# Conflict Resolution in Assumption-Based Frameworks

Martin Baláž[1], Jozef Frtús[1], Giorgos Flouris[2], Martin Homola[1], and Ján Šefránek[1]

[1] Comenius University in Bratislava, Slovakia
[2] FORTH-ICS, Greece

**Abstract.** We show how defeasible reasoning can be embedded into ABF. Differently from other proposals, we do not encode the conflict resolution mechanism for defeasible rules into the ABF's deductive systems. Instead, we formalize the notions of conflict and conflict resolution and make them part of the extended ABF framework (XABF). This improves the control over the conflict resolution process, and allows to devise and compare different domain-dependent conflict resolution strategies. We also show, that no matter which conflict resolution strategy is used, our framework is able to guarantee certain desired properties.

## 1 Introduction

A number of logics from Non-Monotonic Reasoning (NMR) were investigated and tried for agent reasoning capabilities in logic-based multi-agent systems, especially to deal with incompleteness, uncertainty, inconsistency, and other practical aspects of knowledge. Many of these formalisms can be captured in the Assumption-Based Framework (ABF) [3,6]. ABF is particularly well suited for capturing default reasoning, which is useful to deal with incomplete knowledge. In this case some of the statements are *assumptions* (i.e., hypothetical statements) which can be possibly uprooted if a contrary statement is strictly derived. Such conflicts, called *undermining*, are naturally captured and resolved in ABF (by dropping the assumption and upholding the strictly derived statement).

Defeasible reasoning, which is also often associated with ABF, is useful to deal with inconsistent knowledge. In this case conflicts arise between two conflicting statements that are supported by two different derivations. Such conflicts, called *rebutting*, may be resolved by tracing the respective derivations and finding a derivation rule or an assumption that is to be dropped. However, there is often more than one possibility to do this, therefore multiple conflict resolution strategies (CRS) may be applied, sometimes depending on the application domain. Two such strategies are known as last-link principle and weakest-link principle [9].

Defeasible reasoning can also be embedded into ABF, however, since rebutting conflicts are not resolved directly by the ABF semantics, workarounds are often used, e.g., to use names of rules as new assumptions and to encode the conflict resolution strategy using the ABF's deductive system [7,2]. One of the advantages of ABF is its employment of argumentation semantics which allows not only to compute the solution, but also to provide intuitive explanations why necessary assumptions had to be dropped. As we show in Sect. 2, by relying upon workarounds as cited above such explanations are no longer easily found.

We propose the eXtended Assumption-Based Framework (XABF) in which CRS need not to be encoded into the deductive system; it is elevated to a first-class citizen of the formalism. Our main results are summarized as follows:

- CRS is a parameter of the framework, a number of different CRSs may be defined and used according to the specific needs of the application domain.
- Treatment of the CRS in XABF enables to identify the reasons why conflicts were resolved in any particular way much more clearly, in comparison to encoding a CRS into the deductive system.
- Consistency and closure [4], two widely accepted desiderata for defeasible reasoning, are satisfied in our approach in general, for any given CRS.
- Unlike the ASPIC$^+$ framework [9,8] we avoid using transposed rules which cause problems when embedding formalisms with strictly directional rules whose meaning is affected by transposition (see Sect. 7 for details).

## 2 Motivating Example

To motivate our approach, we will use a running example borrowed from [4]:

*Example 1 (Marriage example).* Consider the following set of rules:

$$\begin{array}{l|rl|l} \to \mathit{wears\_ring} & r_1: & \mathit{wears\_ring} \Rightarrow \mathit{married} & \mathit{married} \to \mathit{has\_wife} \\ \to \mathit{goes\_out} & r_2: & \mathit{goes\_out} \Rightarrow \mathit{bachelor} & \mathit{bachelor} \to {\sim}\, \mathit{has\_wife} \end{array}$$

The rules in the above example lead us to conclude conflicting knowledge that a man wearing a ring is married, and therefore has a wife, whereas a man that goes out is a bachelor and therefore does not have a wife. Note that some of the rules are strict ($\to$) whereas others are defeasible ($\Rightarrow$), and that defeasible rules are associated with a name of the form $r_i$.

