \in \mathcal{F}$ are *logically equivalent*, denoted by $\phi \equiv \psi$, iff $\phi \vdash \psi$ and $\psi \vdash \phi$.

A finite subset Φ of \mathcal{F} , denoted by $\Phi \subseteq_f \mathcal{F}$, is *consistent* iff $\Phi \not\vdash \perp$, it is *inconsistent* otherwise. $\text{Max}(\Phi)$ denotes the set of all maximal (for set inclusion) consistent subsets of the set Φ , and $\text{CN}(\Phi)$ denotes its *deductive closure*, i.e., $\text{CN}(\Phi) = \{\phi \in \mathcal{F} \mid \Phi \vdash \phi\}$. Two finite sets of formulas $\Phi, \Psi \subseteq_f \mathcal{F}$ are *equivalent*, denoted by $\Phi \cong \Psi$, iff there is a bijection $f : \Phi \rightarrow \Psi$ such that $\forall \phi \in \Phi, \phi \equiv f(\phi)$. For instance, the two sets $\{p, q\}$ and $\{p \wedge q\}$ are not equivalent, while clearly $\{p, q\} \cong \{p \wedge p, q\}$.

Let $\phi \in \mathcal{F}$. The function $\text{Var}(\phi)$ returns all the variables occurring in ϕ . For instance, $\text{Var}(p \wedge \neg q) = \{p, q\}$. For $p \in \mathcal{P}$, $\phi_{p \leftarrow \perp}$ (resp. $\phi_{p \leftarrow \top}$) denotes the formula obtained by substituting in a uniform way every occurrence of p in the formula ϕ by the Boolean constant \perp (resp. \top). For instance, if $\phi = p \wedge q$, then $\phi_{p \leftarrow \perp} = \perp \wedge q \equiv \perp$ and $\phi_{p \leftarrow \top} = \top \wedge q \equiv q$.

According to [Lang and Marquis, 1998], a formula $\phi \in \mathcal{F}$ is *independent* of a set $V \subseteq \mathcal{P}$, denoted by $V \not\sim \phi$, iff there exists $\psi \in \mathcal{F}$ such that $\phi \equiv \psi$ and $V \cap \text{Var}(\psi) = \emptyset$. ϕ is *simplified* iff for every $p \in \text{Var}(\phi)$, it is not the case that $\{p\} \not\sim \phi$, i.e., ϕ does not contain any variable it is independent of. For example, $p \wedge (q \vee \neg q)$ is not simplified since it is equivalent to p , which doesn't contain the variable q .

Variable forgetting: Let us recall the notion of *forgetting* variables in a formula as defined in [Lang et al., 2003]. The idea is to weaken a formula by ignoring some of its variables. Intuitively, for a formula $\phi \in \mathcal{F}$ and a set of variables $V \subseteq \mathcal{P}$, the *forgetting operator* returns the weakest logical consequence of ϕ that does not contain variables from V . It is inductively defined up to logical equivalence. In order to avoid working “up to equivalence” and to specify the co-domain of the forgetting operator, from now on we use a fixed set of formulas \mathcal{L} , which contains one formula of \mathcal{F} per equivalence class (i.e., for every $\phi \in \mathcal{F}$, there exists a *unique* $\psi \in \mathcal{L}$ such that $\phi \equiv \psi$). Moreover, in order to simplify the presentation, we assume that formulas from \mathcal{L} are simplified.

While we do not specify the elements of \mathcal{L} , we use concrete formulas in the examples, and they are assumed to belong to \mathcal{L} when it is clear from context. In the following definition, we denote by $2^{\mathcal{P}}$ the set of all subsets of \mathcal{P} .

Definition 1 (Variable Forgetting) The forgetting operator $\text{Forget} : \mathcal{F} \times 2^{\mathcal{P}} \rightarrow \mathcal{L}$ is defined inductively as follows:

- $\text{Forget}(\phi, \{\}) \equiv \phi$;
- $\text{Forget}(\phi, \{p\}) \equiv \phi_{p \leftarrow \perp} \vee \phi_{p \leftarrow \top}$;
- $\text{Forget}(\phi, V \cup \{p\}) \equiv \text{Forget}(\text{Forget}(\phi, V), \{p\})$.

Assume for instance that $\mathcal{P} \subseteq \mathcal{L}$ and let $\phi = p \wedge q$. $\text{Forget}(\phi, \{p\}) \equiv (\top \wedge q) \vee (\perp \wedge q)$, so $\text{Forget}(\phi, \{p\}) = q$. Note that for $\psi = \{p \vee q\}$, $\text{Forget}(\psi, \{p\}) = \top$.

Property 1 [Lang et al., 2003] Let $\phi, \psi \in \mathcal{F}$, $V_1, V_2 \subseteq \mathcal{P}$.

1. $\phi \vdash \text{Forget}(\phi, V_1)$
2. If $\phi \vdash \psi$, then $\text{Forget}(\phi, V_1) \vdash \text{Forget}(\psi, V_1)$
3. If $V_1 \subseteq V_2$, then $\text{Forget}(\phi, V_1) \vdash \text{Forget}(\phi, V_2)$
4. $V_1 \not\sim \phi$ iff $\phi \equiv \text{Forget}(\phi, V_1)$
5. $\text{Forget}(\text{Forget}(\phi, V_1), V_2) = \text{Forget}(\phi, V_1 \cup V_2)$

3 Limits of Existing Logics

One of the prominent approaches for handling inconsistency in knowledge bases consists of restoring consistency. It computes all the MCSs of a base, then applies an inference mechanism on them. We recall below the four main inference relations defined initially in [Rescher and Manor, 1970; Benferhat et al., 1993].

