Five weaknesses of ASPIC^{+}

Leila Amgoud

IRIT – CNRS
amgoud@irit.fr

Abstract. This paper analyzes the ASPIC+ argumentation system. It shows that it is grounded on two monotonic logics which are not Tarskian ones. Moreover, the system suffers from five main problems: i) its logical formalism is ill-defined, ii) it may return undesirable results, iii) it builds on some counter-intuitive assumptions, iv) it violates some rationality postulates, and v) it allows counter-intuitive instantiations.

1 Introduction

An *argumentation system* consists of a set of *arguments* and an *attack relation* among them. A *semantics* is applied for evaluating the arguments. It computes one (or more) set(s) of acceptable arguments, called *extension*(s). One of the most abstract systems is that proposed by Dung in [6]. It leaves the definition of the notion of argument and that of attack relation completely unspecified. Thus, the system can be instantiated in different ways for reasoning about defeasible information stored in a knowledge base. An instantiation starts with an underlying *logic* which is a pair $(\mathcal{L}, \mathtt{CN})$. The part $\mathcal{L}$ represents the logical language in which the information of the knowledge base are encoded. It is thus, a set of well-formed formulas. The second component of a logic is its consequence operator $\mathtt{CN}$. It represents the reasoning patterns that are used. In an argumentation system, $\mathtt{CN}$ is used for generating arguments from a knowledge base and also for defining attacks among arguments. It is worth mentioning that in almost all existing argumentation systems, the underlying logic $(\mathcal{L}, \mathtt{CN})$ is monotonic (see [7]). This makes the construction of arguments monotonic, that is an argument remains an argument even when the knowledge base is extended with arbitrary information. However, the status of an argument may change. Consequently, the logic produced by the argumentation system is nonmonotonic. In sum, an argumentation system for defeasible reasoning is grounded on a monotonic logic and produces a nonmonotonic one.

Recently, Prakken proposed an instantiation of Dung's system [10], called ASPIC^{+}. It is an extended version of ASPIC system which is developed in [1]. ASPIC^{+} takes as input an unspecified set of formulas, a contrariness function between formulas and two sets of rules: strict rules and defeasible ones. These rules may either represent knowledge (like penguins do not fly, generally birds fly) or inference rules (like modus ponens). From these inputs, arguments are built and attacks between arguments are specified. Thus, the only parameter which is somehow left unspecified is the underlying logic. In [10], it is claimed that this system satisfies the rationality postulates defined in

[5]. In [9], the authors claimed that the logic underlying ASPIC$^+$ is too general that it captures even Tarskian monotonic logics [11]. It is sufficient to assume that strict rules conform to a Tarskian consequence operator (see the citation below):

> "*If the strict rules in a c-SAF conform to a Tarskian consequence operator, then it should be obvious to see that the cSAF is c-classical*".

Our aim in this paper is to investigate the underpinnings of this system, especially since there are no formal results proving some claims like the previous one on Tarski's logics. We study two main issues: the properties of ASPIC+ and the kind of monotonic logics underlying this system.

Regarding the first objective, we show that ASPIC+ suffers from serious technical problems. Namely, the basic concepts are ill-defined. This is particularly the case of the logical formalism on which the system is built. Consequently, counter-intuitive results may be given as outputs of the system, and the rationality postulates proposed in [5] may be violated. We also show that the system is grounded on counter-intuitive assumptions leading thus to unjustified conclusions. The last weakness of the system consists of allowing counter-intuitive instantiations. Indeed, unlike what is claimed in [9], the logical formalism of ASPIC+ is unable to capture classical logics. The main problem comes from the definition of consistency which is too poor.

In a second part of the paper, we investigate the monotonic logic underlying ASPIC+. Prakken claims that strict and defeasible rules may play two roles: either they encode information of the knowledge base, in which case they are part of the language $\mathcal{L}$, or they represent inference rules, in which case they are part of the consequence operator $\mathtt{CN}$. In this paper, we define precisely the two corresponding logics, and show that they are monotonic but are not among the Tarskian ones. Moreover, we show that when rules encode information, then the system is fully instantiated apart from the kind of formulas that may be stored in $\mathcal{L}$. This unique generality is fatal for the system since it may lead to irrational systems. When the rules are part of the consequence operator, the logic is far from capturing Tarski's logics [11], contrarily to what is claimed in [9]. Indeed, not only it is unable to capture the monotonic logics that do not allow negation in their language, but it is also unable to capture even those which do have negation.

The paper is structured as follows: In section 2, we recall ASPIC$^+$ and show its limits. Section 3 recalls the monotonic logics of Tarski. In section 4, we discuss deeply the case where rules encode knowledge, while in section 5 we study the case where they encode inference rules. The last section is devoted to some concluding remarks.