To obtain an argumentative semantics for the program, it is not sufficient to direct embed ABF into Dung's abstract argumentation framework [5]. The following arguments are respective to the program above:

$$\begin{array}{lll} A_1 = [\to \mathit{wears\_ring}] & A_3 = [A_1 \Rightarrow \mathit{married}] & A_5 = [A_3 \to \mathit{has\_wife}] \\ A_2 = [\to \mathit{goes\_out}] & A_4 = [A_2 \Rightarrow \mathit{bachelor}] & A_6 = [A_4 \to {\sim}\, \mathit{has\_wife}] \end{array}$$

We can see that $A_5$ and $A_6$ are the only conflicting arguments since they have contradictory conclusions. However, defeating only one of them does not resolve the conflict. Since the remaining arguments are not conflicting, we should accept each literal in $\{\mathit{wears\_ring}, \mathit{goes\_out}, \mathit{married}, \mathit{bachelor}\}$. Futhermore, since strict rules have to be always satisfied, we should also believe in $\mathit{has\_wife}$ and ${\sim}\, \mathit{has\_wife}$ and the original conflict reappears. The problem is that the conflict has to be resolved by defeating one of the arguments $A_3$ or $A_4$ which are not directly involved in the conflict.

There exist other approaches [7,2] which encode conflict resolutions into deductive system. All rules are treated as strict, and for each defeasible rule $r_i$ an additional literal of the form $r_i$ is added. The meaning of $r_i$ is "$r_i$ is defeated", and the meaning of ${\sim}\, r_i$ is "$r_i$ is undefeated". The encoding of the program is as follows:

$$\begin{array}{r|r}
\rightarrow \mathit{wears\_ring} & \mathit{wears\_ring}, \sim r_1 \rightarrow \mathit{married} \\
\rightarrow \mathit{goes\_out} & \mathit{goes\_out}, \sim r_2 \rightarrow \mathit{bachelor} \\
\mathit{married} \rightarrow \mathit{has\_wife} & \sim r_1 \rightarrow r_2 \\
\mathit{bachelor} \rightarrow \sim \mathit{has\_wife} & \sim r_2 \rightarrow r_1
\end{array}$$

Now, application of each formerly defeasible rule $r_i$ is guarded by the new assumption $\sim r_i$. The rules $\sim r_1 \rightarrow r_2$ and $\sim r_2 \rightarrow r_1$ serve as implementation of the conflict resolution strategy: in order to resolve the conflict between possible derivation of $\mathit{has\_wife}$ and its contrary $\sim \mathit{has\_wife}$ only one of $\sim r_1$ may hold while the other must be defeated.

However, the link between $\mathit{has\_wife}$ and $\sim r_1$ is hard to see from the two rules $\sim r_1 \rightarrow r_2$ and $\sim r_2 \rightarrow r_1$. Therefore it is not straightforward to obtain an explanation.

## 3 Preliminaries

A *language* is a set $\mathcal{L}$ of well-formed sentences. An *inference rule* over a language $\mathcal{L}$ is an expression $r$ of the form $\varphi_1, \ldots, \varphi_n \rightarrow \varphi_0$ where $0 \leq n$ and each $\varphi_i$, $0 \leq i \leq n$, is a sentence in $\mathcal{L}$. The sentences $\mathit{prem}(r) = \{\varphi_1, \ldots, \varphi_n\}$ are called the *premises* of $r$ and the sentence $\mathit{cons}(r) = \varphi_0$ is called the *consequence* of $r$. A *deductive system* is a pair $(\mathcal{L}, \mathcal{R})$ where $\mathcal{L}$ is a language and $\mathcal{R}$ is a set of inference rules over $\mathcal{L}$.