Definition 2 Let $\Sigma \subseteq_f \mathcal{F}$ and $\phi \in \mathcal{F}$.

- *Free Inference:* $\Sigma \vdash^f \phi$ iff $\bigcap_{\mathcal{S}_i \in \text{Max}(\Sigma)} \mathcal{S}_i \vdash \phi$.
- *Universal Inference:* $\Sigma \vdash^u \phi$ iff $\forall \mathcal{S} \in \text{Max}(\Sigma), \mathcal{S} \vdash \phi$.
- *Argumentative Inference:* $\Sigma \vdash^a \phi$ iff $\exists \mathcal{S} \in \text{Max}(\Sigma)$ s.t. $\mathcal{S} \vdash \phi$ and $\nexists \mathcal{S}' \in \text{Max}(\Sigma)$ s.t. $\mathcal{S}' \vdash \neg \phi$.
- *Existential Inference:* $\Sigma \vdash^e \phi$ iff $\exists \mathcal{S} \in \text{Max}(\Sigma)$ s.t. $\mathcal{S} \vdash \phi$.

The following properties have already been shown.

Property 2 [Benferhat et al., 1993] Let $\Sigma \subseteq_f \mathcal{F}$ and $\phi \in \mathcal{F}$.

- $\Sigma \vdash^f \phi \Rightarrow \Sigma \vdash^u \phi \Rightarrow \Sigma \vdash^a \phi \Rightarrow \Sigma \vdash^e \phi$.
- The set $\{\phi \in \mathcal{F} \mid \Sigma \vdash^f \phi\}$ is consistent.
- The set $\{\phi \in \mathcal{F} \mid \Sigma \vdash^u \phi\}$ is consistent.
- The set $\{\phi \in \mathcal{F} \mid \Sigma \vdash^a \phi\}$ may be inconsistent.
- The set $\{\phi \in \mathcal{F} \mid \Sigma \vdash^e \phi\}$ may be inconsistent.
- $\{\phi \in \mathcal{F} \mid \Sigma \vdash^a \phi\} \neq \text{CN}(\{\phi \in \mathcal{F} \mid \Sigma \vdash^a \phi\})$.

The first property shows that \vdash^f and \vdash^e are respectively the most cautious and most productive inference relations. The last three properties show that the two inference relations \vdash^a and \vdash^e violate the two rationality postulates proposed in [Amgoud, 2014], which claim that the set of consequences drawn from a knowledge base should be both consistent and closed under deduction. These violations are seen as important drawbacks that impede the relevance of the two logics in concrete applications.

Several argument-based logics have been proposed in the literature for handling inconsistency. They are based on the justification of each candidate consequence by arguments, and the evaluation of the latter by extension semantics, namely stable or preferred [Dung, 1995]. Despite differences in spirit and of processes, it has been shown that those logics coincide with one of the four above-mentioned coherence-based logics. Indeed, the logics from [Cayrol, 1995; Amgoud and Besnard, 2013; Vesic, 2013] correspond exactly to \vdash^u while three logics from [Arieli et al., 2018] ultimately boil down respectively to \vdash^f , \vdash^a and \vdash^e . Consequently, argument-based and coherence-based logics suffer from the same drawbacks shown in Property 2, and may miss intuitive consequences as shown in the following example.

Example 1 Consider the following three knowledge bases.

$$\begin{array}{ll} \Sigma_1 = \{p \wedge q, \neg p \wedge q\} & \text{Max}(\Sigma_1) = \{\{p \wedge q\}, \{\neg p \wedge q\}\} \\ \Sigma_2 = \{p \wedge q, \neg p\} & \text{Max}(\Sigma_2) = \{\{p \wedge q\}, \{\neg p\}\} \\ \Sigma_3 = \{p \wedge q, \neg p \wedge t\} & \text{Max}(\Sigma_3) = \{\{p \wedge q\}, \{\neg p \wedge t\}\} \end{array}$$

It is easy to see that $\Sigma_1 \not\vdash^f q$, $\Sigma_1 \vdash^u q$, $\Sigma_2 \not\vdash^u q$, $\Sigma_2 \vdash^a q$, $\Sigma_3 \not\vdash^f q \wedge t$, $\Sigma_3 \not\vdash^u q \wedge t$, $\Sigma_3 \not\vdash^a q \wedge t$, and $\Sigma_3 \not\vdash^e q \wedge t$.

4 Limits of Logical Argument

This section shows that the problem of missing consequences by existing argument-based logics is due to the definition of argument. An argument is a set of premises together with a formula called respectively support and conclusion. The support provides a reasonable justification for inferring the conclusion from a knowledge base. Consequently, in the literature (e.g., [Cayrol, 1995; Besnard and Hunter, 2001; Gorogiannis and Hunter, 2011; Amgoud and Besnard, 2013; Vesic, 2013]) it is assumed to be consistent and minimal (for set inclusion). Consistency ensures *reasonableness* of the justification while minimality guarantees *relevance* of the premises to the conclusion.

Definition 3 (Restricted Argument) A restricted argument built from $\Sigma \subseteq_f \mathcal{F}$ is a pair $\langle \Phi, \phi \rangle$, $\Phi \subseteq \Sigma$, $\phi \in \mathcal{F}$, s.t.

- Φ is consistent (Consistency)
- $\Phi \vdash \phi$ (Validity)
- $\nexists \Phi' \subset \Phi$ such that $\Phi' \vdash \phi$ (Relevance)

Φ and ϕ are called respectively support and conclusion.

