Journal of Artificial Intelligence Research 82 (2025) 389–431 Submitted 09/2024; published 01/2025

Forgetting in Abstract Argumentation: Limits and Possibilities

Ringo Baumann BAUMANN@INFORMATIK.UNI-LEIPZIG.DE
Universität Leipzig, Germany
ScaDS.AI, Dresden/Leipzig, Germany

Matti Berthold BERTHOLD@INFORMATIK.UNI-LEIPZIG.DE
ScaDS.AI, Dresden/Leipzig, Germany

Dov Gabbay DOV.GABBAY@KCL.AC.UK
Odinaldo Rodrigues ODINALDO.RODRIGUES@KCL.AC.UK
King's College London, UK

Abstract

The topic of *forgetting*, which loosely speaking means losing, removing, or even hiding some variables, propositions, or formulas, has been extensively studied in the field of knowledge representation and reasoning for many major formalisms. In this article, we convey this topic to the highly active field of abstract argumentation. We provide an in-depth analysis of desirable syntactical and/or semantical properties of possible forgetting operators. In doing so, we included well-known logic programming conditions, such as *strong persistence* or *strong invariance*. Further, we argue that although abstract argumentation and logic programming are closely related, it is not possible to reduce forgetting in abstract argumentation to forgetting in logic programming in a straightforward manner. The analysis of desiderata, adapted to the specifics of abstract argumentation, includes implications among them, individual and collective satisfiability, and identifying inherent limits for a set of prominent semantics. Finally, we conduct a case study on stable semantics incorporating concrete forgetting operators.

1. Introduction

The notion of *forgetting* is central to knowledge representation and reasoning and has been extensively studied in several major formalisms, such as classical logic, logic programming, and belief change (Gonçalves, Knorr, & Leite, 2016a; Delgrande, 2017; Eiter & Kern-Isberner, 2019). Loosely speaking, forgetting is about losing or removing some variables, propositions, or formulas while minimizing the loss of informational content incurred during the process. This natural human ability can be unintentional or deliberate and plays an important role in many cognitive processes. In reasoning, it can be exploited to focus attention by omitting less relevant information and make the process more efficient, but forgetting is also important in query answering, decision making, reasoning about actions, and belief change, amongst others (see (Lang, Liberatore, & Marquis, 2003) for more details).

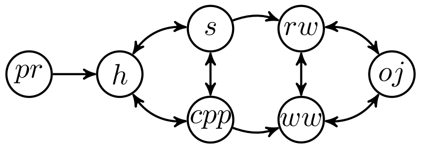
An excellent survey of the literature of forgetting can be found in (Eiter & Kern-Isberner, 2019), where Eiter and Kern-Isberner identify two main interpretations by which it can be accomplished. "Type 1" forgetting involves the removal of atoms from the *language* of a theory, such that it can no longer represent or infer certain concepts (Lin & Reiter, 1994; Lang et al., 2003; Gonçalves et al., 2016a). "Type 2" forgetting is about the removal of information from a theory, making it impossible to *derive* some consequences from it. Notably, a representative of type 2 forgetting in a logical setting is the well-known AGM theory for belief revision and contraction (Alchourrón, Gärdenfors, & Makinson, 1985; Gärdenfors, 1988; Rodrigues, Gabbay, & Russo, 2011). This theory is presented in terms of postulates expressing *desiderata* for the belief change operations. These postulates constrain the relationship between logical theories before and after the operations, stipulating minimal rational requirements for the operations.

Abstract argumentation frameworks (AFs) (Dung, 1995; Baroni, Gabbay, Giacomin, & van der Torre, 2018) were introduced to model important aspects of non-monotonic reasoning and debate, and thus they also play a prominent role in the knowledge representation literature. An AF is a tuple (A, R), where A is a set of (atomic) arguments and $R \subseteq A \times A$ is an attack relation between arguments. Naturally, one may wonder how forgetting can be performed in AFs. A natural approach is to attempt to reuse existing forgetting results from other formalisms. For example, we can apply the results of forgetting in logic programming by translating an AF into a logic program (LP), performing the operation there, and then translating the results back, if possible. Yet, as we shall see, forgetting in abstract argumentation cannot be reduced to forgetting in logic programming due to limits regarding the inversion of the standard translation (Strass, 2013) as well as differences in the expressive power between LPs and AFs (Eiter, Fink, Pührer, Tompits, & Woltran, 2013; Dunne, Dvořák, Linsbichler, & Woltran, 2015). Nevertheless, we draw a lot of inspiration from logic programming and convey many essential properties such as *strong persistence* or *strong invariance* (Knorr & Alferes, 2014; Gonçalves et al., 2016a) to the realm of abstract argumentation.