2 APIC$^+$ argumentation framework

In [10], Prakken has proposed an instantiation of Dung's framework, called ASPIC$^+$. It considers an abstract logical language $\mathtt{L}$, that is a language which may be instantiated in different ways. It may for instance contain propositional formulas, etc. The only requirement on $\mathtt{L}$ is that it is equipped with a notion of *contrariness*, denoted by $\bar{\ }$.

Definition 1 (Contrariness). *Let* L *be a logical language and* $^-$ *is a* contrariness *function from* L *to* $2^{\mathtt{L}}$. *For* $x, y \in \mathtt{L}$, *if* $x \in \bar{y}$, *then if* $y \notin \bar{x}$ *then* x *is called a* contrary *of* y, *otherwise* x *and* y *are called* contradictory.

Remark 1 *It is worth mentioning that the above definition of contrariness does not capture the real intuition of contrary as discussed by several philosophers and logicians (e.g. [4]). Indeed, a formula* x *is a contrary of* y *iff they cannot be both true but they can both be false. They are contradictory if the truth of one implies the falsity of the other and vise versa. Let us consider the following simple example:*

Example 1 *Assume that* $\mathtt{L} = \{>, \geq, <, \leq, =, \neq\}$. *For instance,* $\bar{>} = \{<, \leq, =\}$ *and* $\bar{<} = \{>, \geq, =\}$. *Note that* $>$ *is the contrary of* $<$ *and vise versa and* $< \in \bar{>}$ *and* $> \in \bar{<}$. *According to Def. 1,* $>$ *and* $<$ *are contradictory while this is not the case.*

In addition to the language L, two kinds of rules are assumed: *strict* rules and *defeasible* ones.

Definition 2 (Rules). *Let* $x_1, \ldots, x_n, x$ *be elements of* L. *A* strict rule *is of the form* $x_1, \ldots, x_n \rightarrow x$ *meaning that if* $x_1, \ldots, x_n$ *hold then without exception* x *holds. A* defeasible rule *is of the form* $x_1, \ldots, x_n \Rightarrow x$ *meaning that if* $x_1, \ldots, x_n$ *hold then presumably* x *holds.* $\mathcal{R}_s$ *(resp.* $\mathcal{R}_d$*) stands for the set of all strict (resp. defeasible) rules with* $\mathcal{R}_s \cap \mathcal{R}_d = \emptyset$.

Remark 2 *In [10], no restrictions neither on the formulas of* L *nor on the rules are imposed. Consequently, the two connectors* $\rightarrow$ *and* $\Rightarrow$ *may be used in elements of* L. *Thus, rules can be defined in several ways as shown in the next example.*

Example 2 *The rule* $x \rightarrow (y \rightarrow z)$ *is strict whereas the rules* $a \Rightarrow (b \rightarrow c)$, $(a \rightarrow b) \Rightarrow (x \rightarrow y)$ *are defeasible.*

A notion of consistency is associated to this logical formalism as follows.

Definition 3 (Consistency). *A set* $X \subseteq \mathtt{L}$ *is* consistent *iff* $\nexists\, x, y \in X$ *such that* $x \in \bar{y}$, *otherwise it is* inconsistent.

This notion of consistency was used in ASPIC system [1]. While it is sufficient in that system since the language L contains only literals, it is too poor when considering an arbitrary language. Indeed, it only captures *binary* conflicts between formulas. That is, it does not capture ternary or more conflicts. Let us consider the following example:

Example 3 *Let* L *contain propositional formulas and let* $^-$*stand for classical negation. The set* $X = \{x, x \rightarrow y, \neg y\}$ *is consistent wrt Definition 3 (whereas it is inconsistent wrt propositional logic).*

Remark 3 *This example shows clearly that this formalism cannot capture classical logics (e.g. propositional logic, first order logic, ...) contrarily to what is claimed in [9]. For instance, even if the set of strict rules represent all the valid inference mechanisms of propositional logic, their notions of consistency are different as witnessed by the above example. One formalism declares the set* X *as consistent while the other declares it inconsistent.*

In [10], arguments are built from a knowledge base $\mathcal{K}$. It may contain four categories of information: *axioms* (which are certain information) ($\mathcal{K}_n$), *ordinary* premises ($\mathcal{K}_p$), *assumptions* ($\mathcal{K}_a$) and *issues* ($\mathcal{K}_i$). The set $\mathcal{K}_n$ is assumed to be consistent. These subbases are disjoint and $\mathcal{K} = \mathcal{K}_n \cup \mathcal{K}_p \cup \mathcal{K}_a \cup \mathcal{K}_i$.

Remark 4 *In [10], it is claimed that strict and defeasible rules may encode respectively certain and defeasible information. In this case, it is clear that strict rules and axioms (elements of $\mathcal{K}_n$) represent the same kind of information. Similarly, ordinary premises and defeasible rules refer to the same kind of information. Thus, the system allows some redundancies and this makes its instantiation by users unclear.*

In what follows, for a given argument, the function $\mathtt{Prem}$ (resp. $\mathtt{Conc}$, $\mathtt{Sub}$, $\mathtt{DefRules}$, $\mathtt{TopRule}$) returns all the formulas of $\mathcal{K}$ which are involved in the argument (resp. the conclusion of the argument, its sub-arguments, the defeasible rules that are used, and the last rule used in the argument).

