Assessing Arguments with Schemes and Fallacies*

Pierre Bisquert Florence Dupin de Saint-Cyr Philippe Besnard
INRA, Montpellier, France, IRIT - CNRS, Univ. Paul Sabatier, Toulouse, France
pierre.bisquert@inra.fr {florence.bannay, philippe.besnard}@irit.fr

February 2019

Abstract

We present a logical framework allowing us to express assessment of facts (*is it proven?*) and arguments (*is it sound?*) together with a proof system to answer these questions. Our motivation is to clarify the notion of validity in the context of logic-based arguments along different aspects (such as the formulas used and the inference scheme). Originality lies in the possibility for the user to design their own argument schemes. We show that classical inference obtains when arguments are based on classical schemes (e.g. Hilbert axioms). We go beyond classical logic by distinguishing "proven" formulas from "uncontroversial" ones (whose negation is not proven). Hence a formal definition of a *fallacious argument*: it uses controversial formulas or schemes recognized as illicit. We express some rational arguments and fallacies in the form of schemes.
Keywords: Logic-based arguments, Fallacies, Soundness, Validity, Inference, Hilbert System

Contents

Introduction **2**

Language **4**

Soundness and Completeness of this Framework **6**

Computing licitness and soundness of an argument **7**
Example of Implementation of an Hilbert System 7

*This is a draft version, the paper is published in the proceedings of Logic Programming and Nonmonotonic Reasoning (LPNMR 2019), Philadelphia, PA, USA, 03/06/19-07/06/19, Yuliya Lierler, Stefan Woltran (Eds.), June 2019.

Reasoning with Schemes and Fallacies **8**
Sound and Fallacious use of the Expert Scheme 8
Encoding the Schemes of Some Usual Fallacies 10

Discussion and related work **13**

Introduction

Finding a good way to convince another individual (or oneself) is a crucial task that must have been done from the beginning of humanity and is still part of everyone's daily life. This may explain why this topic has been addressed by many researchers and is still a very hot topic which is studied from many different perspectives: philosophy, psychology, linguistics, logic, artificial intelligence, multi-agent communication, legal reasoning, etc.

There are at least two ways to interpret the word "argument" as expressed by Johnson and Blair [15]: 1) "An interaction, usually verbal and usually between two or more people, that is normally occasioned by a difference of opinion", we will call this option *Argumentation*, 2) "What someone makes or formulates (reasons or evidence) as grounds or support for an opinion (the basis for believing it)". We will call this second option *Assessing Arguments*. Hence the first sense is more related to dialogues where people *argue* by giving arguments and counter-arguments. In artificial intelligence, it concerns researchers working on action communication languages (see e.g. [2, 10, 25]), dialogues [30, 3], and abstract argumentation [8] (where arguments are represented by vertices of a graph whose arcs are attacks between them). The second sense is the one we are going to use in this paper where, as expressed by [15], "The account for argument cogency is that of acceptability, relevance or sufficiency (or good grounds)". In this context, arguments are structures containing reasons and conclusions such that the reasons are intended to be seen as proofs of the conclusions. However, *Argumentation* and *Assessing Arguments* coincide when a proof is simulated by a dialogue between an agent PRO (in favor of a formula) and an agent CON (against it) [17]. Moreover, inquiry dialogues as defined by [30] show also the need to bring together "validity" and communication act since, in this type of dialogues, participants aim to jointly find a "proof" for a particular formula.

In contrast with the first view of *Argumentation* where the question of what is an argument is often not evoked at all, the definition of an argument is at the heart of the *Assessing Arguments* research field. A first definition could be found in the diagrams of [31]. Later, [27] decomposes structurally an argument in five sub-components (the Claim, the Data supporting this Claim, the Warrant providing a licence to infer the Claim from the Data, the Backing for this Warrant and the Rebuttal condition that encapsulates exceptions). A more recent work by [29] defines argument schemes on the

basis of critical questions. Beyond this precise decomposition of an argument, there seems to be a consensus on the definition of a deductive structured argument with two parts (the premises and the claim) where the premises constitute a minimal proof of its claim. In that simpler context, assessing an argument amounts to check if it is *sound*, i.e. quoting Kelley [16]: "We evaluate [deductive] arguments by two basic standards: 1) Are the premises true? 2) How well do the premises support the conclusion?"

A major trend of research must be mentioned: the study of fallacy. Quoting Woods [32]: "in the broadest sense of the term, a fallacy is any error in reasoning. But the term is normally restricted to certain patterns of errors that occur with some frequency, usually because the reasoning involved has a certain surface plausibility. To the unwary, the premises of a fallacious argument seem relevant to the conclusion, even though they are not; or the argument seems to have more strength than it actually does. This is why fallacies are committed with some frequency". Understanding fallacious reasoning has two benefits: first, learn how to detect it in everyday life; second, progress in the understanding of what is a good argument by opposition to fallacies. This explains why fallacies have been broadly studied and seminal works [13] categorizing them in patterns is famous.