A purely *syntactical* interpretation of forgetting is often too simplistic as it disregards the acceptability of arguments before and after the operation. Most of what we can say about such a syntactical approach is concerned with what the initial argumentation framework looks like after the operation of forgetting. However, from the *semantical* point of view, we can interpret forgetting as the suppression of the acceptability of one or more arguments. Although this can obviously be achieved by simply syntactically modifying the argumentation graph, for example, by removing unwanted arguments, in more realistic scenarios, simply denying the existence of some argument(s) may not be possible. In such cases, a subtler, more tactical removal is necessary, perhaps even by introducing counter-arguments. All of this may affect the acceptance of other arguments, so in practice we need to consider how an argumentation framework is interpreted under a *specific* argumentation semantics (i.e., its "extensions" under the semantics), whether or not the argument(s) to be forgotten belong to some (or all) of the extensions, and finally the actual *intention* of the operation.

Let us start with a motivating example highlighting different aspects of forgetting.

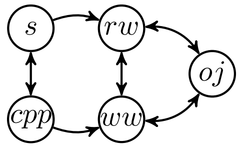
Example 1. *Consider the scenario where Antoine celebrates his birthday and wants to plan the food and drinks to be served at his party. There will be either hare (h), salmon (s), or cheese-and-potato pie (cpp). For drinks, Antoine will offer either red wine (rw), white wine (ww), or orange juice (oj). It goes without saying that he will not serve red wine with salmon, nor white wine with the cheesy option. Antoine's daughter has a pet rabbit (pr) and therefore strongly opposes eating hare. The scenario may be represented by the argumentation framework below.*



Under the stable semantics there are four different culinary outcomes:

$$\mathbb{E}_0 = \{\{pr, s, ww\}, \{pr, s, oj\}, \{pr, cpp, rw\}, \{pr, cpp, oj\}\}$$

Since Antoine's daughter is certainly attending his party, in no scenario will she even consider hares as part of the conversation. The hare option (and the existence of the pet rabbit) are straightforwardly discarded yielding the new argumentation framework below. In other words, these arguments and their respective attacks are syntactically removed (see s_1 *in Section 4), but the relationships between the remaining arguments are preserved.*



We are now left with the four outcomes (stable extensions) below:

$$\mathbb{E}_1 = \{\{s, ww\}, \{s, oj\}, \{cpp, rw\}, \{cpp, oj\}\}$$

There are, however, more complicated considerations to be made, depending on whether some of Antoine's friends are also attending. Benita is currently pregnant and will not drink alcohol. From her point of view, she wants the wine options excluded from the outcomes in $\mathbb{E}_1$ *above (see* e_1 *in Section 4):*

$$\mathbb{E}_2 = \{\{s\}, \{s, oj\}, \{cpp\}, \{cpp, oj\}\}$$

In $\mathbb{E}_2$*, the outcomes of the forgetting operation are the same as in* $\mathbb{E}_1$*, except that we ignore the forgotten arguments (with wine) in each extension. This type of extensional forgetting is not always possible as we will see.*

Another friend of Antoine, who shall remain anonymous, has a rather strict opinion against alcohol. If wine is being served, he will not attend the party at all. This person simply disregards all extensions in $\mathbb{E}_1$ *that include red or white wine. Therefore, there are only two potential scenarios in which he attends the party (see* e_4 *in Section 4):*

$$\mathbb{E}_3 = \{\{s, oj\}, \{cpp, oj\}\}$$

To provide a systematic analysis of these possibilities, our methodology follows the approach outlined in (Alchourrón et al., 1985), stipulating desiderata for the forgetting operation. The main idea is that one can focus on combinations of these desiderata to characterize the desired properties of a specific operation to be defined. To facilitate their use, we then classify our desiderata under specific *aspects* related to, for example, the magnitude of change to the argumentation framework, the relationship between its extensions before and after the operation, the acceptability statuses of the arguments under credulous and skeptical reasoning modes, and so on.