Definition 4 (Argument). *An* argument A *is:*

- x *if* $x \in \mathcal{K}$ *with* $\mathtt{Prem}(A) = \{x\}$, $\mathtt{Conc}(A) = x$, $\mathtt{Sub}(A) = \{A\}$, $\mathtt{DefRules}(A) = \emptyset$, $\mathtt{TopRule}(A)$ = *undefined.*
- $A_1, \ldots, A_n \longrightarrow x$[1] *(resp.* $A_1, \ldots, A_n \Longrightarrow x$*) if* $A_1, \ldots, A_n$*, with* $n \geq 0$*, are arguments such that there exists a strict rule* $\mathtt{Conc}(A_1), \ldots, \mathtt{Conc}(A_n) \to x$ *(resp. a defeasible rule* $\mathtt{Conc}(A_1), \ldots, \mathtt{Conc}(A_n) \Rightarrow x$*).*
  $\mathtt{Prem}(A) = \mathtt{Prem}(A_1) \cup \ldots \cup \mathtt{Prem}(A_n)$,
  $\mathtt{Conc}(A) = x$,
  $\mathtt{Sub}(A) = \mathtt{Sub}(A_1) \cup \ldots \cup \mathtt{Sub}(A_n) \cup \{A\}$,
  $\mathtt{DefRules}(A) = \mathtt{DefRules}(A_1) \cup \ldots \cup \mathtt{DefRules}(A_n)$ *(resp.* $\mathtt{DefRules}(A) = \mathtt{DefRules}(A_1) \cup \ldots \cup \mathtt{DefRules}(A_n) \cup \{\mathtt{Conc}(A_1), \ldots \mathtt{Conc}(A_n) \Rightarrow x\}$*),*
  $\mathtt{TopRule}(A) = \mathtt{Conc}(A_1), \ldots, \mathtt{Conc}(A_n) \to x$ *(resp.* $\mathtt{TopRule}(A) = \mathtt{Conc}(A_1), \ldots, \mathtt{Conc}(A_n) \Rightarrow x$*).*

An argument A is strict *if* $\mathtt{DefRules}(A) = \emptyset$*,* defeasible *if* $\mathtt{DefRules}(A) \neq \emptyset$*,* firm *if* $\mathtt{Prem}(A) \subseteq \mathcal{K}_n$*, and* plausible *if* $\mathtt{Prem}(A) \nsubseteq \mathcal{K}_n$*.*

Arguments may not have equal strengths in [10]. Thus, a partial preorder denoted by $\succeq$ is available. For two arguments, A and B, the notation $A \succeq B$ means that A is at least as good as B. The strict version of $\succeq$ is denoted by $\succ$. This preorder should satisfy two basic requirements: The first one ensures that firm and strict arguments are stronger than all the other arguments, while the second condition says that a strict rule cannot make an argument weaker or stronger.

Definition 5 (Admissible argument ordering). *Let $\mathcal{A}$ be a set of arguments. A partial preorder $\succeq$ on $\mathcal{A}$ is an* admissible argument ordering *iff for any pair $A, B \in \mathcal{A}$:*

R_1**:** *if A is firm and strict and B is defeasible or plausible, then $A \succ B$.*
R_2**:** *if $A = A_1, \ldots, A_n \to x$, then $\forall i = 1, n$, $A_i \succeq A$ and for some $i = 1, n$, $A \succeq A_i$.*

[1] The symbols $\longrightarrow$ and $\Longrightarrow$ are used to denote arguments. In [10], arguments are denoted by $\to$ and $\Rightarrow$ while these latter are used for defining rules.

Remark 5 *The previous requirements reduce the generality of the preference relation. For instance, requirement R_1 is violated by the preference relation proposed in [3]. In that paper, each argument promotes a value and the best argument is the one which promotes the most important value. Assume that there are two arguments A and B which promote respectively values v_1 and v_2 with v_1 being more important than v_2. If we assume that A is defeasible and B is both strict and firm, then from requirement R_1, B should be preferred to A. However, in [3], A is strictly preferred to B.*

Since information in a knowledge base $\mathcal{K}$ may be inconsistent, arguments may be conflicting too. In [10], Prakken has modeled three ways of attacking an argument: undermining its conclusion, or one of its premises or one of its defeasible rules. The following definition reports exactly

Definition 6 (Attacks). *Let A and B be two arguments.*

- A undercuts B *(on C) iff* $\texttt{Conc}(A) \in \bar{C}$ *with* $C \in \texttt{Sub}(B)$ *and of the form* $C_1, \ldots, C_n \Longrightarrow x$.
- A rebuts B *(on C) iff* $\texttt{Conc}(A) \in \bar{x}$ *for some* $C \in \texttt{Sub}(B)$ *of the form* $C_1, \ldots, C_n \Longrightarrow x$. A contrary-rebuts B *iff* $\texttt{Conc}(A)$ *is a contrary of* x.
- A undermines B *iff* $\texttt{Conc}(A) \in \bar{x}$ *for some* $x \in \texttt{Prem}(B) \setminus \mathcal{K}_n$. A contrary-undermines B *iff* $\texttt{Conc}(A)$ *is a contrary of* x *or if* $x \in \mathcal{K}_a$.