Example 1 (Cont.) Restricted arguments from Σ_1 include: $\langle \{p \wedge q\}, p \rangle$, $\langle \{p \wedge q\}, q \rangle$, $\langle \{p \wedge q\}, p \wedge q \rangle$, $\langle \{p \wedge q\}, p \vee q \rangle$. Note that $\langle \{p \wedge q, \neg p \wedge q\}, t \rangle$ and $\langle \{p \wedge q, \neg p \wedge q\}, q \rangle$ are not restricted arguments. Indeed, their supports are respectively inconsistent and not minimal (for set inclusion).

Despite its large use in the literature, the above definition of restricted arguments suffers from two problems:

- Some formulas may not be justified by arguments.
- An argument's support may not be concise as it may contain variables that are irrelevant for the conclusion.

The problem of unjustified consequences is due to consistency of supports. While consistency discards unreasonable pairs like $\langle \{p, \neg p\}, q \rangle$, it forbids some formulas from being justified by restricted arguments, i.e., it is not possible to build from a KB restricted arguments in their favor. Consequently, those conclusions, even if they are intuitive, cannot be inferred from the base. This is the case for the formula $q \wedge t$ in Σ_3 . Indeed, $q \wedge t$ follows from Σ_3 while the pair $\langle \Sigma_3, q \wedge t \rangle$ is not a restricted argument since the support is inconsistent.

P1: Consistency of supports may prevent some formulas from having restricted arguments. Thus, they can never be inferred from a knowledge base.

While the minimality condition ensures relevance of support's premises to an argument's conclusion, it does not guarantee conciseness of the support. Indeed, a support may contain information that is useless for inferring the conclusion. This is the case for the argument $A = \langle \{p \wedge q\}, q \rangle$ built from

Σ_2 . Note that p is not crucial for getting the conclusion q . A side effect of non-conciseness is that an argument can be attacked on its irrelevant information, leading thus to its rejection and to blocking its conclusion from being inferred from the knowledge base. For instance, in Σ_2 the argument A is attacked by $B = \langle \{\neg p\}, \neg p \vee \neg q \rangle$. This attack does not show that q is false; it is rather about p . Nevertheless, it leads to rejecting A and thus to not inferring q from Σ_2 . To sum up:

P2: Non-conciseness of argument's support may lead to unexpected attacks, that may prevent the argument's conclusion from being inferred.

In the next section, we propose a solution for the two problems (**P1** and **P2**) using a notion of compilation of arguments.

5 Compilation of Unrestricted Arguments

In order to solve the problem of unjustified formulas (**P1**), we propose to rule out the consistency condition as follows.

Definition 4 (Unrestricted Argument) An unrestricted argument built from $\Sigma \subseteq_f \mathcal{F}$ is a pair $\langle \Phi, \phi \rangle$, with $\Phi \subseteq \Sigma$ and $\phi \in \mathcal{F}$, such that $\Phi \vdash \phi$.

Every restricted argument built from Σ is also an unrestricted argument from Σ . The converse does not hold.

Example 1 (Cont.) The pair $\langle \Sigma_3, q \wedge t \rangle$ is an unrestricted argument built from Σ_3 .

While the notion of unrestricted argument solves (**P1**), it presents two disadvantages: First, some unrestricted arguments are not reasonable. Indeed, their supports may be completely unrelated to their conclusions, like $\langle \{p, \neg p\}, q \rangle$, or not minimal like $\langle \{p, q\}, q \rangle$. Second, every formula in a KB can be contradicted, even the *free* formulas, i.e., those that do not belong to any minimal (for set inclusion) inconsistent sub-base. This increases greatly the size of argumentation graphs. We introduce next a notion of compilation of unrestricted arguments which will solve (**P2**) and these problems. Indeed, compilation addresses the following five objectives:

- It removes all unreasonable arguments, like $\langle \{p, \neg p\}, q \rangle$.
- It restores minimality by removing all unnecessary formulas from the support, eg. removing p from $\langle \{p, q\}, q \rangle$.
- It restores consistency of supports.
- It decomposes complex arguments, like $\langle \{p \wedge q\}, p \vee q \rangle$, into elementary ones, namely $\langle \{p\}, p \vee q \rangle$ and $\langle \{q\}, p \vee q \rangle$. This allows more focused attacks. For instance, if the KB contains $\neg p$, then only $\langle \{p\}, p \vee q \rangle$ can be attacked.
- It makes arguments concise by removing all the variables which are useless for getting the argument's conclusion, eg. removing p from $\langle \{p \wedge q\}, q \rangle$.

All these objectives are addressed using the *forgetting operator* from [Lang et al., 2003]. Before giving the formal definitions, let us first give some useful notations.

Notations: Let $\Sigma \subseteq_f \mathcal{F}$. We denote by $\text{Arg}(\Sigma)$ the set of all unrestricted arguments built from Σ . For any $A = \langle \Phi, \phi \rangle \in \text{Arg}(\mathcal{F})$, the functions Supp and Conc return respectively the support Φ and the conclusion ϕ of A . For any $\mathcal{E} \subseteq \text{Arg}(\mathcal{F})$, $\text{Base}(\mathcal{E}) = \cup \text{Supp}(A), A \in \mathcal{E}$.