In this process, questions that naturally arise include the relationship between the desiderata and whether specific combinations are realisable under a particular semantics. As it turns out, the satisfiability of certain desiderata imply or prevent the satisfiability of others, and as a result they can be naturally arranged by strength and also checked for compatibility under specific semantics. Our investigation of these issues covered seven well-known argumentation semantics, namely stage, stable, semi-stable, preferred, grounded, ideal and eager semantics.

To summarise, our contributions are as follows:

- We postulate, investigate and classify a total number of 25 syntactical and/or extensional desiderata for potential forgetting operation and establish dependencies between them.

- We then investigate the individual as well as the combined satisfiability of single or promising combinations of desiderata, respectively. This means, we tackle the question whether there are forgetting operators at all for the semantics under consideration.

- We further shed light on the issue simultaneous vs. iterative forgetting. The analysis shows that no clear statement can be made, as sometimes simultaneous and sometimes iterative forgetting yields a more compact output.

- Finally, we provide three concrete constructions regarding stable semantics performing forgetting w.r.t. one of the most central extensional desideratum, namely removing extensions that overlap with forgotten arguments.

In doing so, the article significantly expands upon the preliminary investigations in (Baumann, Gabbay, & Rodrigues, 2020; Baumann & Berthold, 2022) and provides the first comprehensive overview of forgetting in abstract argumentation, establishing the limits and possibilities for applying this operation. We decided to follow the lines of (Baumann & Berthold, 2022) and consider the more general problem of forgetting entire sets of arguments, as forgetting single arguments can be seen as a corner case. Moreover, we have introduced new desiderata, a new construction, and provided declarative forms of the presented operators. Additionally, we offer more detailed explanations, illustrative examples, and discussion of related work.

The rest of the paper is organised as follows: In Section 2, we motivate the investigation through a number of illustrating examples. Section 3 provides the necessary background in argumentation theory as well as logic programming. In Section 4, we present the main desiderata for forgetting operations and establish the key relationships between them. Section 5 investigates the limits of the applicability of these desiderata to the considered semantics by presenting satisfiability results for each. In Section 6, we provide a discussion of the application of forgetting to the central stable semantics. Finally, in Section 7, we conclude with a summary of the main results and outline aspects deserving further investigation.

2. Motivating Examples

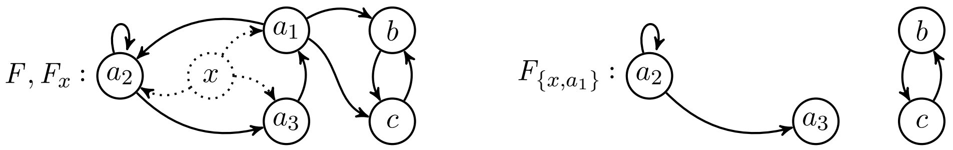
In this section we present a few motivating examples showing some challenges and peculiarities of forgetting in Dungean Frameworks.

2.1 Syntactical Forgetting of Single Arguments

It is quite surprising that the issue of how to forget *single* arguments has not received much attention yet. The most straightfoward way of forgetting an argument is simply by removing it from the argumentation framework (as considered in (Bisquert, Cayrol, de Saint-Cyr, & Lagasquie-Schiex, 2011)). Whilst such a syntactical approach obviously guarantees that the extensions of the resulting AF will not contain the argument, it leaves the precise semantical relationship between the input and output frameworks unclear.

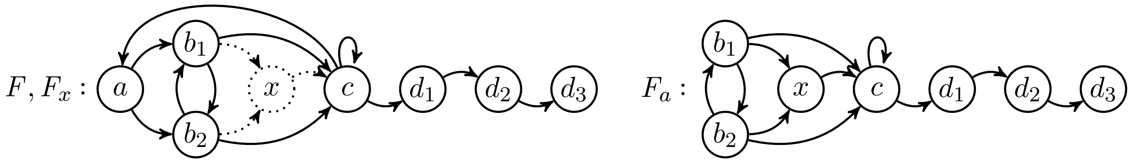
The following examples show some undesired effects of the syntactical removal of single arguments and some possible alternatives. For precise definitions of argumentation semantics and reasoning modes, please refer to Section 3.1. Let $F = (A, R)$ be an AF, x be a single argument and X a set of arguments. For notational convenience we use F_x (respectively, F_X) to denote the AF F restricted to $A \setminus \{x\}$ (respectively, $A \setminus X$). This means, F_x (F_X) is obtained from F by syntactically deleting x (any argument in X) and all edges involving x (arguments in X).