Remark 6 *Note that the first relation (undercut) applies the notion of contrariness on an argument. This is technically wrong since contrariness $^-$ is a function defined for formulas of the language* L *and arguments should not be elements of* L*. In informal discussions with Prakken, he claimed that each defeasible rule has a name which is a formula of* L*. Thus, the notation $\texttt{Conc}(A) \in \bar{C}$ means that the conclusion of argument A is the contrary of the name of the defeasible rule $C_1, \ldots, C_n \Longrightarrow x$. However, the above definition does not capture this intuition since the notion of name of a rule is not defined. Note that even if this intuition is correctly defined, ASPIC+ does not make any difference between formulas of* L *that encode information and formulas that encode names of rules. This may lead to undesirable results as shown on the following example.*

Example 4 *Let $\mathcal{K}_n = \{b\}$, $\mathcal{R}_d = \{b \Rightarrow f\}$ where b stands for "Bird" and f for "Generally, birds fly". Assume that the name of the rule $b \Rightarrow f$ is $\neg f$ (nothing in the definitions forbids this assumption). Note also that in the formal definition, it is not specified in which set $\mathcal{K}_j$ ($j \in \{n, p, i, a\}$) the names of rules should we stored. Assume that these names are not stored in the four bases. Thus, we get two arguments:*

A: b
B: $b, b \Rightarrow f$

The argument B is self-defeating (it undercuts itself). Thus, this system has one preferred extension $\{A\}$, and it returns a unique output which is b. The system is unable to conclude that this bird flies.

Let us now assume that the name of the rule is a certain information that is stored in $\mathcal{K}_n$, thus $\mathcal{K}_n = \{b, \neg f\}$. Note that this set contains two information of different nature. In this case, there is an additional argument C:

C: $\neg f$

It can be checked that C rebuts B. The system has one preferred extension $\{A, C\}$ and returns the set $\{b, \neg f\}$ as its output. This latter is certainly not intuitive.

Remark 7 *Besides, in [10], the author claims that the three attack relations are syntactic and do not reflect any preference between arguments. However, in the definition of rebut, it is clear that an argument whose top rule is strict cannot be attacked by an argument with a defeasible top rule. Thus, the former is preferred to the latter. Moreover, this preference is not intuitive as shown by the following example.*

Example 5 *Assume an argumentation system with the two following arguments:*
$A :\Rightarrow p, p \Rightarrow q, q \Rightarrow r, r \rightarrow x$
$B :\rightarrow d, d \rightarrow e, e \rightarrow f, f \Rightarrow \neg x.$
According to the above definition, A rebuts B but B does not rebut A. Thus, A is assumed stronger than B. This is not intuitive since there is a lot of uncertainty on r. Works on non-monotonic reasoning would rather prefer B to A since defeasible rules violate transitivity and thus, p may not be r.

As in any preference-based argumentation system, in [10] preferences between arguments are used in order to decide which attacks result in defeats.

Definition 7 (Defeat). *Let A and B be two arguments.*

- A successfully rebuts B *if A rebuts B on C and either A contrary-rebuts C or not($C \succ A$).*
- A successfully undermines B *if A undermines B on x and either A contrary-undermines B or not($x \succ A$).*
- A defeats B *iff no premise of A is an issue and A undercuts or successfully rebuts or successfully undermines B.*

Remark 8 *Note that the previous definition uses preferences only when the attack relation is symmetric. While this avoids the problem of conflicting extensions described in [2], it greatly restricts the use of preferences.*

The instantiation of Dung's system is thus the pair $(\mathcal{A}_s, \text{ defeat})$ where $\mathcal{A}_s$ is the set of all arguments built using Definition 4 from $(\mathcal{K}, \texttt{Cl}(\mathcal{R}_s), \mathcal{R}_d)$. $\texttt{Cl}(\mathcal{R}_s)$ is the closure of strict rules under contraposition, that is $\texttt{Cl}(\mathcal{R}_s)$ contains all strict rules and all their contrapositions. A contraposition of a strict rule $x_1, \ldots, x_n \rightarrow x$ is, for instance, the strict rule $x_2, \ldots, x_n, \bar{x} \rightarrow \bar{x_1}$, where $\bar{x}$ and x are contradictory. Recall that in [5], contraposition was proposed for ensuring the rationality postulate on consistency.

Dung's acceptability semantics are applied for evaluating the arguments.