We propose a unified system for dealing with fallacies, since as far as we know, few authors [34, 33, 7] attempted to set a generic logical system that helps a user to assess an argument. Note that the introduction of meta-level predicates for assessing arguments has been explored but restricted to a dialectical argumentation framework [14, 24]. Indeed, in this paper we propose a logical system that takes an argument and some knowledge as input then, either the argument is sound and the licit schemes that were implicitly used in the argument are listed to the user, or the argument is not sound and the system answers that some premises are missing and/or gives a list of the fallacious schemes that were used. It is important to note that the aim of our work is not to formalize argumentation schemes "*à la* Walton/Toulmin", but to provide a logic-based formalization of arguments considered as structured proofs. In this regard, the argumentation schemes of Walton/Toulmin are particular cases of "non standard" inference rules hence can be seen as licit schemes in our framework.

The paper is organized as follows: we define a formal language that enables us to give a formal definition of concepts related to argument assessment (validity, soundness, etc.) wrt. a set of argumentation schemes, and we show that this can mimic classical logic when the schemes amount to classical inference, exemplifying it with Hilbert axioms. We give a list of fallacious schemes (fallacies) and a definition of a fallacious argument wrt. a set of recorded (potentially fallacious) schemes. We show that a fallacious argument will be detected as "non robust".

Language

We use a language L split into two parts: $L = L_0 \cup L_1$, L_0 is the language for describing the world, L_1 is a metalanguage for describing inferences between arguments and formulas based on schemes and facts defined in the language. L_0 is based on a finite set of user-defined predicates P_0, a finite set of variables $\mathcal{X}_0$ and a finite set of constants $\mathcal{C}_0$. A term of L_0 is a variable of $\mathcal{X}_0$ or a constant of $\mathcal{C}_0$, a vector of terms is denoted $\vec{T}$. An atom has the form $p(\vec{T})$ where p is a n-ary predicate of P_0 and either $n = 0$ and $\vec{T}$ is empty; or for all $i \in [1...n]$, $\vec{T}[i]$ is a term. Let At_0 be the set of all atoms based on P_0, $\mathcal{X}_0$ and $\mathcal{C}_0$, they will represent factual information on the world. Let $\mathcal{X}_1$ be a set of variable symbols (starting with a capital letter) that can represent any member of At_0 (atoms of L_0), they will be used in L_1. Let I be an index set (serving as scheme identifiers).

Definition 1 (Syntax of L_0 and L_1)

$$\begin{array}{lll} L_0: & \varphi, \psi & :: \; p(\vec{T}); X; \varphi \rightarrow \psi; \neg\varphi \\ L_1 & & :: \; licit(\Psi, \varphi); proven(\varphi); sound(\Psi, \varphi); unctrv(\varphi); robust(\Psi, \varphi) \\ where & & p(\vec{T}) \in At_0 \quad and \quad X \in \mathcal{X}_1 \quad and \quad \Psi :: \{\}; \{\varphi\} \cup \Psi \end{array}$$

Let $K \subseteq L_0$ be a set of formulas representing a set of factual knowledge, and $S \subseteq I \times 2^{L_0} \times L_0$ be a list of triples of the form (id, Ψ, φ) where id is an identifier from I, Ψ is a set of formulas of L_0 (the premise) and φ is a formula of L_0 (the conclusion), that represent the recorded "schemes" defined on L_0. S needs not to represent axioms that capture classical logic. We will see both the cases where S allows to capture classical logic and where S captures other kinds of schemes.

Licit, proven, sound are L_1-counterparts to validity and soundness in classical logic. An *argument is licit* if obtained by a substitution upon a recorded scheme. Since using an argument can be viewed as applying an inference scheme, a formula is *proven* if it can be reached by a sequence of inference schemes from the knowledge base. An *argument is sound* if it is licit and its premises are proven.

Definition 2 (Semantics of licit, proven and sound)

- $K, S \models_L licit(\Psi, \varphi)$ *iff there exist* $(id, A, \alpha) \in S$ *and a substitution* $\sigma : \mathcal{X}_1 \rightarrow At_0$ *s.t.* $(\Psi, \varphi) = (\sigma(A), \sigma(\alpha))$.
- $K, S \models_L proven(\varphi)$ *if* $\varphi \in K$ *or* $\exists \Psi \in 2^{L_0}$ *s.t.* $K, S \models_L sound(\Psi, \varphi)$.
- $K, S \models_L sound(\Psi, \varphi)$ *if* $K, S \models_L licit(\Psi, \varphi)$ *and* $\forall \psi \in \Psi$, $K, S \models_L proven(\psi)$.
- *The last two items are the only way to establish* $proven(\varphi)$ *and* $sound(\Psi, \varphi)$ *(*structural minimality*).*

Example 1 *Let K be a knowledge base expressing that it rains and that observing rain implies taking an umbrella* $K = \{rain, rain \rightarrow take(umbrella)\}$.

Let modus ponens be the only licit scheme, $S = \{(modusponens, \{X,\ X \rightarrow Y\}, Y)\}$.

The argument saying that "since it rains and due to the implication between rain and taking an umbrella, then the user should take an umbrella" is licit and sound, and allows us to prove that the user should take an umbrella. Formally, with $\Psi = \{rain, rain \rightarrow take(umbrella)\}$ *and* $\varphi = take(umbrella)$*, it holds that* $K, S \models_L licit(\Psi, \varphi)$ *and* $K, S \models_L sound(\Psi, \varphi)$ *and* $K, S \models_L proven(\varphi)$.

It is easier to visualize that a formula is proven by building a proof tree, according to Definition 3 and Proposition 1 (proof trees are used in some proofs).