A *default derivation for* a sentence $\varphi \in \mathcal{L}$ is an expression of the form $D = [\varphi]$. The consequence of $D$ is the sentence $\mathit{cons}(D) = \varphi$. A default derivation has only one premise $\mathit{prem}(D) = \{\varphi\}$ and only one subderivation $\mathit{subderiv}(D) = \{D\}$. A *deductive derivation for* a sentence $\varphi$ is defined as an expression $D = [D_1, \ldots, D_n \rightarrow \varphi]$ where $D_i$ is a default or deductive derivation for $\varphi_i$, $0 < i \leq n$, and $\varphi_1, \ldots, \varphi_n \rightarrow \varphi$ is an inference rule in $\mathcal{R}$. The consequence of $D$ is the sentence $\mathit{cons}(D) = \varphi$, the premises of $D$ are the sentences $\mathit{prem}(D) = \mathit{prem}(D_1) \cup \cdots \cup \mathit{prem}(D_n)$, and the subderivations of $D$ are $\mathit{subderiv}(D) = \{D\} \cup \mathit{subderiv}(D_1) \cup \cdots \cup \mathit{subderiv}(D_n)$. A *derivation for* a sentence $\varphi$ is a default or deductive derivation for $\varphi$. We will say that derivation $D'$ is a *subderivation* of $D$ (denoted by $D' \sqsubseteq D$) iff $D' \in \mathit{subderiv}(D)$; similarly, we will say that derivation $D'$ is a *proper subderivation* of $D$ (denoted by $D' \sqsubset D$) iff $D' \sqsubseteq D$ and $D' \neq D$. A *theory* is a set $S$ of sentences. A sentence $\varphi$ is a *consequence* of a theory $S$ iff there exists a derivation $D$ for $\varphi$ such that $\mathit{prem}(D) \subseteq S$. By $\mathit{Cn}_{\mathcal{R}}(S)$ we will denote the set of all consequences of $S$.

An *assumption-based framework* is a tuple $\mathcal{F} = (\mathcal{L}, \mathcal{R}, \mathcal{A}, \overline{\phantom{x}})$ where $(\mathcal{L}, \mathcal{R})$ is a deductive system, $\mathcal{A} \subseteq \mathcal{L}$ is a set of *assumptions*, and $\overline{\phantom{x}} \colon \mathcal{A} \mapsto \mathcal{L}$ is a mapping called *contrariness function*. We say that the sentence $\overline{\alpha}$ is the *contrary* of an assumption $\alpha$. A *context* is a set $\Delta$ of assumptions. We say that $\Delta$ is *conflict-free* iff $\{\alpha, \overline{\alpha}\} \nsubseteq \mathit{Cn}_{\mathcal{R}}(\Delta)$ for each assumption $\alpha$; and that $\Delta$ is *closed* iff $\mathit{Cn}_{\mathcal{R}}(\Delta) \cap \mathcal{A} \subseteq \Delta$. An assumption-based framework is *flat* iff each context is closed. A context $\Delta$ *attacks* an assumption $\alpha$ iff $\overline{\alpha} \in \mathit{Cn}_{\mathcal{R}}(\Delta)$. A context $\Delta$ *defends* an assumption $\alpha$ iff each closed context attacking $\alpha$ contains an assumption attacked by $\Delta$. A closed context $\Delta$ is *attack-free* iff $\Delta$ does not attack an assumption in $\Delta$. An attack-free context $\Delta$ is *admissible* iff $\Delta$ defends each assumption in $\Delta$. A closed context $\Delta$ is *complete* iff $\Delta$ is admissible and contains all assumptions defended by $\Delta$; *grounded*[3] iff $\Delta$ is a subset-maximal admissible context contained in all complete contexts; *preferred* iff $\Delta$ is a subset-maximal admissible context; *ideal* iff $\Delta$ is a subset-maximal admissible context contained in all preferred contexts; *stable* iff $\Delta$ is an attack-free context attacking each assumption which does not belong to $\Delta$.