Definition 8 (Acceptability semantics). *Let $\mathcal{B} \subseteq \mathcal{A}_s$, $A \in \mathcal{A}_s$. $\mathcal{B}$ is* conflict-free *iff $\nexists A, B \in \mathcal{B}$ s.t. A defeats B. $\mathcal{B}$* defends *A iff $\forall B \in \mathcal{A}_s$ if B defeats A, then $\exists C \in \mathcal{B}$ s.t. C defeats B. $\mathcal{B}$ is* admissible *iff it is conflict-free and defends all its elements. $\mathcal{B}$ is a* preferred *extension iff it is a maximal (for set inclusion) admissible set.*

In [10], it was shown under several conditions, that ASPIC+ satisfies the rationality postulates defined in [5], namely closure under strict rules and direct and indirect consistency. Unfortunately, due to the described limits of this system, counter-examples can be provided. For instance, the following example shows that an instance of ASPIC+ that violate closure and indirect consistency.

Example 6 *Let* $\mathcal{R}_d = \{\Rightarrow a, \Rightarrow b, \Rightarrow x, \Rightarrow z, a \Rightarrow (x \rightarrow y), b \Rightarrow (z \rightarrow \neg y)\}$*, and let all the other sets be empty and contrariness stand for classical negation. The following arguments can be built:*

A: $\Rightarrow a$
B: $\Rightarrow b$
C: $\Rightarrow x$
D: $\Rightarrow z$
E: $\Rightarrow a, a \Rightarrow (x \rightarrow y)$
F: $\Rightarrow b, b \Rightarrow (z \rightarrow \neg y)$

It can be checked that this system has one preferred extension: $\{A, B, C, D, E, F\}$ *which is not closed under strict rules. Note that in this example strict rules are derived from defeasible ones. Moreover,* $a, b, x, z, x \rightarrow y, z \rightarrow \neg y$ *are outputs of the system. It can be checked that it is indirectly inconsistent.*

Another source of problems for ASPIC$^+$ is the abstract aspect of the language L and its poor notion of consistency. The following example shows that some instantiations may lead to undesirable results, namely inconsistent ones.

Example 7 *Assume that* L *contains propositional formulas, thus contrariness encodes classical negation. Assume also that all the sets are empty except* $\mathcal{R}_d = \{\Rightarrow x, \Rightarrow \neg x \vee y, \Rightarrow \neg y\}$*. Here the rules encode information. Only the following three arguments can be built:* $A_1 :\Rightarrow x$*,* $A_2 :\Rightarrow \neg x \vee y$*,* $A_3 :\Rightarrow \neg y$*. Note that the three arguments are not attacking each other. Thus, the set* $\{A_1, A_2, A_3\}$ *is an admissible/preferred set. This set supports three conclusions which are inconsistent (according to the semantics given to contrariness and to formulas in* L*). Note that this example would return this undesirable result even if the definition of consistency was more general than the one given in Definition 3.*

The previous example reveals another problem with ASPIC$^+$ system. Its result is not closed under classical logic. Indeed, the three conclusions x, $\neg x \vee y$, and $\neg y$ are inferred while y which follows from the two first ones is not deduced. In works on non-monotonic reasoning, namely the seminal paper [8], it is argued that one should accept as plausible consequences all that is logically implied by other plausible consequences. This property is known as *right weakining*.

3 Tarski's monotonic logics

Before studying the underlying logic of ASPIC+, let us first recall the abstract logic $(\mathcal{L}, \mathtt{CN})$ as defined by Tarski [11]. While there is no requirement on the language $\mathcal{L}$, the consequence operator CN should satisfy the following basic axioms.

1. $X \subseteq \texttt{CN}(X)$ (Expansion)
2. $\texttt{CN}(\texttt{CN}(X)) = \texttt{CN}(X)$ (Idempotence)
3. $\texttt{CN}(X) = \bigcup_{Y \subseteq_f X} \texttt{CN}(Y)$[2] (Finiteness)
4. $\texttt{CN}(\{x\}) = \mathcal{L}$ for some $x \in \texttt{L}$ (Absurdity)
5. $\texttt{CN}(\emptyset) \neq \mathcal{L}$ (Coherence)

Once $(\mathcal{L}, \texttt{CN})$ is fixed, a notion of *consistency* arises as follows:

Definition 9 (Consistency). *Let* $X \subseteq \mathcal{L}$. X *is* consistent *w.r.t. the logic* $(\texttt{L}, \texttt{CN})$ *iff* $\texttt{CN}(X) \neq \mathcal{L}$. *It is* inconsistent *otherwise.*

Almost all well-known monotonic logics (classical logic, intuitionistic logic, modal logic, etc.) are special cases of the above notion of an abstract logic. The following logic for representing the color and the size of objects is another Tarskian logic.