Definition 3 (Proof tree, $\vdash_S$**)** *Given a knowledge base* K *and a set of schemes* S *s.t. scheme* $(id, \Psi, \varphi) \in S$*, a graph* $G = (V, E)$ *where each vertex of* V *contains exactly one formula of* L_0 *is a* proof tree *for* φ *wrt.* K *and* S *iff*

- *either* G *is a tree of only one node* v_0 *containing* φ *which is a leaf:* $G = (\{v_0\}, \emptyset)$ *and* $\varphi \in K$*,*
- *or* G *is a directed tree of root* v_0 *containing* φ *and* v_0 *is a node with* $k \leq \sup\{|\Psi| : (id, \Psi, \varphi) \in S\}$ *children* $v_1, \ldots, v_k$ *s.t.*
  - $\forall i \in [1, k]$*,* v_i *contains a formula* φ_i*,* v_i *is the root of a proof tree of* φ_i*,*
  - $(\{\varphi_1, \ldots, \varphi_k\}, \varphi)$ *is s.t. there exist* $(id, A, \alpha) \in S$ *and a substitution* $\sigma : \mathcal{X}_1 \rightarrow At_0$ *s.t.* $(\{\varphi_1, \ldots, \varphi_k\}, \varphi) = (\sigma(A), \sigma(\alpha))$.

Notation: $K \vdash_S \varphi$ *iff there exists a finite proof tree for* φ *wrt* K *and* S.

Proposition 1 $K, S \models_L proven(\varphi)$ *iff* $K \vdash_S \varphi$

In L_1 the expressions uncontroversial and robust are cautious counterparts of proven and sound as is standard [4, 9]. A formula is uncontroversial if its negation is not proven and either it is a fact or the conclusion of a robust argument where a robust argument is a licit one whose premises are all uncontroversial.

Definition 4 (semantics of uncontroversial and robust)

- $K, S \models_L unctrv(\varphi)$ *iff* $K, S \not\models_L proven(\neg\varphi)$ *and* $(\varphi \in K$ *or* $\exists \Psi \in 2^L$ *s.t.* $K, S \models_L robust(\Psi, \varphi))$.
- $K, S \models_L robust(\Psi, \varphi)$ *iff* $K, S \models_L licit(\Psi, \varphi)$ *and* $\forall \psi \in \Psi$*,* $K, S \models_L unctrv(\psi)$.

Example 1 (continued): *Supplement* K *with* $color(umbrella, yellow)$ *and* $\neg rain$*:* $K' = \{rain \rightarrow take(umbrella), rain, \neg rain, color(umbrella, yellow)\}$. *Then,* $K, S \models_L proven(take(umbrella))$ *and* $K, S \not\models_L uncontroversial($ *take (umbrella)). This is because the argument* $(\{rain, rain \rightarrow take(umbrella)\}$,

take(umbrella)) is no longer robust due to rain ceasing to be uncontroversial (now, both proven(rain) and proven(¬rain) hold).

Even if K is inconsistent, it is possible to infer that some non absurd formula hold, since K, $S \models_L$ unctrv(color(umbrella, yellow)) (there is no proof tree concluding ¬color(umbrella, yellow) because the fact does not exist and no implication concludes this negation). Such an inference system is paraconsistent, although not very powerful: e.g. modus tollens *is not a licit scheme in it.*

The next property shows that uncontroversial is a particular case of proven.

Proposition 2 *If $K, S \models_L unctrv(\varphi)$ then $K, S \models_L proven(\varphi)$.*

proof : If φ is uncontroversial then it is possible to build a particular proof tree for φ wrt. K and S (where each formula ψ of a node is such that $proven(\neg\psi)$ does not hold). Hence $K \vdash_S \varphi$. □

Soundness and Completeness of this Framework

In this section we show that the framework is sound and complete when the set of schemes S is licit and complete wrt. classical logic. For any set of schemes $S = \bigcup_{i \in I_S} (i, \Psi_i, \varphi_i)$, we say that S is *cl-valid* (standing for valid wrt classical logic) iff $\forall i \in I_S, \Psi_i \models \varphi_i$. We say that S is *cl-complete* iff $\models \varphi$ implies $\vdash_S \varphi$.

Proposition 3 (cl-validity) *Let $S = \bigcup_{i \in I_S} (i, \Psi_i, \varphi_i)$ be a set of cl-valid schemes, $I_S \subseteq I$, $\forall \varphi \in L_0$, $\forall K \subseteq L_0$, if $K, S \models_L proven(\varphi)$ then $K \models \varphi$.*

proof : Due to Prop. 1, $K, S \models_L proven(\varphi)$ implies $K \vdash_S \varphi$. Since S is cl-valid, this implies that $K \models \varphi$. □

Proposition 4 (cl-completeness) *Let $S = \bigcup_{i \in I_S} (i, \Psi_i, \varphi_i)$ be a cl-complete set, $I_S \subseteq I$, $\forall \varphi \in L_0$, $\forall K \subseteq L_0$, if $K \models \varphi$ then $K, S \models_L proven(\varphi)$*

proof : By mapping the classical proof tree of φ to a proof tree for $proven(\varphi)$ wrt. K and S (inverting the arcs) and using Prop. 1 we get $K, S \models_L proven(\varphi)$. □

Next, as expected, we show that introducing the notions of *uncontroversial* and *robust* provides a nice way to circumvent the ex falso quodlibet[1].