## 4 Conflict Resolution Strategies

Conflict resolution in standard ABF is performed using sets of assumptions (i.e., contexts). Other formalisms, like ASPIC$^+$ [9,8], or the framework proposed in [12] use structures similar to derivations. Here, we propose to use derivations to define conflicts and conflict resolution in ABF. Thus, each different pair of derivations (that allow us to conclude an assumption and its contrary) leads to a different conflict, even if these different conflicts are all generated by the same pair of contexts. Consequently, a conflict resolution deals with one possible "cause" of conflict (where "cause" here means a pair of derivations). This approach allows a very fine-grained treatment of conflicts and resolutions, as motivated in Section 2, as well as the distinction between different kinds of conflicts (rebutting, undermining). Formally:

**Definition 1 (Conflict).** *We say that derivation $D_1$* is in conflict with *derivation $D_2$ iff there is some $\alpha \in \mathcal{A}$ such that $cons(D_1) = \alpha$ and $cons(D_2) = \overline{\alpha}$. A* conflict *is a pair $(D_1, D_2)$ such that $D_1$ is in conflict with $D_2$.*

Now let's consider two conflicts $(D_1, D_2)$ and $(D'_1, D'_2)$. It is clear that if $D_1 \sqsubseteq D'_1$ and $D_2 \sqsubseteq D'_2$, then by resolving the first conflict, the second is automatically resolved as well. Therefore, it makes sense to resolve "smaller" conflicts first. This leads us to introduce the notion of *subconflict*, as follows:

**Definition 2 (Subconflict).** *Let $(D_1, D_2)$ and $(D'_1, D'_2)$ be conflicts. We say that $(D'_1, D'_2)$ is a* subconflict of *$(D_1, D_2)$ (denoted by $(D'_1, D'_2) \sqsubseteq (D_1, D_2)$) iff $D_1 \sqsubseteq D'_1$ and $D_2 \sqsubseteq D'_2$. We say that $(D_1, D_2)$ is a proper subconflict of $(D'_1, D'_2)$ (denoted by $(D_1, D_2) \sqsubset (D'_1, D'_2)$) iff $(D_1, D_2) \sqsubseteq (D'_1, D'_2)$ and $(D_1, D_2) \neq (D'_1, D'_2)$.*

Conflict resolution can be simply defined as a triple, where the first two elements indicate a conflict, whereas the third indicates the assumption chosen to abandon in order to eliminate that conflict. Formally:

**Definition 3 (Conflict Resolution).** *A* conflict resolution *is a triple $\rho = (D_1, D_2, \alpha)$ such that $D_1$ is in conflict with $D_2$ and $\alpha \in prem(D_1) \cup prem(D_2)$. The contrary of $\alpha$ is called the* resolution *of $\rho$, and denoted by $res(\rho)$. The* context *of a conflict resolution, denoted by $ctx(\rho)$, is the set:*

$$ctx(\rho) = \begin{cases} prem(D_1) \cup prem(D_2) & \textit{whenever } \alpha \in prem(D_1) \cap prem(D_2) \\ (prem(D_1) \cup prem(D_2)) \setminus \{\alpha\} & \textit{otherwise} \end{cases}$$

[3] Grounded context is called well-founded in [3]; we call it grounded to be consistent with [5].

In Definition 3, $res(\rho)$ intuitively refers to the contrary of an assumption that the conflict resolution chose to drop from our set of assumptions, whereas the set $ctx(\rho)$ refers to the assumptions that essentially "cause" the chosen assumption to be dropped. Note that this is true only as far as the specific $\rho$ is concerned; the interplay between different conflict resolutions and the choices they encode need also to be considered, as explained in detail in Section 5.