Example 8 *Let* $\mathcal{L} = \mathcal{L}_{col} \cup \mathcal{L}_{size} \cup \mathcal{L}_{err}$ *with* $\mathcal{L}_{col} = \{$*white, yellow, red, orange, blue, black*$\}$, $\mathcal{L}_{size} = \{$*tiny, small, big, huge*$\}$, *and* $\mathcal{L}_{err} = \{\perp\}$. *The consequence operator captures the fact that if two different colors or two different sizes are present in the description of an object, then information concerning that object is inconsistent. We define* CN *as follows: for all* $X \subseteq \mathcal{L}$,

$$\texttt{CN}(X) = \begin{cases} \mathcal{L} & \textit{if } (\exists x, y \in X \ s.t. \ x \neq y \\ & \quad \textit{and } (\{x,y\} \subseteq \mathcal{L}_{col} \textit{ or } \{x,y\} \subseteq \mathcal{L}_{size})) \\ & \textit{or if } (\perp \in X) \\ X & \textit{else} \end{cases}$$

For example, $\texttt{CN}(\emptyset) = \emptyset$, $\texttt{CN}(\{red, big\}) = \{red, big\}$, $\texttt{CN}(\{red, blue, big\}) = \texttt{CN}(\{\perp\}) = \mathcal{L}$. *The set* $\{$*red, big*$\}$ *is consistent, while* $\{$*red, blue, big*$\}$ *is inconsistent. Note that this logic does not need any connector of negation.*

4 Rules as object level language

As said before, the (strict and defeasible) rules in ASPIC+ may either encode information or reasoning patterns. In this section, we investigate the first case. Our aim is to study the properties of the corresponding logic, denoted by $(\mathcal{L}_o, \texttt{CN}_o)$.

The language $\mathcal{L}_o$ is composed of the logical formulas of the set L (in [10]). Note that in [10] no particular requirement is made on L, neither on the kind of connectors that are used nor on the way of defining the formulas. However, as said before, L is equipped with the contrariness function $\bar{\ }$. The language $\mathcal{L}_o$ contains also two kinds of information: strict rules (elements of $\mathcal{R}_s$) encoding *certain knowledge* like 'penguins do not fly' and defeasible rules (elements of $\mathcal{R}_d$) encoding defeasible information like 'generally birds fly'. Note that in ASPIC system [1], the same language is considered with the difference that L contains only literals. Thus, $\mathcal{L}_o = \texttt{L} \cup \mathcal{R}_s \cup \mathcal{R}_d$ with $\texttt{L} \cap (\mathcal{R}_s \cup \mathcal{R}_d) = \emptyset$.

An important question now is what are the contents of the bases $\mathcal{K}_n$, $\mathcal{K}_p$, $\mathcal{K}_a$ and $\mathcal{K}_i$ in this case. Since $\mathcal{K}_n$ contains axioms, or undefeasible information, this may be

[2] $Y \subseteq_f X$ means that Y is a finite subset of X.

represented by strict rules since these latter encode certain information. Similarly, since ordinary premises are defeasible, then they should be represented by defeasible rules. Otherwise, the language would be redundant and ambiguous. In sum, $\mathcal{K} = \mathcal{K}_a \cup \mathcal{K}_i$.

When strict and defeasible rules encode knowledge, the consequence operator of the logic used in ASPIC$^+$ is not specified. The only indication can be found in Definition 4 of the notion of argument. A *possible* $\mathtt{CN}_o$ would be the following:

Definition 10 (Consequence operator). *$\mathtt{CN}_o$ is a function from $2^{\mathcal{L}_o}$ to $2^{\mathcal{L}_o}$ s.t. for all $X \subseteq \mathcal{L}_o$, $x \in \mathtt{CN}_o(X)$ iff there exists a sequence $x_1, \ldots, x_n$ s.t.*

1. *x is x_n, and*
2. *for each $x_i \in \{x_1, \ldots, x_n\}$,*
   - *$\exists y_1, \ldots, y_j \rightarrow x_i \in X \cap \mathcal{R}_s$ (resp. $\exists y_1, \ldots, y_j \Rightarrow x_i \in X \cap \mathcal{R}_d$) s.t. $\{y_1, \ldots, y_j\} \subseteq \{x_1, \ldots, x_{i-1}\}$,* or
   - *$x_i \in X \cap \mathtt{L}$*

Example 9 *Let $X = \{x, x \rightarrow y, t \Rightarrow z\}$, $\mathtt{CN}_o(X) = \{x, y\}$.*

Property 1 *Let $X \subseteq \mathcal{L}_o$.*

- *$\mathtt{CN}_o(X) \subseteq \mathtt{L}$*
- *$\mathtt{CN}_o(\emptyset) = \mathtt{cl}(\mathcal{R}_s \cup \mathcal{R}_d)$ where $\mathtt{cl}(\mathcal{R}_s \cup \mathcal{R}_d)$ is the smallest set such that:*
  - *if $\rightarrow x \in \mathcal{R}_s$ (resp. $\Rightarrow x \in \mathcal{R}_d$), then $x \in \mathtt{cl}(\mathcal{R}_s \cup \mathcal{R}_d)$*
  - *if $x_1, \ldots, x_n \rightarrow x \in \mathcal{R}_s$ (resp. $x_1, \ldots, x_n \Rightarrow x \in \mathcal{R}_d$) and $\{x_1, \ldots, x_n\} \subseteq \mathtt{cl}(\mathcal{R}_s \cup \mathcal{R}_d)$, then $x \in \mathtt{cl}(\mathcal{R}_s \cup \mathcal{R}_d)$*