[1] The *ex falso quodlibet* expresses that from inconsistency anything can be deduced.

Proposition 5 (Escaping *ex falso quodlibet*) *Let* $S = \bigcup_{i \in I_S} (i, \Psi_i, \varphi_i)$ *be a set of schemes,* $I_S \subseteq I$*. If* S *is both cl-valid and cl-complete then* $\forall \varphi \in L_0$, $\forall K \subseteq L_0$, $\quad K, S \models_L unctrv(\varphi) \quad$ *iff* $\quad K \not\models \bot$ *and* $K \models \varphi$

proof : ($\Rightarrow$) $K, S \models_L unctrv(\varphi)$ by Prop. 2, $K, S \models_L proven(\varphi)$, by Prop. 3, $K \models \varphi$. Now, $K, S \models_L unctrv(\varphi)$ implies $K, S \not\models_L proven(\neg\varphi)$, due to Prop. 4 $K \not\models \neg\varphi$, i.e., $K \not\models \bot$ (since $K \models \bot$ implies $\forall \psi, K \models \psi$).
($\Leftarrow$) Due to Prop. 4, $K \models \varphi$ implies $K, S \models_L proven(\varphi)$. Due to Prop. 1, $K \vdash_S \varphi$. Assume that there is a node v containing a formula ψ in this proof tree s.t. $K, S \models_L proven(\neg\psi)$ then due to Prop. 3 $K \models \neg\psi$, moreover v should be s.t. $K, S \models_L proven(\psi)$ (by Def. 1). Due to Prop. 3 this implies $K \models \psi$), i.e., $K \models \bot$. Hence if $K \models \varphi$ and $K \not\models \bot$ then there is a proof tree for φ in K, S s.t. for each node containing any formula ψ in this tree $K, S \not\models_L proven(\neg\psi)$ which is a particular proof tree translating that $K, S \models_L unctrv(\varphi)$. $\square$

Computing licitness and soundness of an argument

A Prolog program has been implemented that assesses arguments. For the sake of efficiency, we define a predicate `arg/2` with which the user declares all the arguments to be used in the proof. The implementation is an encoding of the above definitions via the predicates *proven*, *licit sound*, *uncontroversial*, *robust*. In Prolog, these predicates have a parameter which can be set to an unbound variable that will contain a list of schemes and facts to be used to prove a formula.

Example 1 (continued): *The knowledge base given above is implemented as*

```
|?- proven([take(umbrella)], Schemes).
Schemes = [[modusponens, fact(rain),
            fact(implies(rain, take(umbrella)))]]
```

This means that we are able to prove take(umbrella) *based on the facts* rain *and* rain $\rightarrow$ take(umbrella) *and the* modus ponens *scheme.*

Example of Implementation of an Hilbert System

We now show how our framework captures classical logic by encoding a Hilbert system, namely Mendelson's axiom system for *implies* and *not*. These axioms are all valid and *modus ponens* preserves validity. As to completeness, the case is similar hence the schemes corresponding to this Hilbert system allows us to capture classical entailment, as is stated by the next corollary.

Corollary 1 (Inference with Hilbert Schemes)
Let $S_H = \{(hilbertK, \varnothing, X \to (Y \to X)), \quad (modusponens, \{X, X \to Y)\}, Y)$
$(hilbertS, \varnothing, (X \to (Y \to Z)) \to ((X \to Y) \to (X \to Z)))$
$(hilbertNot, \varnothing, (\neg Y \to \neg X) \to ((\neg Y \to X) \to Y))\}$
$\forall K \subseteq L_0$, $\forall \varphi \in L_0$, *we have* $K, S_H \models_L proven(\varphi)$ *iff* $K \models \varphi$

proof : The Hilbert axiomatic system has been shown to be valid, *modus ponens* has been shown to preserve validity hence S_H is valid, using Prop. 3 we get the implication from left to right, Hilbert system with modus ponens has been show to be complete, using Prop. 4 we get the reverse implication. □

It is then possible to check if $f \to f$ can be proven:

```
|?- proven([implies(f,f)], S).
S = [[modusponens,[modusponens,[hilbertK],[hilbertS]],[hilbertK]]]
```

This list gives the sequence of schemes that are used to prove $f \to f$: *hilbertK*, *hilbertS*, *modus ponens*, *hilbertK* and *modus ponens*.

Reasoning with Schemes and Fallacies

In this section, we show how our framework can be used to assess arguments using particular argument schemes or, possibly, fallacies.

Sound and Fallacious use of the Expert Scheme

We start with an example in which it is possible to produce "expert arguments", i.e. arguments using an expert's opinion to support a conclusion. Such arguments can be fallacious or sound according to the credibility of the expert (called "Authority" in the fallacy 2a "Appeal to Authority", see next section). Let the facts $K_1 = \{expert(doctorWho, weather)$, $topic(sunny, weather)$, $said(doctorWho, sunny)\}$ and schemes $S_1 = \{\{($ *expertarg*, $\{expert\ (Agent, Topic), topic\ (Claim, Topic), said\ (Agent, Claim)\}$, $Claim\}\}$ form the knowledge base. It is possible to construct the argument: $a_1 = (\{expert\ (doctorWho, weather), topic\ (sunny, weather), said\ (doctorWho\ , sunny)\}, sunny)$ such that we have:

$$K_1, S_1 \models_L licit(a_1) \;\; K_1, S_1 \models_L robust(a_1) \;\; K_1, S_1 \models_L unctrv(sunny)$$

Indeed, the argument a_1 follows exactly the "expert argument" scheme provided in S_1 (a_1 is thus licit) and its premises belong to K (it is the case of no contradicting piece of information) so it is robust. Since the argument is robust, its conclusion *sunny* is uncontroversial. If K does not contain *expert(doctorWho,weather)* then argument a_1 is no longer sound (nor robust) but it remains licit wrt. S_1.