Definition 5 (Compilation) Let $\Sigma \subseteq_f \mathcal{F}$ and $A = \langle \{\phi_1, \dots, \phi_n\}, \phi \rangle \in \text{Arg}(\Sigma)$. A pre-compilation of A is any ordered pair of the form

$\langle \{\text{Forget}(\phi_1, V_1), \dots, \text{Forget}(\phi_n, V_n)\}, \phi \rangle$, where

- $V_i \subseteq \text{Var}(\phi_i)$, for every $i \in \{1, \dots, n\}$
- the set $\{\text{Forget}(\phi_1, V_1), \dots, \text{Forget}(\phi_n, V_n)\}$ is consistent
- $\{\text{Forget}(\phi_1, V_1), \dots, \text{Forget}(\phi_n, V_n)\} \vdash \phi$
- for every $i \in \{1, \dots, n\}$ and every $p \in \text{Var}(\phi_i) \setminus V_i$, $\{\text{Forget}(\phi_1, V_1), \dots, \text{Forget}(\phi_i, V_i \cup \{p\}), \dots, \text{Forget}(\phi_n, V_n)\} \not\vdash \phi$.

The set of all compilations of A is the set $\text{Comp}(A) = \{B^* \mid B \text{ is a pre-compilation of } A\}$, where $\langle \Phi, \phi \rangle^*$ is the pair $\langle \{\psi \in \Phi \mid \psi \neq \top\}, \phi \rangle$.

Thus, compilations are obtained by removing tautologies from pre-compilations. The tautologies may appear in the process of pre-compilation of unrestricted arguments, due to the lack of *minimality* and *consistency* of argument's support. However, we will show in Proposition 2 that the resulting compilations are always restricted arguments.

Notation: We denote by ArgC a function that returns for any $\Sigma \subseteq_f \mathcal{F}$, the set of compilations of arguments in $\text{Arg}(\Sigma)$,

$$\text{ArgC}(\Sigma) = \bigcup_{A \in \text{Arg}(\Sigma)} \text{Comp}(A).$$

Example 1 (Cont.) Consider the unrestricted arguments A, B, C, D built from Σ_2 and E built from Σ_3 .

- $A = \langle \{p \wedge q\}, q \rangle$ (non-concise argument)
- $B = \langle \{p \wedge q, \neg p\}, q \rangle$ (non-minimal argument)
- $C = \langle \{p \wedge q\}, p \vee q \rangle$ (complex argument)
- $D = \langle \{p \wedge q, \neg p\}, t \rangle$ (non-reasonable argument)
- $E = \langle \{p \wedge q, \neg p \wedge t\}, q \wedge t \rangle$ (inconsistent argument)

$\text{Comp}(A) = \{\langle \{q\}, q \rangle\}$ since $\text{Forget}(p \wedge q, \{p\}) = q \equiv (\top \wedge q) \vee (\perp \wedge q)$. Note that the compilation of A is concise (it does no longer contain the useless formula p in its support).

$\text{Comp}(B) = \{\langle \{q, \top\}, q \rangle^*\} = \{\langle \{q\}, q \rangle\}$ since $\text{Forget}(\neg p, \{\neg p\}) = \top \equiv \perp \vee \top$. Note that the compilation of B has a minimal (for set \subseteq) support ($\neg p$ is removed). Furthermore, it is concise (p is removed from $p \wedge q$).

$\text{Comp}(C) = \{\langle \{p\}, p \vee q \rangle, \langle \{q\}, p \vee q \rangle\}$ contains two ways of getting $p \vee q$ from $p \wedge q$. This shows that compilation decomposes complex supports into elementary ones.

$\text{Comp}(D) = \emptyset$ meaning that this non-reasonable argument is discarded. Finally, $\text{Comp}(E) = \{\langle \{q, t\}, q \wedge t \rangle\}$, i.e, the compilation of E keeps from E the necessary parts for getting $q \wedge t$ and removes the others.

The following result investigates the number of compilations of an unrestricted argument.

Proposition 1 For every $A = \langle \{\phi_1, \dots, \phi_n\}, \phi \rangle \in \text{Arg}(\Sigma)$,

- The set $\text{Comp}(A)$ is finite.
- If $\text{Supp}(A)$ is consistent, then $\text{Comp}(A) \neq \emptyset$.

In Definition 5 we compile unrestricted arguments whose supports are subsets of a knowledge base Σ under study. Supports of their compilations, however, do not necessarily use formulas from Σ . We identify the set of formulas they use.

Definition 6 (Σ^*) For any $\Sigma \subseteq_f \mathcal{F}$, we denote by Σ^* the set of all formulas $\phi \in \mathcal{L}$ such that

- $\phi \neq \top$, and
- $\exists \psi \in \Sigma$ such that i) ψ is consistent, and ii) $\exists V \subseteq \text{Var}(\psi)$ such that $\phi = \text{Forget}(\psi, V)$

Example 1 (Cont.) $\Sigma_2^* = \{p \wedge q, p, q, \neg p\}$ and $\Sigma_3^* = \{p \wedge q, p, q, \neg p \wedge t, \neg p, t\}$.

Now we show that compilations of arguments from $\text{Arg}(\Sigma)$ are restricted arguments built from Σ^* . This means that their supports are consistent, minimal and valid.

Proposition 2 Let $\Sigma \subseteq_f \mathcal{F}$, $A \in \text{Arg}(\Sigma)$. For every $\langle \Phi, \phi \rangle \in \text{Comp}(A)$, the following hold:

- $\langle \Phi, \phi \rangle \in \text{Arg}(\Sigma^*)$.
- $\langle \Phi, \phi \rangle$ is a restricted argument.

The next result emphasizes the *idempotency* property of compilation: if an argument is a compilation of another argument, then it is the only compilation of itself.

Proposition 3 Let $\Sigma \subseteq_f \mathcal{F}$. For every $A \in \text{Arg}(\Sigma)$ and every $B \in \text{Comp}(A)$, $\text{Comp}(B) = \{B\}$.

Now we recall a definition from [Amgoud *et al.*, 2014]. It states that two arguments are *equivalent* if their supports are equivalent sets, and their conclusions are equivalent formulas.