Example 10 *Let $X = \{x, z \rightarrow y, \rightarrow z\}$, $\mathtt{CN}_o(\emptyset) = \{z, y\}$.*

Now that the logic $(\mathcal{L}_o, \mathtt{CN}_o)$ is defined, let us see whether it is a Tarskian one.

Proposition 1. *Let $(\mathcal{L}_o, \mathtt{CN}_o)$ be as defined above. $\mathtt{CN}_o$ is monotonic, satisfies idempotence, and finiteness axioms.*

The next result shows that $\mathtt{CN}_o$ violates expansion and absurdity axioms.

Proposition 2. *Let $(\mathcal{L}_o, \mathtt{CN}_o)$ be as defined above.*

- *For all $X \subseteq \mathcal{L}_o$ s.t. either $X \cap \mathcal{R}_s \neq \emptyset$ or $X \cap \mathcal{R}_d \neq \emptyset$, it holds that $X \not\subseteq \mathtt{CN}_o(X)$.*
- *There is no $x \in \mathcal{L}_o$ s.t. $\mathtt{CN}_o(\{x\}) = \mathcal{L}_o$.*

The previous result shows that the logic $(\mathcal{L}_o, \mathtt{CN}_o)$ is not a Tarskian one since $\mathtt{CN}_o$ violates the key axioms proposed in [11]. Moreover, the notion of consistency given in Definition 3 is weaker than that proposed in [11]. According to Tarski, a set $X \subseteq \mathcal{L}$ is consistent iff $\mathtt{CN}(X) \neq \mathcal{L}$. Thus, this notion captures not only binary minimal conflicts (as with Definition 3), but also ternary or more ones.

5 Rules as reasoning patterns

In the previous section, we have seen how strict and defeasible rules are used for encoding certain and defeasible information. The second way of using these rules is as *inference* rules. In [10], it is argued that strict rules may represent classical reasoning patterns, like modus ponens whereas defeasible rules may capture argument schemes. In this section, we study the resulting logic denoted by $(\mathcal{L}_i, \texttt{CN}_i)$.

Let us start by defining the logical language $\mathcal{L}_i$. In this case, it is exactly the set $\texttt{L}$, that is $\mathcal{L}_i = \texttt{L}$. Thus, the only requirement on $\mathcal{L}_i$ is that it has a contrariness function $\bar{\ }$. It is worth mentioning that the distinction made in [10] between the four bases $\mathcal{K}_n, \mathcal{K}_p, \mathcal{K}_i, \mathcal{K}_a$ is meaningful. Thus, arguments are built from these bases.

Let us now define the consequence operator $\texttt{CN}_i$. It is similar to $\texttt{CN}_o$, except that strict and defeasible rules express inference schemas.

Definition 11 (Consequence operator). *$\texttt{CN}_i$ is a function from $2^{\mathcal{L}_i}$ to $2^{\mathcal{L}_i}$ s.t. for all $X \subseteq \mathcal{L}_i$, $x \in \texttt{CN}_i(X)$ iff there exists a sequence $x_1, \ldots, x_n$ s.t.*

1. *x is x_n, and*
2. *for each $x_i \in \{x_1, \ldots, x_n\}$,*
   - *$\exists y_1, \ldots, y_j \to x_i \in \mathcal{R}_s$ (resp. $\exists y_1, \ldots, y_j \Rightarrow x_i \in \mathcal{R}_d$) s.t. $\{y_1, \ldots, y_j\} \subseteq \{x_1, \ldots, x_{i-1}\}$,* or
   - *$x_i \in X$*

Proposition 3. *The logic $(\mathcal{L}_i, \texttt{CN}_i)$ is monotonic. It satisfies expansion, idempotence, and finiteness.*

Property 2
- *$\texttt{CN}_i(\emptyset) = \texttt{cl}(\mathcal{R}_s \cup \mathcal{R}_d)$ where $\texttt{cl}(\mathcal{R}_s \cup \mathcal{R}_d)$ is the smallest set such that:*
  - *if $\to x \in \mathcal{R}_s$ (resp. $\Rightarrow x \in \mathcal{R}_d$), then $x \in \texttt{cl}(\mathcal{R}_s \cup \mathcal{R}_d)$*
  - *if $x_1, \ldots, x_n \to x \in \mathcal{R}_s$ (resp. $x_1, \ldots, x_n \Rightarrow x \in \mathcal{R}_d$) and $\{x_1, \ldots, x_n\} \subseteq \texttt{cl}(\mathcal{R}_s \cup \mathcal{R}_d)$, then $x \in \texttt{cl}(\mathcal{R}_s \cup \mathcal{R}_d)$*
- *$\texttt{CN}_i(\emptyset) = \emptyset$ iff $\nexists \to x \in \mathcal{R}_s$ and $\nexists \Rightarrow x \in \mathcal{R}_d$ for any $x \in \texttt{L}$*