Let us now observe how the addition of new information may give another result regarding the robustness of a_1 with the following knowledge base:

$$\begin{aligned} K_2 = K_1 \cup \{&nodiploma(doctorWho, weather),\\ &nodiploma(Agent, Topic) \rightarrow \neg expert(Agent, Topic)\},\\ S_2 = S_1 \cup \{&(modusponens, \{X, X \rightarrow Y\}, Y)\}. \end{aligned}$$

With the argument $a_2 = \big(\{nodiploma\ (doctorWho,\ weather),\ nodiploma\ (Agent, Topic) \rightarrow \neg\, expert(Agent, Topic)\}, \neg\, expert(doctorWho, weather)\big)$, we get:

$$\begin{array}{ll} K_2, S_2 \models_L licit(a_2) & K_2, S_2 \models_L sound(a_2) \\ K_2, S_2 \not\models_L robust(a_2) & K_2, S_2 \models_L proven(\neg expert(doctorWho, weather)) \end{array}$$

Because of a_2, the provability of one of the premises of a_1 has been challenged. I.e., we still have a proof for *expert(doctorWho,weather)*, but we also have a proof for its negation. And, thus, the conclusion of a_1 is not uncontroversial anymore:

$$K_2, S_2 \not\models_L unctrv(sunny) \qquad K_2, S_2 \models_L proven(sunny).$$

However, *sunny* is still proven since a_1 is still sound (its premises are still in the knowledge base). Our Prolog implementation provides the list of schemes used for assessing the argument or for proving the formula.

```
|?- proven([sunny], S).
S = [[expertarg]]
|?- proven([neg(expert(doctorWho, weather))], T).
T = [[modusponens]]
```

This provides the schemes used for proving respectively *sunny* (*expertarg*) and *¬(expert(doctorWho,weather))*. Since we are able to list every scheme that is used then it is possible to detect those that are regarded fallacious, and to let the user know about it. We illustrate this on the following knowledge base:

$$\begin{aligned} K_3 = K_1 \cup \{&young(doctorWho)\},\\ S_3 = S_1 \cup \{&(tooyoung, \{young(Agent)\}), \neg expert(Agent, weather)\}. \end{aligned}$$

This scheme expresses that a young person cannot be an expert about weather, which is fallacious (viewed as an instance of *Hasty Generalization* meaning that "young" implies "inexperienced" hence "not expert"). Yet, one can have argument

$$a_3 = (\{young(doctorWho)\}, \neg expert(doctorWho, weather)).$$

In this context, a_3 is licit regarding S_3 and it challenges a_1 in the same way as a_2. However, the possibility to detect this particular *tooyoung* fallacious scheme might allow to prompt the user to change its arguments or provide grounding for its (hitherto fallacious) scheme.

Encoding the Schemes of Some Usual Fallacies

In this section, we show how our framework is able to handle usual fallacies. Since Aristotle's *On Sophistical Refutations*, there have been a lot of work on fallacies, including [13], along time the list of fallacies is growing, and is exposed in books or even on web pages[2]. Here, we choose to use the classification given by [16] who studied fallacies with the same goal as many authors including Aristotle, i.e., first for helping people to identify and avoid them, second because "understanding why these patterns of arguments are fallacious will help us understand the nature of good reasoning". In this section, we propose to examine fallacies that Kelley discussed in [16], Chapter 5. For example we do not consider fallacies that refer to an opponent's argument like "strawman" (misrepresenting someone's argument to make it easier to attack). Quoting Kelley, "the varieties of bad reasoning are too numerous to catalog here" hence we restrict to Kelley's four categories:

1. Subjectivist fallacies: these are inferences that involve the violation of objectivity in one way or another.
   (a) Subjectivism: "I believe in p" or "I want p" **hence** p holds.
   (b) Appeal to majority: The majority believes p **hence** p is true.
   (c) Appeal to emotion: use (explicitly or implicitly) emotion instead of evidence to make accepted a belief.
   (d) Appeal to force (*Argumentum ad Baculum*): use a threat instead of evidence (which may be regarded as an appeal to the emotion "scared").
2. Fallacies involving credibility:
   (a) Appeal to Authority (*Argumentum ad Verecundiam*): agent A says p **hence** p is true. It is a fallacy when A has not been proven to be competent and objective, when the conditions of credibility are not satisfied.
   (b) *Ad Hominem*: using a negative trait of a speaker as evidence that his statement is false: A says p, A has some negative trait **hence** p is false.
3. Fallacies of Context: "jumping to conclusions."
   (a) False Alternative[3]: Either p or q, $\neg q$ **hence** p which is deductively valid but the soundness depends on whether the premises take into account all the relevant alternatives.
   (b) *Post Hoc*[4]: X occurred before Y **hence** X caused Y.
   (c) Hasty Generalization: drawing conclusions too quickly, on the basis of insufficient evidence (with not enough variety to be representative).
   (d) Accident or Hasty application: applying a generalization to a spe-

[2] See e.g. https://www.logicallyfallacious.com/tools/lp/Bo/LogicalFallacies
[3] Also called False Dichotomy when the premises posit just two alternatives.
[4] This is short for *post hoc ergo propter hoc*: "after this, therefore because of this."

cial case without regard to the circumstances that make the case an exception to the general rule.