**Definition 4.** *A* conflict resolution strategy $\sigma$ *is a mapping which assigns to an ABF a set of conflict resolutions of this ABF.*

Intuitively, a conflict resolution strategy takes an ABF and returns a set of conflict resolutions; note that a conflict resolution strategy does not necessarily resolve all conflicts that appear in an ABF, i.e., it may opt to leave some of the conflicts unresolved. In the following example, we explain the notions of conflict, conflict resolution and conflict resolution strategy:

*Example 2 (Marriage example revisited).* Continuing Example 1, we note that the following derivations can be created:

$$D_1 = [[[\to goes\_out], [\sim r_2] \to bachelor] \to \sim has\_wife]$$
$$D_2 = [[[\to wears\_ring], [\sim r_1] \to married] \to has\_wife]$$

We can see that there are two alternatives for resolving the conflict $(D_1, D_2)$. We defeat either $\sim r_2$ or $\sim r_1$. Thus, we can define the following conflict resolutions: $\rho_1 = (D_1, D_2, \sim r_2)$, $\rho_2 = (D_1, D_2, \sim r_1)$. It follows that $ctx(\rho_1) = \{\sim r_1\}$, $res(\rho_1) = r_2$, and $ctx(\rho_2) = \{\sim r_2\}$, $res(\rho_2) = r_1$. We define a conflict resolution strategy that includes both resolutions, i.e., $\sigma(\mathcal{F}) = \{\rho_1, \rho_2\}$.

## 5 Argumentation Semantics

In this section, we will show how we can model conflict resolution and conflict resolution strategies. In the following, by XABF we mean an arbitrary, but fixed, ABF $\mathcal{F}$ and the set of conflict resolutions $\mathcal{P} = \sigma(\mathcal{F})$, for an arbitrary, but fixed, conflict resolution strategy $\sigma$. Each conflict resolution $\rho$ in $\mathcal{P}$ represents a choice as to how a conflict should be resolved; this choice actually determines the assumption to be dropped in order for the conflict to disappear. This essentially implies that certain assumptions invalidate other assumptions (more precisely, the assumptions in $ctx(\rho)$ invalidate $res(\rho)$). This idea can be captured nicely using the notion of "attack" appearing in ABFs [3,6], where a context $\Delta$ "attacking" an assumption $\alpha$ intuitively means that $\Delta \subseteq \mathcal{A}$ would imply that $\alpha \notin \mathcal{A}$. This is of course generalized to all super-contexts of $\Delta$, which attack contexts including $\alpha$.

However, the above viewpoint considers only the effects of a single conflict resolution, not taking into account the interplay between conflict resolutions in the conflict resolution strategy. In effect, the choices made by the different conflict resolutions in the strategy are not independent, because the resolution proposed by a conflict resolution may implicitly resolve other conflicts as well, thereby making another conflict resolution void. As a result, a certain context (that is causing a certain conflict, which is being resolved in a manner prescribed by the chosen strategy), may prevent another conflict from appearing, therefore it may *defend* some assumption (resulting from the corresponding conflict resolution) from attack. This notion of defence can also be described using ABFs, via their inherent notion of defence.

In Definitions 5, 6 we formally define the notions of attack and defence, which depend on the actual conflict resolution strategy considered and are thus an extension of the corresponding notions described in Section 3 and in [3,6].

**Definition 5 (Attack-Freeness).** *A context* $\Delta$ attacks *an assumption* $\alpha$ *iff there exists some* $\rho \in \mathcal{P}$ *with* $ctx(\rho) \subseteq \Delta$ *and* $res(\rho) = \overline{\alpha}$. *We denote:*

$$Attack_{\mathcal{P}}(\Delta) = \{\alpha \in \mathcal{A} \mid \exists \rho \in \mathcal{P}\colon res(\rho) = \overline{\alpha} \wedge ctx(\rho) \subseteq \Delta\}$$

*A context* $\Delta$ *is* attack-free *iff* $\Delta$ *does not attack any assumption in* $\Delta$*, i.e. iff* $Attack_{\mathcal{P}}(\Delta) \cap \Delta = \emptyset$.

**Definition 6 (Admissibility).** *A context* $\Delta$ defends *an assumption* $\alpha$ *iff* $\Delta$ *attacks an assumption in each context attacking* $\alpha$. *We will denote*

$$Defence_{\mathcal{P}}(\Delta) = \{\alpha \in \mathcal{A} \mid \forall \rho \in \mathcal{P}\colon res(\rho) = \overline{\alpha} \Rightarrow ctx(\rho) \cap Attack_{\mathcal{P}}(\Delta) \neq \emptyset\}$$

*An attack-free context* $\Delta$ *is* admissible *iff* $\Delta$ *defends each assumption in* $\Delta$*, i.e. iff* $\Delta \subseteq Defence_{\mathcal{P}}(\Delta)$.