Example 2 (Forget x from Extensions). *Consider the AF F below, whose preferred extensions are $\{x, b\}$ and $\{x, c\}$. The syntactical removal of x from F guarantees the semantical removal of x as well, since F_x's sole preferred extension is empty. However, we end up losing b and c as well from F's preferred extensions.*



If we were to delete the set of arguments $X = \{x, a_1\}$ instead, we would end up precisely with the "semantical" removal of x from F's preferred extensions, namely $\{b\} = \{x, b\} \setminus \{x\}$ and $\{c\} = \{x, c\} \setminus \{x\}$ (namely, F_X's preferred extensions). Confer Extensional Desiderata in Section 4 for more information.

Example 3 (Forget x from Sceptical Acceptance). *F's sole preferred extension in the AF below is $\{a, x, d_1, d_3\}$, and hence the arguments a, x, d_1, and d_3 are all sceptically accepted under the preferred semantics. The syntactical removal of x does not preserve any sceptical acceptance at all since F_x's sole preferred extension is empty.*



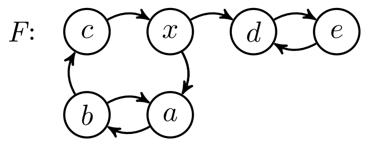
Now consider the AF F_a instead, obtained by the deletion of the argument a from F. F_a still allows us to forget x semantically, since its preferred extensions are $\{b_1, d_1, d_3\}$ and

$\{b_2, d_1, d_3\}$. Notice that unlike F_x, F_a does preserve the sceptical acceptances of d_1 and d_3. See Reasoning Desiderata in Section 4 for properties dealing with different acceptance modes.

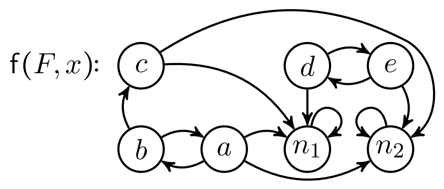
2.2 Iterative Forgetting

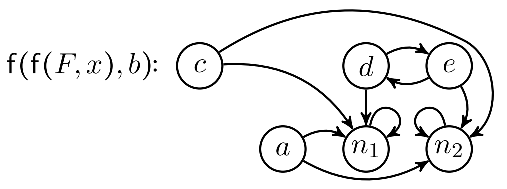
The next issue is concerned with forgetting *multiple* arguments. One natural approach for forgetting a set of arguments is to iteratively apply an already existing forgetting operator for single arguments.

Example 4 (Forget X in Different Orders)**.** *Consider the AF F below. It is easy to see that F's stable extensions are $stb(F) = \{\{x, b, e\}, \{a, c, d\}, \{a, c, e\}\}$. Now suppose we would like to forget the set of arguments $X = \{x, b\}$. One potential reasonable expectation for the forgetting operation is that we end up with an AF F' whose extensions do not include any extension overlapping with a former stable extension of F containing x or b (confer Desideratum e_4 in Section 4). This means, we are looking for an AF F' with $stb(F') = \{\{a, c, d\}, \{a, c, e\}\} = D$.*

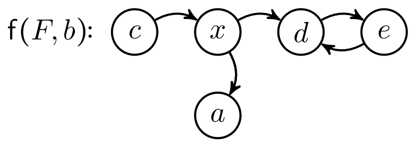


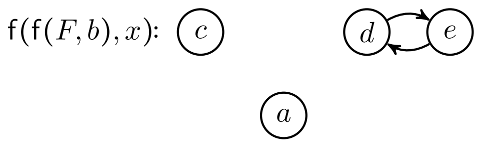
In (Baumann et al., 2020, Algorithm 1, Example 4) an operator f for forgetting single arguments (respecting the Desideratum e_4) was introduced. In order to forget the whole set X we could simply iteratively apply the operation f until all elements of X are forgotten.[1] *Obviously, there are two options, namely forgetting x first, and then b or vice-versa. We start with the first option, obtaining the AF $\mathsf{f}(\mathsf{f}(F, x), b)$ on the right below.*





If we were to forget b first and subsequently x, we would obtain the following AF.