(e) Slippery Slope: Action X will lead to Y that will lead to Z, Z is very bad **hence** [5] X should be avoided.

(f) Composition (and Division): inferring p is true of a part (the whole) must be true of the whole (a part) without considering whether the nature of p makes it rational.

4. Fallacies of Logical Structure

(a) Begging the Question (Circular Argument): p **hence** p, usually p is formulated in two different ways[6].

(b) Equivocation: a word used in premise and conclusion switches meaning.

(c) Appeal to Ignorance: $\neg p$ has not been proven true **hence** p is true[7].

(d) Diversion: changing the issue in the middle of an argument. Another form of diversion is called the *Straw man* argument: distorts an opponent's position and then refutes it. An extreme form is the *Non sequitur* fallacy when the premises are completely unrelated to the conclusion.

Table 1 is a first attempt to encode the schemes that could be associated to these fallacies in our framework. *We regard all the items followed by a star as rational schemes.* However, a number of instances of these schemes are fallacious because they are used with unproven premises. As already said, this is the case for the *Authority* argument which is not fallacious by itself, it is fallacious when $expert(A, T)$ fails to be $unctrv$. It is also the case for *False Alternative*: the scheme is rational but the premise may not be true in the context, i.e., there may be other alternatives than Y when X does not hold ($\neg X \rightarrow Y$ is false).

The items for which no scheme is proposed in the table are those that either are based on natural language or semantic interpretations like *Emotion*, *Force*, *Equivocation* and *Non Sequitur*. *Appeal to Ignorance* is of another type since it is a meta-argument that speaks about provability; we could encode it with $(f4c, \{\neg holds(X)\}, \neg X)$, however this would require to have a more complex definition of the language L_0 that includes the predicate *holds*. This would lead to a more complex definition of the semantics of L.

Definition 5 *Given a knowledge base K and a set of (rational and sophistic) schemes $S = S_R \cup S_S$ and an argument $a \in 2^{L_0} \times L_0$:*

$$a \text{ is fallacious wrt. } K, S \text{ iff } K, S_R \not\models robust(a)$$

[5] There could be any number of items in the series of projected consequences.

[6] This fallacy occurs when the circle is enlarged to include more than one step: The conclusion p is supported by premise q, which in turn is supported by p (though there could be any number of intervening steps).

[7] One application is the legal principle that a person is innocent until proven guilty.

Fallacy	**Scheme**
Subjectivism	$(f1a, \{likeable(X)\}, X)$
Majority	$(f1b, \{majoritarian(X)\}, X)$
Authority*	$(f2a,\ \{expert(A,T),\ topic(X,T),\ said(A,X)\},\ X)$
Ad Hominem	$(f2b, \{said(A,X), \neg likeable(A)\},\ \neg X)$
False Alternative*	$(f3a, \{X \rightarrow \neg Y,\ \neg X \rightarrow Y, \neg X\}, Y)$
Post Hoc	$(f3b, \{before(X,Y)\},\ cause(X,Y))$
Hasty Generalization	$(f3c, \{hasProp(X,P),\ Y \rightarrow X\},\ hasProp(Y,P))$
Accident	$(f3d, \{hasProp(X,P),\ X \rightarrow Y\},\ hasProp(Y,P)$
Slippery Slope	$(f3e, \{cause(X,Y),\ cause(Y,Z)\ \neg likeable(Z)\},\ \neg do(X)\}$
Composition	$(f3f, \{hasProp(X,P),\ part(X,Y)\},\ hasProp(Y,P))$
Begging the Question*	$(f4a, \{X\}, X)$

Table 1: Proposal of fallacious schemes encoding.

This definition allows us to emphasize the fallacious aspects of arguments in our model: $a = (\Psi, \varphi)$ is fallacious in the following cases:

1. $\exists \psi \in \Psi,\ K, S_R \not\models_L proven(\psi)$: it uses a premise that is not rationally proven,
2. $\forall \psi \in \Psi,\ K, S_R \models_L proven(\psi)$ and $\exists \psi' \in \Psi\ K, S_R \models_L proven(\neg \psi')$: there is a controversial premise,
3. $K, S_R \not\models_L licit(a)$: it uses a sophistic scheme or an unrecorded scheme.

The last case allows us to characterize the *Non Sequitur* fallacy which seems appropriate here. This also enables us to cover cases like *Appeal to Emotion* and *Appeal to Force* where the use of a premise that refers to emotion or threat is not following any rational deductive scheme towards a conclusion. The occurrence of the third case may disappoint a user by pointing out that her argument is not licit because not based on a recorded scheme. However our program will inform her about all the licit schemes and uncontroversial premises that she has used.