Example 11 *Let $\mathcal{R}_s = \{x, y \to z; \to x\}$ and $\mathcal{R}_d = \{\Rightarrow y\}$. $\texttt{cl}(\mathcal{R}_s \cup \mathcal{R}_d) = \{x, y, z\}$.*

The previous property shows that the coherence axiom of Tarski may be violated by $\texttt{CN}_i$. It is namely the case when $\texttt{cl}(\mathcal{R}_s \cup \mathcal{R}_d) = \texttt{L}$. $\texttt{CN}_i$ does not guarantee neither the absurdity axiom. Indeed, there is no $x \in \mathcal{L}_i$ such that $\texttt{CN}_i(\{x\}) = \mathcal{L}_i$. In case strict rules encode propositional logic, then such formula exists. However, we can build other logics which do not offer such possibility. Let us consider the following logic which expresses the links between the six symbols of comparisons described in Example 1.

Example 1 (Continued): Assume that $\mathcal{L}_i = \{>, \geq, <, \leq, =, \neq, \geq \wedge \leq, > \vee <\}$, $\mathcal{R}_d = \emptyset$, and $\mathcal{R}_s = \{> \to \geq, < \to \leq, = \to \geq \wedge \leq, \neq \to > \vee <\}$. Note that there is no element in $\mathcal{L}_i$ that has the whole set $\mathcal{L}_i$ as a set of consequences.

As a consequence, the logic $(\mathcal{L}_i, \texttt{CN}_i)$ is not a Tarskian one since it violates the coherence and absurdity axioms. Note that this result holds even when $\texttt{CN}_i$ encodes exactly the classical inference $\vdash$. The reason in this case is due to the poor definition of consistency. As shown is Example 3, in propositional logic the set $\{x, x \to y, \neg y\}$ is inconsistent while it is consistent according to Definition 3.

It is also worth mentioning that there is another family of Tarskian logics that cannot be captured by the monotonic logic $(\mathcal{L}_i, \texttt{CN}_i)$. It is the family of logics whose language does not allow negation or contrariness like the one given in Example 8.

6 Conclusion

This paper investigated ASPIC+ argumentation system. It shows that this system suffers from the following drawbacks: i) The logical formalism is ill-defined, ii) The system may thus lead to undesirable results, iii) The system is grounded on several assumptions which may appear either counter-intuitive like the one on rebut, or restrictive like the one on the preference relation between arguments. iv) The system may violate the rationality postulates on closure and indirect consistency, v) The system returns results which may not be closed under classical logic. Thus, it violates the right weakening axiom [8]. vi) The system may not support intuitive instantiations due to its poor notion of consistency. Contrarily to what is claimed in [10, 9], the underlying logics of ASPIC+ cannot encode the Tarskian ones: neither the ones which make use of a notion of negation nor the ones which do not.

References

1. L. Amgoud, M. Caminada, C. Cayrol, MC. Lagasquie, and H. Prakken. Towards a consensual formal model: inference part. In *Deliverable D2.2 of ASPIC project*, 2004.
2. L. Amgoud and S. Vesic. A new approach for preference-based argumentation frameworks. *Ann. Math. Artif. Intell.*, 63(2):149–183, 2011.
3. T. J. M. Bench-Capon. Persuasion in practical argument using value-based argumentation frameworks. *J. of Logic and Computation*, 13(3):429–448, 2003.
4. J.-Y. Béziau. New light on the square of oppositions and its nameless corner. *Logical Investigations*, 10:218–233, 2003.
5. M. Caminada and L. Amgoud. On the evaluation of argumentation formalisms. *AIJ*, 171 (5-6):286–310, 2007.
6. P. Dung. On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games. *AIJ*, 77:321–357, 1995.
7. A. Hunter. Base logics in argumentation. In *Proceedings of COMMA'10*, pages 275–286.
8. S. Kraus, S. Lehmann, and D. Magidor. Nonmonotonic reasoning, preferential models and cumulative logics. *AIJ*, 44:167–207, 1990.
9. S. Modgil and H. Prakken. Revisiting preferences and argumentation. In *Proceedings of IJCAI'11*, pages 1021–1026.
10. H. Prakken. An abstract framework for argumentation with structured arguments. *Argument and Computation*, pages 1–31, 2010.
11. A. Tarski. *Logic, Semantics, Metamathematics (E. H. Woodger, editor))*, chapter On Some Fundamental Concepts of Metamathematics. Oxford Uni. Press, 1956.