Definition 7 (Equivalent Arguments) Let $\Sigma \subseteq_f \mathcal{F}$ and $A, B \in \text{Arg}(\Sigma)$. A and B are equivalent, denoted $A \approx B$, iff

$$(\text{Supp}(A) \cong \text{Supp}(B)) \text{ and } (\text{Conc}(A) \equiv \text{Conc}(B)).$$

For instance, $\langle \{p \wedge q\}, p \wedge q \rangle \approx \langle \{\neg(p \rightarrow \neg q)\}, p \wedge q \rangle$. However, none of them is equivalent to $\langle \{p, q\}, p \wedge q \rangle$.

Equivalent arguments have equivalent compilations.

Proposition 4 Let $\Sigma \subseteq_f \mathcal{F}$ and $A, A' \in \text{Arg}(\Sigma)$ such that $A \approx A'$. There exists a bijection $f : \text{Comp}(A) \rightarrow \text{Comp}(A')$ such that $B \approx f(B)$ for every $B \in \text{Comp}(A)$.

We have already mentioned that compiling an argument eliminates the variables that are useless for the argument. There are two kinds of such variables:

- variables that formulas are independent of. For instance, in $\langle \{p \wedge (t \vee \neg t), p \rightarrow q\}, q \rangle$, the formula $p \wedge (t \vee \neg t)$ is independent of t , since $p \wedge (t \vee \neg t) \equiv p$.
- variables on which formulas depend but they are irrelevant for proving the conclusion. For instance, in $\langle \{p \wedge q\}, q \rangle$, it is clear that the formula $p \wedge q$ depends on p , however p is useless for inferring the conclusion q .

We start with the first type of variables.

Proposition 5 For every $A = \langle \{\phi_1, \dots, \phi_n\}, \phi \rangle \in \text{Arg}(\Sigma)$, if $B = \langle \{\text{Forget}(\phi_1, V_1), \dots, \text{Forget}(\phi_n, V_n)\}, \phi \rangle$ is a pre-compilation of A , then, for every $i \in \{1, \dots, n\}$ and $p \in \mathcal{P}$, if ϕ_i is independent of p , then $p \notin \text{Var}(\text{Forget}(\phi_i, V_i))$.

In the following definition, we focus on variables which are not essential for proving the conclusion.

Definition 8 (Weakly Irrelevant Variables) Let $\Sigma \subseteq_f \mathcal{F}$ and $V \subseteq \mathcal{P}$. The set V is weakly irrelevant for an argument $A = \langle \{\phi_1, \dots, \phi_n\}, \phi \rangle \in \text{Arg}(\Sigma)$ iff

$$\{\text{Forget}(\phi_1, V), \dots, \text{Forget}(\phi_n, V)\} \vdash \phi.$$

By the definition above and the third item of Property 1 it is obvious that if V is weakly irrelevant for A , then every $V' \subseteq V$ is weakly irrelevant for A as well.

Example 2 Let $A = \langle \{p \wedge q \wedge r, (p \vee q) \rightarrow s\}, s \rangle$. Then both $V_1 = \{p, r\}$ and $V_2 = \{q, r\}$ are weakly irrelevant for A . Indeed, for V_1 note that $\text{Forget}(p \wedge q \wedge r, \{p, r\}) = q$, $\text{Forget}((p \vee q) \rightarrow s, \{p, r\}) = q \rightarrow s$ and $\{q, q \rightarrow s\} \vdash s$, and for V_2 that $\{p, p \rightarrow s\} \vdash s$. However, the set $V = V_1 \cup V_2 = \{p, q, r\}$ is not weakly irrelevant for A , since $\text{Forget}(p \wedge q \wedge r, V) = \text{Forget}((p \vee q) \rightarrow s, V) = \top$.

This explains the term “weakly” in Definition 8 – in the example above we can eliminate p from $\text{Supp}(A)$ only under the condition that we do not eliminate q as well, since one of those two variables is necessary for proving s .

A weakly irrelevant set of variables can be eliminated from the support of at least one compilation of an argument.

Proposition 6 Let $\Sigma \subseteq_f \mathcal{F}$ and $V \subseteq \mathcal{P}$. If V is weakly irrelevant for $A \in \text{Arg}(\Sigma)$, then there exists $B \in \text{Comp}(A)$ such that $(\bigcup_{\phi \in \text{Supp}(B)} \text{Var}(\phi)) \cap V = \emptyset$.

Now we consider a stronger variant of irrelevance. The following definition is about the variables which can be always eliminated from the support of the considered argument.

Definition 9 (Strongly Irrelevant Variables) Let $\Sigma \subseteq_f \mathcal{F}$. A variable $p \in \mathcal{P}$ is strongly irrelevant for an argument $A = \langle \{\phi_1, \dots, \phi_n\}, \phi \rangle \in \text{Arg}(\Sigma)$ iff for all $V_1, \dots, V_n \subseteq \mathcal{P}$, if $\{\text{Forget}(\phi_1, V_1), \dots, \text{Forget}(\phi_n, V_n)\} \vdash \phi$, then $\{\text{Forget}(\phi_1, V_1 \cup \{p\}), \dots, \text{Forget}(\phi_n, V_n \cup \{p\})\} \vdash \phi$. The set of all variables which are strongly irrelevant for A is denoted by $\text{Irr}(A)$.

Example 2 (Cont.) Clearly $r \in \text{Irr}(A)$ while $p, q, s \notin \text{Irr}(A)$.