Comparing the resulting frameworks reveals that both $\mathsf{f}(\mathsf{f}(F, b), x)$ and $\mathsf{f}(\mathsf{f}(F, b), x)$ possess the same desired set of stable extensions $D = \{\{a, c, d\}, \{a, c, e\}\}$. However, the two

1. Roughly speaking, the construction consists of two steps: First, the argument in question is syntactically removed. In the second step, any newly arisen undesired extensions E are removed by adding a self-defeating argument that is not attacked by E. More detailed information about this operator is provided in Section 6.1. The main point of this example is the order of forgetting.

frameworks are distinctively different, with the second framework being much more compact than the first. This shows that the order of application of the operation can play a decisive role in the process. See Iteration Desiderata in Section 4 for properties dealing with different orders.

2.3 Forgetting via Translation

In order to obtain reasonable forgetting operators for abstract argumentation one may try to take advantage of results and operators from related formalisms. The area of logic programming with its plenty of approaches to forgetting is a good candidate (cf. (Gonçalves, Knorr, & Leite, 2016b) for an excellent overview). However, the following two examples show that forgetting in abstract argumentation cannot be reduced to forgetting in logic programming in a straightforward manner.

Example 5 (Limits of the Standard Translation)**.** *Consider the following AF F, whose set of stable extensions is $stb(F) = \{\{b, d\}\}$. Now suppose we want to forget the argument b. We may expect as the result of the forgetting operation an AF F', s.t. $stb(F') = \{\{d\}\}$ (see Desideratum e_1). Note that by simply deleting b, we would end up with the AF F_b below, whose set of stable extensions are $stb(F_b) = \{\{a, d\}\}$. Such a purely syntactical removal would render acceptable the previously unaccepted argument a.*

$$\begin{tikzcd} F, F_b : & a & b \arrow[l, dotted] & c \arrow[l, dotted] & d \arrow[l] \end{tikzcd}$$

Let us instead consider the standard translation[2] *from AFs to logic programs (LPs) (Strass, 2013), which yields the following equivalent logic program P. Indeed P's only answer set is $\{b, d\}$ (see Section 3.2 for precise definitions).*

$$\begin{aligned} P: \quad & a \leftarrow not\, b \\ & b \leftarrow not\, c \\ & c \leftarrow not\, d \\ & d \end{aligned}$$

Now if we apply the already defined forgetting operator f_{SP} (Berthold, Gonçalves, Knorr, & Leite, 2019) to P and b (to forget b from P), we would obtain the program $\mathsf{f}_{SP}(P, b)$ given below.

$$\begin{aligned} \mathsf{f}_{SP}(P, b): \quad & a \leftarrow not\, not\, c \\ & c \leftarrow not\, d \\ & d \end{aligned}$$

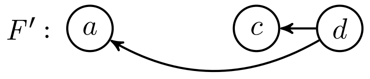
As desired, $\mathsf{f}_{SP}(P, b)$'s unique answer set is $\{d\}$. Unfortunately, $\mathsf{f}_{SP}(P, b)$ is a non-AF-like LP. Therefore, in general it is not possible to simply translate back from the resulting LP to

2. Roughly speaking, for each argument a, we associate a rule where a is the head, and the body contains all attackers of a preceded by the default negation 'not'.

an AF. In this specific case, we can find an LP P' equivalent to $\mathsf{f}_{SP}(P,b)$ which is indeed AF-like:

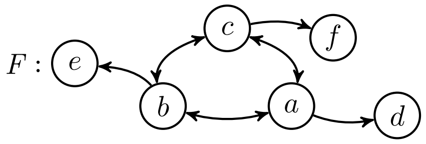
$$\begin{aligned} P': \quad & a \leftarrow not\, d \\ & c \leftarrow not\, d \\ & d \end{aligned}$$

Translating P' back results in the AF F', and indeed $stb(F') = \{\{d\}\}$, as desired.



We want to emphasize that the issue presented above, namely retranslating from logic programs to argumentation frameworks, applies to any forgetting operator in LP and not exclusively to f_{SP}. The same holds true for the next problem, as the inherent limitations of argumentation semantics cannot be ignored.