Proposition 6 *Given a knowledge base K and a set of schemes $S = S_R \cup S_S$ s.t. $K \cup \{(\sigma(\Psi) \rightarrow \sigma(\varphi) \mid \sigma \in \mathcal{X}_1 \rightarrow At_0,\ (id, \Psi, \varphi) \in S_R\} \nvdash \bot$, for any argument $a \in 2^{L_0} \times L_0$, a is fallacious wrt. K, S iff $K, S_R \not\models_L sound(a)$*

This last result shows that when the rational schemes do not allow to infer inconsistent formulas from the knowledge base then a fallacious argument is simply an unsound argument. Hence a "non fallacious" argument uses the rational schemes with proven premises (which cannot be controversial in

this context). This goes beyond classical logic because schemes $((id, \Psi, \varphi))$ need not be cl-valid ($\Psi \models \varphi$) to be applied (i.e. to belong to S_R).

Discussion and related work

This framework, and its Prolog implementation, allows us to assess arguments with regard to a knowledge base and a set of argumentation schemes. A merit of our work is to clarify various forms of validity depending on the nature of the target. Namely, we have distinguished three targets: logical deduction, instantiated argument, generic argument scheme. Each of them can be associated with a different definition of validity, which leads us to propose different names for them: "valid/unvalid" applies to a deduction between logical formulas, "licit/illicit" and "sound/unsound" concern an instantiated argument, a "rational scheme" is opposed to a "sophism" in order to qualify an argument scheme. More precisely, an instantiated argument is said to be licit if it follows a recognized scheme. It is said to be sound (or robust) if it has proven (or uncontroversial) premises. Distinguishing between proven and uncontroversial formulas allows in turn to circumvent the ex falso quodlibet that derives anything. Our framework is flexible enough to represent Hilbert axioms, granting the possibility to express classical logic, but could also be used with "argument schemes" or even sophistic schemes. One benefit of the encoding in a formal logical language is the ability to express and decide about soundness of arguments in the logical language itself.

The idea to axiomatize invalid statements is not new: it is called rejection calculi, first introduced by Jan Łukasiewicz [18] and has been developed for different logics like classical logic, intuitionistic logic, modal logics [26, 22, 11]. Some proposals were dedicated to the detection of one particular fallacy, like [19]'s dialogue game for detecting the fallacy of *petitio principii*. In contrast, our approach deals with multiple fallacies and is highly flexible since it may be used with any user-defined inference scheme. For instance, by allowing the user to define non-classical inference schemes, our system may allow the closed world assumption or defeasible reasoning. This unified formalism may also allow us to better circumscribe usual commonsense inferences done with e.g., causes and counterfactuals that should deserve specific schemes (as also claimed by [20]).

While this paper is about the assessment of arguments, there are interesting links with the other interpretation of the word "argument", that is the subject of dialectical argumentation. The latter focuses on the study of argument validity in the sense of winning the dispute: "can this argument defend itself against any other argument?". We take the viewpoint of logic through argumentation, trying to extract the intrinsic validity of an argument, i.e. its soundness, from the way it is built. Some approaches like ABA [21], ASPIC+[23] and Carneades [12] are relating structured arguments to

Dung like interaction argumentation, and base the assessment of argument on its relation to counter-arguments. The problem with such approaches is that they use argumentation semantics, where these semantics do not depend on the intrinsic content of arguments but is only based on the interactions, leading to counter-intuitive results as proven in [1]. An idea could be to detect fallacies based on the existence of attacks. Moreover, if no counter-argument has been stated against a given (fallacious) argument, this does not mean that the argument is a correct evidence for its conclusion.

Our work is but an opening for a number of new studies. Thus, it would be interesting to study what schemes can be added to cover more types of rational reasoning (and their possible flaws). Another perspective is to extend our definitions so as to allow for more complex arguments, e.g. directly referring to another argument (as a premise or counterargument) with the long term view to handle dialogues. Our aim is not to help users build convincing tricky fallacies, our aim is to help people build efficiently *sound* arguments and to allow them to fight fallacies: the closer a fallacy is from a sound argument the more the agent can be inclined to use it especially in case of low cognitive availability [6].

Our long term goal is to use our framework to offer a protocol governing the authorized moves in a dialogue. It would be worth incorporating it as a part of a "dialogue support system" that could ensure for instance the correct use of the speech act *Argue* (that commits the agent to be able to provide a sound proof of her claim from some premises). So, our proposal enables an automatic verification of compliance of this speech act with regard to a set of rational schemes. The dialog support system could make the user aware of her biased reasoning and prompt her to give "better" grounds for her argumentation. An idea could be to take into account the notion of critical questions [27, 29, 5] in order to assess arguments and following the work of Verheij [28] we could help a user to provide more accurate justifications for any unproven premises (via other arguments), or even to introduce new justified schemes in her base.

References

[1] Leila Amgoud and Philippe Besnard. Logical limits of abstract argumentation frameworks. *Journal of Applied Non-Classical Logics*, 23(3):229–267, 2013.

[2] Leila Amgoud, Nicolas Maudet, and Simon Parsons. An argumentation-based semantics for agent communication languages. In *15th Eur. Conf. Artif. Intell.*, volume 2, pages 38–42, July 2002.

[3] Trevor Bench-Capon, Paul Dunne, and Paul Leng. A dialogue game for dialectical interaction with expert systems. In *12th Ann. Conf. Expert Syst. & Appl.*, pages 105–113, 1992.