The next result states that for every $B \in \text{Comp}(A)$, all the variables appearing in $\text{Supp}(B)$ belong to the set $\mathcal{P} \setminus \text{Irr}(A)$. Moreover, each variable which is not strongly irrelevant for A appears in the support of at least one compilation of A .

Proposition 7 Let $\Sigma \subseteq_f \mathcal{F}$. For every $A \in \text{Arg}(\Sigma)$,

$$\bigcup_{\phi \in \text{Base}(\text{Comp}(A))} \text{Var}(\phi) = \mathcal{P} \setminus \text{Irr}(A).$$

Example 2 (Cont.) We have that $\text{Comp}(A) = \{\langle \{q, q \rightarrow s\}, s \rangle, \langle \{p, p \rightarrow s\}, s \rangle\}$, so $\text{Base}(\text{Comp}(A)) = \{p, q, q \rightarrow s, p \rightarrow s\}$. Therefore, $\bigcup_{\phi \in \text{Base}(\text{Comp}(A))} \text{Var}(\phi) = \{p, q, s\}$. According to Proposition 7, $\text{Irr}(A) = \mathcal{P} \setminus \{p, q, s\}$.

Remark: It is important to point out that the process of compiling goes beyond forgetting irrelevant variables. For example, it is easy to check that the argument $A = \langle \{p \rightarrow$

$q\} \wedge (s \rightarrow q), (p \rightarrow q) \rightarrow s\}, s \rangle$ does not contain even a weakly irrelevant variable. However, $\text{Comp}(A) = \{B\}$, where $B = \langle \{p \rightarrow q, (p \rightarrow q) \rightarrow s\}, s \rangle$ is obtained by forgetting s in the first formula of $\text{Supp}(A)$. This compilation avoids possible attacks on the formula $s \rightarrow q$, which is useless for inferring s in A .

6 Novel Argument-based Logic

Section 3 has shown that there is a need for a novel logic that is more productive than \vdash^u but more cautious than \vdash^a , since the latter violates consistency. We propose such a logic and show that it satisfies the desirable properties.

The new logic is built on top of classical propositional logic (\mathcal{F}, \vdash) . From now on, we consider a fixed but arbitrary knowledge base Σ . The latter is a finite subset of \mathcal{F} . Moreover, without loss of generality and for the sake of simplicity, Σ is assumed to be free of tautologies, i.e., $\forall \phi \in \Sigma, \phi \notin \text{CN}(\emptyset)$. We define an argumentation system over Σ that instantiates the general framework from [Dung, 1995]. Its set of arguments is $\text{ArgC}(\Sigma)$, hence it contains all compilations of unrestricted arguments built from Σ (in symbols, $\text{ArgC}(\Sigma) = \bigcup_{A \in \text{Arg}(\Sigma)} \text{Comp}(A)$). Its attack relation is the so-called *direct defeat* from [Gorogiannis and Hunter, 2011].

Definition 10 (Direct-Defeat) Let $\langle \Phi, \phi \rangle$ and $\langle \Psi, \psi \rangle$ be two restricted arguments. $\langle \Phi, \phi \rangle$ direct-defeats $\langle \Psi, \psi \rangle$ iff $\exists \delta \in \Psi$ such that $\phi \vdash \neg \delta$.

An instance of Dung’s abstract argumentation framework is then defined and called argumentation system below.

Definition 11 (AS) An argumentation system (AS) over a KB $\Sigma \subseteq_f \mathcal{F}$ is the pair $\mathbf{A} = \langle \text{ArgC}(\Sigma), \mathcal{R} \rangle$ where $\mathcal{R} \subseteq \text{ArgC}(\Sigma) \times \text{ArgC}(\Sigma)$ is such that $\forall A, B \in \text{ArgC}(\Sigma), (A, B) \in \mathcal{R}$ iff A direct-defeats B .

Restricted arguments of an AS are evaluated using *stable semantics*, which was proposed in [Dung, 1995].

Definition 12 (Stable Extensions) Let $\mathbf{A} = \langle \text{ArgC}(\Sigma), \mathcal{R} \rangle$ be an AS. $\mathcal{E} \subseteq \text{ArgC}(\Sigma)$ is a stable extension of \mathbf{A} iff $\nexists A, B \in \mathcal{E}$ such that $(A, B) \in \mathcal{R}$, and $\forall A \in \text{ArgC}(\Sigma) \setminus \mathcal{E}, \exists B \in \mathcal{E}$ such that $(B, A) \in \mathcal{R}$. $\text{Ext}(\mathbf{A})$ denotes the set of all stable extensions of \mathbf{A} .

The non-monotonic consequence relation of the argument-based logic infers formulas that are supported by at least one argument in every stable extension of the AS.

Definition 13 (Consequence Relation \vdash) Let $\mathbf{A} = \langle \text{ArgC}(\Sigma), \mathcal{R} \rangle$ be an AS over $\Sigma \subseteq_f \mathcal{F}$. A formula $\phi \in \mathcal{F}$ is a consequence of Σ , denoted $\Sigma \vdash \phi$, iff $\forall \mathcal{E} \in \text{Ext}(\mathbf{A}), \exists \langle \Phi, \phi \rangle \in \mathcal{E}$.

It is obvious that any formula inferred from Σ using \vdash is also classically inferred from Σ .

Property 3 The inclusion $\{\phi \in \mathcal{F} \mid \Sigma \vdash \phi\} \subseteq \text{CN}(\Sigma)$ holds.

Example 1 (Cont.) Note that $\Sigma_1 \vdash q$, $\Sigma_2 \vdash q$ and $\Sigma_3 \vdash q, t, q \wedge t$. This shows that the new logic treats properly the critical cases where the existing argument-based and coherence-based logics fail.