Example 6 (Representational Limits)**.** *Consider now the slightly more elaborate AF F below, whose set of stable extensions is $stb(F) = \{\{a,e,f\},\{b,f,d\},\{c,d,e\}\}$.*



Let us assume again that we want to forget the argument b (as specified by Desideratum e_1). Consequently, the expected set of extensions would be $D = \{\{a,e,f\},\{f,d\},\{c,d,e\}\}$. Since D forms a $\subseteq$-antichain, there is an LP P realising it (Eiter et al., 2013). However, we will never find an equivalent AF-like LP P' since D does not satisfy the so-called tightness property (Dunne et al., 2015) (see Section 3.1). In particular, $\{f,d\} \cup \{e\} \notin D$ but $\{e,f\} \subseteq \{a,e,f\}$ and $\{d,e\} \subseteq \{c,d,e\}$.

The examples above demonstrate that we may want to impose certain constraints defining how we want the resulting AF to look like after a forgetting operation is applied. They motivate the need for a more thorough investigation on how sets of arguments can be forgotten from argumentation frameworks. We therefore propose several possible forgetting desiderata, and conduct a systematic and comprehensive analysis of their relationships. We suggest potential operations and highlight their limits. In addition, we pay special attention to forgetting under the stable semantics firstly because it is one of the most prominent semantics, and secondly, because it shows quite a different behaviour regarding the fulfilment of the desiderata.

3. Background

3.1 Argumentation Theory

Syntax and Semantics Let $\mathcal{U}$ be an infinite background set. An *abstract argumentation framework (AF)* (Dung, 1995) is a directed graph $F = (A,R)$ with $A \subseteq \mathcal{U}$ representing arguments and $R \subseteq A \times A$ interpreted as attacks. If $(a, b) \in R$ we say that *a attacks b* or *a* is *an attacker of b*. Moreover, a set E *defends* an argument a if any attacker of a is attacked by some argument in E. In this paper, we consider finite AFs only and use the symbol $\mathcal{F}$ to denote the set of all finite AFs. Moreover, for a set $E \subseteq A$ we use E^+ for $\{b \mid (a,b) \in R, a \in E\}$ and define $E^{\oplus} = E \cup E^+$. Given an AF $F = (A, R)$, we use $A(F)$ to refer to the set A and $R(F)$ to refer to the relation R. For two AFs F and G, we define the expansion of F by G, in symbols $F \sqcup G$, as expected: $F \sqcup G = (A(F) \cup A(G), R(F) \cup R(G))$. Finally, for notational convenience we use F_X (respectively, F_x) for the AF F restricted to $A(F) \setminus X$ (respectively, F restricted to $A(F) \setminus \{x\}$). More formally, $F_X = F|_{A(F) \setminus X} = (A(F) \setminus X, R(F) \cap (A(F) \setminus X \times A(F) \setminus X))$

An *extension-based semantics* $\sigma : \mathcal{F} \to 2^{2^{\mathcal{U}}}$ is a function which assigns to any AF F a set of sets of arguments $\sigma(F) \subseteq 2^{A(F)}$. Each set of arguments $E \in \sigma(F)$ is considered to be acceptable with respect to F and is called a *σ-extension*. The most basic requirements of an extension are called *conflict-freeness* (*cf*) and *admissibility* (*ad*). Other well-studied mature semantics include complete (*co*), stage (*stg*), stable (*stb*), semi-stable (*ss*), preferred (*pr*), grounded (*gr*), ideal (*il*) and eager (*eg*). The requirements of each semantics are summarised below. An overview of argumentation semantics can be found in (Baroni, Caminada, & Giacomin, 2018).