[4] Salem Benferhat, Didier Dubois, and Henri Prade. Argumentative inference in uncertain and inconsistent knowledge bases. In *9th C. on Uncertainty in AI*, pages 411–419, 1993.

[5] Philippe Besnard, Alejandro Javier García, Anthony Hunter, Sanjay Modgil, Henry Prakken, Guillermo Ricardo Simari, and Francesca Toni. Introducing structured argumentation. *Argument & Computation*, 5(1):1–4, 2014.

[6] Pierre Bisquert, Madalina Croitoru, Florence Dupin de Saint Cyr, and Abdelraouf Hecham. Formalizing Cognitive Acceptance of Arguments: Durum Wheat Selection Interdisciplinary Study. *Minds & Machines*, 27(1):233–252, avril 2017.

[7] Marcello D'Agostino and Sanjay Modgil. Classical logic, argument and dialectic. *Artificial Intelligence J.*, 262:15–51, 2018.

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

[9] Morten Elvang-Gøransson, Paul Krause, and John Fox. Dialectic reasoning with inconsistent information. In *9th C. on Uncertainty in AI*, pages 114–121, 1993.

[10] FIPA. ACL message structure specification. *Foundation for Intelligent Physical Agents, http://www.fipa.org/specs/fipa00061/SC00061G.html (30.6.2004)*, 2002.

[11] Valentin Goranko. Refutation systems in modal logic. *Studia Logica*, 53(2):299–324, 1994.

[12] Thomas F Gordon, Henry Prakken, and Douglas Walton. The Carneades model of argument and burden of proof. *Artificial Intelligence J.*, 171(10-15):875–896, 2007.

[13] Charles L. Hamblin. *Fallacies.* London: Methuen, 1970.

[14] Anthony Hunter. Reasoning about the appropriateness of proponents for arguments. In *23rd AAAI Conf. on Artificial Intelligence*, pages 89–94, 2008.

[15] Ralph Henry Johnson and J. Anthony Blair. *Logical self-defense.* New York: IDEA, 2006.

[16] David Kelley. *The art of reasoning: An introduction to logic and critical thinking.* New York: W.W. Norton & Company, 2013.

[17] Kuno Lorenz and Paul Lorenzen. *Dialogische Logik.* Darmstadt: WBG, 1978.

[18] Jan Lukasiewicz. *Aristotle's Syllogistic from the Standpoint of Modern Formal Logic, 2nd edition.* Oxford: Clarendon Press, 1957.

[19] Jim D Mackenzie. Question-begging in non-cumulative systems. *J. of Philosophical Logic*, 8(1):117–133, 1979.

[20] Elliott Mendelson. *Introduction to Mathematical Logic.* CRC Press, 6th edition, 2015.

[21] Sanjay Modgil and Henry Prakken. A general account of argumentation with preferences. *Artificial Intelligence J.*, 195:361–397, 13.

[22] Johannes Oetsch and Hans Tompits. Gentzen-type refutation systems for three-valued logics with an application to disproving strong equivalence. In *11th Int. Conf. on Logic Programming and Nonmonotonic Reasoning*, volume 6645 of *LNCS*, pages 254–259, 2011.

[23] Henry Prakken. An abstract framework for argumentation with structured arguments. *Argument & Computation*, 1(2):93–124, 2010.

[24] Henry Prakken, Adam Z. Wyner, Trevor J. M. Bench-Capon, and Katie Atkinson. A formalization of argumentation schemes for legal case-based reasoning in ASPIC+. *J. Logic & Computation*, 25(5):1141–1166, 2015.

[25] Munindar P Singh. Agent communication languages: Rethinking the principles. *Computer*, 31(12):40–47, 1998.

[26] Tomasz F. Skura. Refutation systems in propositional logic. In Dov Gabbay and Franz Guenthner, editors, *Handbook of Philosophical Logic*, volume 16, pages 115–157. Springer, 2nd edition, 2011.

[27] Stephen Toulmin. *The Uses of Argument*. Cambridge University Press, 1958.

[28] Bart Verheij. Evaluating arguments based on Toulmin's scheme. *Argumentation*, 19(3):347–371, Dec 2005.

[29] Douglas Walton and Thomas F. Gordon. Critical questions in computational models of legal argument. In *Argumentation in Artificial Intelligence and Law Workshop*, pages 103–111. Wolf Legal Publishers, June 10 2005.

[30] Douglas Walton and Erik C. W. Krabbe. *Commitment in dialogue: Basic concepts of interpersonal reasoning*. Albany: State University of New York Press, 1995.

[31] John Henry Wigmore. *The Principles of Judicial Proof*. Little, Brown, 2nd edition, 1931.

[32] John Woods. Is the theoretical unity of the fallacies possible? *Informal Logic*, XVI:77–85, 1994.

[33] Michael Wooldridge, Peter McBurney, and Simon Parsons. On the meta-logic of arguments. In *2nd Int. W. on Arg. in Multi-Agent Systems*, volume 4049 of *LNCS*, pages 560–567, 2006.

[34] Tangming Yuan, Suresh Manandhar, Tim Kelly, and Simon Wells. Automatically detecting fallacies in system safety arguments. In *Principles and Practice of Multi-Agent Systems*, volume 9935 of *LNCS*, pages 47–59. Springer, 2016.