**Definition 7 (Extension).** *A context* $\Delta$ *is*

- complete *iff* $\Delta$ *is admissible and* $Defence_{\mathcal{P}}(\Delta) \subseteq \Delta$
- grounded *iff* $\Delta$ *is a subset-maximal admissible context contained in all complete contexts*
- preferred *iff* $\Delta$ *is a subset-maximal admissible context*
- ideal *iff* $\Delta$ *is a subset-maximal admissible context contained in all preferred contexts*
- stable *iff* $\Delta = \mathcal{A} \setminus Attack_{\mathcal{P}}(\Delta)$

*If* $\Delta$ *is a* $\Sigma_{\mathcal{P}}$*-context then* $\mathcal{E} = Cn_{\mathcal{R}}(\Delta)$ *is a* $\Sigma_{\mathcal{P}}$*-extension of* $\mathcal{F}$ *for each* $\Sigma \in \{complete, grounded, preferred, ideal, stable\}$.

In the following, we will use the term *standard semantics* to refer to any of the semantics of extensions that appear in Definition 7. An extension corresponds to a set of assumptions (aka, context) that is acceptable, under the given conflict resolution strategy. This extension is essentially the result of the conflict resolution process, where all different conflict resolutions in the strategy, as well as their interplay, have been considered in selecting what to drop and what to keep. Recall that this process does not guarantee a conflict-free set of assumptions $\mathcal{A}$, as, by design, we allow some conflicts to remain unresolved. An important difference of the above viewpoint compared to standard approaches is that the actual reason for defeating an assumption is not an argument, but the conflicts themselves (and their resolutions), which force us to drop some of the assumptions.

*Example 3 (Marriage example continued).* If we take the ABF $\mathcal{F}$ with the set $\mathcal{P}$ of conflict resolutions from Example 2, we have three complete contexts, namely $\Delta_1 = \{\sim r_1, \sim \mathit{bachelor}\}$, $\Delta_2 = \{\sim r_2, \sim \mathit{married}, \sim \mathit{has\_wife}\}$, and $\Delta_3 = \{\}$. They correspond to three extensions $\mathcal{E}_1 = Cn_{\mathcal{R}}(\Delta_1) = \Delta_1 \cup \{\mathit{married}, \mathit{has\_wife}\}$, $\mathcal{E}_2 = Cn_{\mathcal{R}}(\Delta_2) = \Delta_2 \cup \{\mathit{bachelor}\}$, and $\mathcal{E}_3 = Cn_{\mathcal{R}}(\Delta_3) = \{\}$. In the context $\Delta_1$, the conflict resolution $\rho_1$ explains why the assumption $\sim r_2$ is defeated. Since $ctx(\rho_1) \subseteq \Delta_1$ and $res(\rho_1) = r_2$, the assumption $\sim r_2$ is defeated in order to resolve conflicts between derivations $D_1$ and $D_2$. Similarly, the conflict resolution $\rho_2 \in \mathcal{P}$ is an explanation for defeating $\sim r_1$ in the context $\Delta_2$.

## 6 Properties

In this section, we show the properties of the constructions defined in Section 5. In particular, we show that XABF behaves in a reasonable manner, according to well-established properties present in [4] (Propositions 1, 2, 3). Finally, we show the role of "minimal" subconflicts (see Definitions 9, 10 and Propositions 4, 5).

In the rest of this section, we assume an arbitrary, but fixed, ABF $\mathcal{F}$ and the set of conflict resolutions $\mathcal{P} = \sigma(\mathcal{F})$ for an arbitrary, but fixed, conflict resolution strategy $\sigma$, as well as any given standard semantics $\Sigma_{\mathcal{P}}$.