6.1 Properties of the Logic

We provide next full characterizations of stable extensions of an argumentation system $\mathbf{A} = \langle \text{ArgC}(\Sigma), \mathcal{R} \rangle$ as well as of the set of consequences produced by the inference relation \sim .

We show that there is a one-to-one correspondence between the stable extensions of \mathbf{A} and the MCSs of Σ^* . Furthermore, the set of conclusions of arguments of an extension is consistent, and a stable extension is closed under ArgC . This means that the novel instance of Dung's framework satisfies the consistency and closure under sub-arguments rationality postulates from [Amgoud, 2014].

Theorem 1 *Let $\mathbf{A} = \langle \text{ArgC}(\Sigma), \mathcal{R} \rangle$ be an AS over Σ . For every $\mathcal{E} \in \text{Ext}(\mathbf{A})$, the following properties hold.*

- $\text{Base}(\mathcal{E}) \in \text{Max}(\Sigma^*)$
- The set $\{\phi \mid \langle \Phi, \phi \rangle \in \mathcal{E}\}$ is consistent
- $\mathcal{E} = \text{ArgC}(\text{Base}(\mathcal{E}))$

The next result shows that every MCS of Σ^* has a corresponding stable extension.

Theorem 2 *Let $\mathbf{A} = \langle \text{ArgC}(\Sigma), \mathcal{R} \rangle$ be an AS over Σ . For every $\Phi \in \text{Max}(\Sigma^*)$, the following properties hold:*

- $\Phi = \text{Base}(\text{ArgC}(\Phi))$
- $\text{ArgC}(\Phi) \in \text{Ext}(\mathbf{A})$

The next result shows that there is a one-to-one correspondence between stable extensions of \mathbf{A} and MCSs of Σ^* .

Theorem 3 *Let $\mathbf{A} = \langle \text{ArgC}(\Sigma), \mathcal{R} \rangle$ be an AS over Σ . ArgC is a bijective function from $\text{Max}(\Sigma^*)$ to $\text{Ext}(\mathbf{A})$ (i.e., $\text{ArgC} : \text{Max}(\Sigma^*) \rightarrow \text{Ext}(\mathbf{A})$).*

From the above result, it follows that any argumentation system $\mathbf{A} = \langle \text{ArgC}(\Sigma), \mathcal{R} \rangle$ always has stable extensions.

Corollary 1 *Let $\mathbf{A} = \langle \text{ArgC}(\Sigma), \mathcal{R} \rangle$ built over $\Sigma \subseteq_f \mathcal{T}$. If for any $\phi \in \Sigma$, $\phi \vdash \perp$, then $\text{ArgC}(\Sigma) = \emptyset$. Otherwise, $\text{Ext}(\mathbf{A}) \neq \emptyset$.*

The following result shows that the set of consequences produced by \sim is consistent and closed under CN.

Theorem 4 *Let $\mathbf{A} = \langle \text{ArgC}(\Sigma), \mathcal{R} \rangle$ be an AS over Σ .*

- $\{\phi \in \mathcal{F} \mid \Sigma \sim \phi\}$ is consistent.
- $\{\phi \in \mathcal{F} \mid \Sigma \sim \phi\} = \text{CN}(\{\phi \in \mathcal{F} \mid \Sigma \sim \phi\})$

6.2 Comparison with Related Logics

We show next that our novel logic is more productive (i.e. infers more formulas) than the universal inference relation and its corresponding argument-based logics (see Section 3).

Proposition 8 *Let $\Sigma \subseteq_f \mathcal{F}$, $\phi \in \mathcal{F}$. The implication $\Sigma \sim^u \phi \Rightarrow \Sigma \sim \phi$ holds.*

The converse does not hold as shown below.

Example 1 (Cont.) We have seen that $\Sigma_2 \not\sim^u q$ while $\Sigma_2 \sim q$, and $\Sigma_3 \not\sim^u q, t, q \wedge t$ while $\Sigma_3 \sim q, t, q \wedge t$.

The two consequence relations \sim^e and \sim may lead to different consequences. Remember that unlike \sim , the relation \sim^e may infer an inconsistent set of consequences.

Example 1 (Cont.) It is easy to check that $\Sigma_3 \sim^e p, \neg p$ while $\Sigma_3 \not\sim^e p, \neg p$. Furthermore, $\Sigma_3 \sim q \wedge t$ while $\Sigma_3 \not\sim^e q \wedge t$.

The two consequence relations \sim^a and \sim may also lead to different consequences. The relation \sim^a was proposed in [Benferhat *et al.*, 1993] for solving the limit of universal logic. Indeed, we have seen in the previous example that \sim^u is syntax-dependent, and may thus miss intuitive conclusions like q in Σ_2 . As correctly noticed in [Benferhat *et al.*, 1993], the missing conclusions have the particularity of being classically inferred from at least one MCS while their negations do not follow from any other MCS. For instance in Σ_2 , q follows from $\{p \wedge q\}$ but $\neg q$ does not follow from any MCS of Σ_2 . However, as recalled in Property 2, the set $\{\phi \in \mathcal{F} \mid \Sigma \sim^a \phi\}$ may still be inconsistent. The reason for this inconsistency is that there are two types of argumentative inferences:

- *Safe inferences*, which are formulas supported by arguments in every extension, and
- *Unsafe inferences*, which are formulas supported by arguments in some extensions only.

Unsafe inferences may lead to inconsistency and may even be counter-intuitive. Unlike \sim , the relation \sim^a does not distinguish between these two cases.