As already mentioned, a conflict resolution strategy does not need to resolve all conflicts. However, strategies that do resolve all conflicts have some interesting properties and will be called *total*. Formally:

**Definition 8.** *A set of conflict resolutions $\mathcal{P}$ is* total *iff for each context $\Delta$, which is not conflict-free, there is a resolution $\rho$ with $ctx(\rho) \subseteq \Delta$ and $res(\rho) \in \Delta$.*

The following results show that our framework satisfies a generalized version of the rationality conditions proposed in [4]. Note that for Proposition 1, the hypothesis of totality is crucial.

**Proposition 1.** *If $\mathcal{P}$ is total then each $\Sigma_{\mathcal{P}}$-extension is conflict-free.*

**Proposition 2.** *If $\emptyset$ is not conflict-free then each $\Sigma_{\mathcal{P}}$-extension is not conflict-free.*

**Proposition 3.** *Each $\Sigma_{\mathcal{P}}$-extension is closed under $Cn_{\mathcal{R}}$.*

Subconflicts (Definition 2) were introduced to capture the intuition that the resolution of "smaller" conflicts (in the sense of $\sqsubseteq$) also resolves "larger" ones. Thus, resolutions resolving the $\sqsubseteq$-minimal conflicts are of special interest.

**Definition 9.** *The bottom of $\mathcal{P}$ is a set $\lfloor \mathcal{P} \rfloor = \{(D_1, D_2, \alpha) \in \mathcal{P} \mid \forall (D_1', D_2') \sqsubset (D_1, D_2)\colon (D_1', D_2', \alpha) \notin \mathcal{P}\}$.*

**Proposition 4.** *A theory $\mathcal{E}$ is a $\Sigma_{\mathcal{P}}$-extension iff $\mathcal{E}$ is a $\Sigma_{\lfloor \mathcal{P} \rfloor}$-extension.*

Proposition 4 implies that all conflict resolutions that are not in the bottom of the original set can be dropped without changing the semantics. If the set $\mathcal{P}$ of conflict resolution is in addition downward closed then $\sqsubseteq$-minimal conflicts take precedence during resolution. As a special case, undermining takes precedence over rebutting as already suggested by Prakken and Sartor [10]. In our case, the same precedence in conflict resolutions is given also in the case of two $\sqsubseteq$-related rebutting conflicts (i.e., the subconflict of the two should be removed).

**Definition 10.** *We say that a set $\mathcal{P}$ of conflict resolutions is* downward closed *iff for each conflict resolution $(D_1, D_2, \alpha) \in \mathcal{P}$ there exists a minimal subconflict $(D_1', D_2')$ of $(D_1, D_2)$ with $(D_1', D_2', \alpha) \in \mathcal{P}$.*

**Proposition 5.** *The bottom of a downward closed set $\mathcal{P}$ of conflict resolutions contains only $\sqsubseteq$-minimal conflicts.*

An important consequence of Proposition 5 is that, for constructing a conflict resolution strategy, one only needs to be concerned with minimal conflicts. This is a very useful property from the practical viewpoint, as it allows not dealing with all conflicts during the construction of a strategy, only with minimal ones. Note that this intuition cannot be extended to minimal derivations, as they do not always lead to minimal conflicts: a pair of minimal derivations for an assumption could "hide" a subconflict on another assumption.

## 7 Related Work

The formal notion of CRS occurs in previous works [1,2], where the focus was however entirely on DeLP. In the current work, these ideas are largely pushed forward resulting into the extension of ABF as a generic framework with improved capabilities and properties, capable to embed any non-monotonic formalism, not only DeLP.

ASPIC$^+$ [9,8] also enables general purpose defeasible reasoning and it satisfy the both consistency and closure properties, though it relies on the addition of transposed rules. XABF offers more flexible notion of CRS than ASPIC$^+$ and, in addition, it does not rely in transposition of rules which may introduce undesired consequences. For instance rules in logic programming (LP) are directional. Rule $\neg b \rightarrow \neg a$ does not allow to derive anything from the theory $\{a\}$, after the addition of the transposed rule $a \rightarrow b$ we derive $b$ as a consequence of the theory $\{a\}$. This does not allow to directly embed[4] of LP into ASPIC$^+$. In XABF we avoid this problem.

The generalized ABF of Toni [11] also assure closure and consistency. On the other hand, the properties of generalized ABF are not investigated in detail on the abstract level. While sharing several goals with Toni, we proposed a principal extension of ABF with well motivated and useful generalizations, such as flexible CRSs (which also capture domain specific preferences), which are not investigated by Toni.

## 8 Conclusions

We proposed XABF, an extended ABF framework that enables improved treatment of defeasible reasoning via assumption-based argumentation. Lifting the CRS to a first-class citizen in XABF enables multiple CRSs to be formalized and used with XABF.

[4] Direct embedding is such that LP rules will become the rules of the deductive system in ASPIC$^+$. Note that we do not claim that a more complex, indirect embedding cannot be done.

Moreover, it provides a customized choice of the CRS to use, based on the specific needs of the application domain. Furthermore, it is possible to identify the reasons why conflicts were resolved in any particular way and thus to provide for explanations. The semantics we proposed for XABF takes care that the widely accepted properties of consistency and closure [4] are satisfied *for any given CRS*. These properties are important, as they guarantee that conflicts are not resolved just cosmetically and then consequently derived again by the deductive system.

Our approach also allows to compare different CRSs and to study their formal properties. To demonstrate this, we formally characterized a class of CRSs which are minimal in the sense that resolution of superconflicts can be propagated to resolving their minimal subconflicts; that is, in these CRSs all conflicts can be resolved more effectively by considering a smaller number of cases.

### Acknowledgements

This work resulted from the Slovak–Greek bilateral project co-financed by APVV (as SK-GR-0070-11) and GSRT together with the EU (as 12SLO_ET29_1087). The Slovak side further acknowledges support from the VEGA project no. 1/1333/12. Martin Baláž and Martin Homola are also supported by the APVV project no. APVV-0513-10.

## References

1. Baláž, M., Frtús, J., Homola, M.: Conflict resolution in structured argumentation. In: LPAR-19 (Short Papers). EPiC, vol. 26, pp. 23–34. EasyChair (2014)
2. Baláž, M., Frtús, J., Homola, M., Šefránek, J., Flouris, G.: Embedding defeasible logic programs into generalized logic programs. In: WLP. To appear in CEUR-WS. (2014)
3. Bondarenko, A., Dung, P.M., Kowalski, R.A., Toni, F.: An abstract, argumentation-theoretic approach to default reasoning. Artif. Intell. 93(1-2), 63–101 (1997)
4. Caminada, M., Amgoud, L.: On the evaluation of argumentation formalisms. Artif. Intell. 171(5-6), 286–310 (2007)
5. Dung, P.M.: On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games. Artif. Intell. 77(2), 321–357 (1995)
6. Dung, P.M., Mancarella, P., Toni, F.: Computing ideal sceptical argumentation. Artif. Intell. 171(10-15), 642–674 (2007)
7. Kowalski, R.A., Toni, F.: Abstract argumentation. Artif. Intell. Law 4(3-4), 275–296 (1996)
8. Modgil, S., Prakken, H.: Revisiting preferences and argumentation. In: IJCAI (2011)
9. Prakken, H.: An abstract framework for argumentation with structured arguments. Argument & Computation 1(2), 93–124 (2010)
10. Prakken, H., Sartor, G.: Argument-based extended logic programming with defeasible priorities. Journal of Applied Nonclassical Logics 7(1), 25–75 (1997)
11. Toni, F.: Assumption-based argumentation for closed and consistent defeasible reasoning. In: JSAI 2007 (Revised Selected Papers). vol. 4914, pp. 390–402. Springer (2008)
12. Vreeswijk, G.A.W.: Abstract argumentation systems. Artif. Intell. 90(1-2), 225–279 (1997)