Example 3 *Consider Σ_2 from Example 1 and $\Sigma_4 = \{p, \neg p, p \rightarrow t\}$. Recall that q is supported by $\{\{p \wedge q\}, q\}$ whose compilation is $\{\{q\}, q\}$. The latter is not attacked and thus q can safely be inferred. Hence, $\Sigma_2 \sim^a q$ and $\Sigma_2 \sim q$. Consider now t in Σ_4 . It is supported by $A = \{\{p, p \rightarrow t\}, t\}$ whose compilation is A . Note that A is attacked by $\{\{\neg p\}, \neg p\}$ while p is crucial for proving t . Thus, A is weak and one may hardly rely on t . Note that $\Sigma_4 \sim^a t$ while $\Sigma_4 \not\sim t$.*

Unlike \sim , the relation \sim^a is also not closed under CN, thus it may miss intuitive conclusions as shown below.

Example 1 (Cont.) In Σ_3 , it is easy to check that $\Sigma_3 \sim^a q, t$ while $\Sigma_3 \not\sim^a q \wedge t$. However, $\Sigma_3 \sim q, t, q \wedge t$.

7 Conclusion

The paper discussed weaknesses of the existing argument-based logics, and showed that the definition of logical arguments is at their origin. It then introduced a novel logic that is powerful, more productive than existing argument-based and coherence-based logics, and satisfies desirable properties. The logic is based on the key idea of compilation of arguments, which amounts at forgetting in arguments' supports irrelevant variables and formulas for arguments' conclusions.

The computational complexity of logical arguments, forgetting operator and stable semantics are already studied respectively in [Parsons *et al.*, 2003; Lang *et al.*, 2003; Dunne and Bench-Capon, 2002]. Our future work consists of defining efficient algorithms for computing compiled arguments, and inferences drawn by the new logic.

Acknowledgments

Support from the ANR-3IA Artificial and Natural Intelligence Toulouse Institute is gratefully acknowledged.

References

- [Amgoud and Besnard, 2013] Leila Amgoud and Philippe Besnard. Logical limits of abstract argumentation frameworks. *Journal of Applied Non-Classical Logics*, 23(3):229–267, 2013.
- [Amgoud *et al.*, 2014] Leila Amgoud, Philippe Besnard, and Srdjan Vesic. Equivalence in logic-based argumentation. *Journal of Applied Non-Classical Logics*, 24(3):181–208, 2014.
- [Amgoud, 2014] Leila Amgoud. Postulates for logic-based argumentation systems. *International Journal of Approximate Reasoning*, 55(9):2028–2048, 2014.
- [Arieli *et al.*, 2018] Ofer Arieli, Annemarie Borg, and Christian Straßer. Reasoning with maximal consistency by argumentative approaches. *Journal of Logic and Computation*, 28(7):1523–1563, 2018.
- [Batens, 2003] Diderik Batens. A strengthening of the Rescher Manor consequence relations. *Logique & Analyse*, 46(183–184):289–313, 2003.
- [Benferhat *et al.*, 1993] Salem Benferhat, Didier Dubois, and Henri Prade. Argumentative inference in uncertain and inconsistent knowledge bases. In *Proceedings of the International Conference on Uncertainty in Artificial Intelligence (UAI)*, pages 411–419, 1993.
- [Besnard and Hunter, 2001] Philippe Besnard and Anthony Hunter. A logic-based theory of deductive arguments. *Artificial Intelligence*, 128(1-2):203–235, 2001.
- [Cayrol and Lagasque-Schiex, 1995] Claudette Cayrol and Marie-Christine Lagasque-Schiex. Non-monotonic syntax-based entailment: A classification of consequence relations. In *Proceedings of European Conference on Symbolic and Quantitative Approaches to Reasoning and Uncertainty (ECSQARU)*, pages 107–114, 1995.
- [Cayrol, 1995] Claudette Cayrol. On the relation between argumentation and non-monotonic coherence-based entailment. In *Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence (IJCAI)*, pages 1443–1448, 1995.
- [Dung, 1995] Phan Minh Dung. On the Acceptability of Arguments and its Fundamental Role in Non-Monotonic Reasoning, Logic Programming and n-Person Games. *Artificial Intelligence*, 77:321–357, 1995.
- [Dunne and Bench-Capon, 2002] Paul E. Dunne and Trevor J. M. Bench-Capon. Coherence in finite argument systems. *Artificial Intelligence*, 141(1/2):187–203, 2002.
- [Gorogiannis and Hunter, 2011] Nikos Gorogiannis and Anthony Hunter. Instantiating abstract argumentation with classical logic arguments: Postulates and properties. *Artificial Intelligence*, 175(9-10):1479–1497, 2011.
- [Lang and Marquis, 1998] Jérôme Lang and Pierre Marquis. Complexity results for independence and definability in propositional logic. In *Proceedings of the Sixth International Conference on Principles of Knowledge Representation and Reasoning (KR)*, pages 356–367, 1998.
- [Lang *et al.*, 2003] Jérôme Lang, Paolo Liberatore, and Pierre Marquis. Propositional independence-formula-variable independence and forgetting. *Journal of Artificial Intelligence Research*, 18:391–443, 2003.
- [Parsons *et al.*, 2003] Simon Parsons, Michael J. Wooldridge, and Leila Amgoud. Properties and complexity of some formal inter-agent dialogues. *Journal of Logic and Computation*, 13(3):347–376, 2003.
- [Rescher and Manor, 1970] N. Rescher and R. Manor. On inference from inconsistent premises. *Journal of Theory and Decision*, 1:179–219, 1970.
- [Vesic, 2013] Srdjan Vesic. Identifying the class of max-consistent operators in argumentation. *Journal of Artificial Intelligence Research*, 47:71–93, 2013.