Definition 1. *Let $F = (A, R)$ be an AF and $E \subseteq A$.*

1. *$E \in cf(F)$ iff for no $a, b \in E$, $(a, b) \in R$,*
2. *$E \in ad(F)$ iff $E \in cf(F)$ and E defends all its elements,*
3. *$E \in co(F)$ iff $E \in ad(F)$ and for any $a \in A$ defended by E, $a \in E$,*
4. *$E \in stg(F)$ iff $E \in cf(F)$ and for no $\mathcal{I} \in cf(F)$, $E^{\oplus} \subset \mathcal{I}^{\oplus}$,*
5. *$E \in stb(F)$ iff $E \in cf(F)$ and $E^{\oplus} = A$,*
6. *$E \in ss(F)$ iff $E \in ad(F)$ and for no $\mathcal{I} \in ad(F)$, $E^{\oplus} \subset \mathcal{I}^{\oplus}$,*
7. *$E \in pr(F)$ iff $E \in co(F)$ and for no $\mathcal{I} \in co(F)$, $E \subset \mathcal{I}$,*
8. *$E \in gr(F)$ iff $E \in co(F)$ and for no $\mathcal{I} \in co(F)$, $\mathcal{I} \subset E$,*
9. *$E \in il(F)$ iff $E \in ad(F)$, $E \subseteq \bigcap pr(F)$ and there is no $\mathcal{I} \in ad(F)$ satisfying $\mathcal{I} \subseteq \bigcap pr(F)$ s.t. $E \subset \mathcal{I}$,*
10. *$E \in eg(F)$ iff $E \in ad(F)$, $E \subseteq \bigcap ss(F)$ and there is no $\mathcal{I} \in ad(F)$ satisfying $\mathcal{I} \subseteq \bigcap ss(F)$ s.t. $E \subset \mathcal{I}$.*

Existence, Reasoning and Expressibility A semantics σ is *universally defined*, if $\sigma(F) \neq \emptyset$ for any $F \in \mathcal{F}$. If even $|\sigma(F)| = 1$ we say that σ is *uniquely defined*. Apart from the stable semantics all the semantics considered in this paper are universally defined. The grounded (Dung, 1995), ideal (Dung, Mancarella, & Toni, 2007) and eager semantics (Caminada, 2007) are uniquely defined (cf. (Baumann & Spanring, 2015) for an overview). Given a semantics σ, two AFs F and G are *(ordinarily) equivalent* w.r.t. σ, i.e. $F \equiv_{\sigma} G$, iff $\sigma(F) = \sigma(G)$. Moreover, two AFs are *strongly equivalent* w.r.t. σ, i.e. $F \equiv^{s}_{\sigma} G$, iff $F \sqcup H \equiv_{\sigma} G \sqcup H$ for any AF H (Oikarinen & Woltran, 2011; Baumann, 2012).

With respect to the acceptability of arguments, we consider two main reasoning modes. Given a semantics σ, an AF F, and an argument $a \in A(F)$, we say that a is *credulously accepted w.r.t.* σ if $a \in \bigcup \sigma(F)$ and that a is *sceptically accepted w.r.t.* σ if $\sigma(F) \neq \emptyset$ and $a \in \bigcap \sigma(F)$.

We say that a set of sets $\mathcal{E} \subseteq 2^{\mathcal{U}}$ is *realisable w.r.t. a semantics* σ if there is an AF F s.t. $\sigma(F) = \mathcal{E}$. realisability under stable semantics is given if and only if i) $\mathcal{E}$ forms a $\subseteq$-antichain[3] and ii) $\mathcal{E}$ is *tight* (Dunne et al., 2015). Tightness is fulfilled if for all $E \in \mathcal{E}$ and $a \in \bigcup \mathcal{E}$ we have: if $E \cup \{a\} \notin \mathcal{E}$ then there exists an $e \in E$, s.t. $(a, e) \notin \{(b, c) \mid \exists E' \in \mathcal{E} : \{b, c\} \subseteq E'\}$. Confer the set D in Example 6 for a non-tight set.

Moreover, we will frequently use that stage, semi-stable as well as preferred semantics satisfy I-maximality too (cf. (Baumann, 2018) for an overview).

3.2 Logic Programming

Syntax and Semantics We assume a set of propositional atoms $\mathcal{U}$. A *logic program* P over $\mathcal{U}$ is a finite set of rules of the form (Lifschitz, Tang, & Turner, 1999):

$$a_1 \vee \ldots \vee\ a_k \leftarrow b_1, \ldots, b_l,\ \mathit{not}\, c_1, \ldots, \mathit{not}\, c_m,\ \mathit{not}\,\mathit{not}\, d_1, \ldots, \mathit{not}\,\mathit{not}\, d_n.$$

For such a rule r let $H(r) = \{a_1, \ldots a_k\}$, $B^+(r) = \{b_1, \ldots, b_l\}$, $B^-(r) = \{c_1, \ldots, c_m\}$ and $B^{--}(r) = \{d_1, \ldots, d_n\}$. We define $\mathcal{U}(P) = \bigcup_{r \in P} H(r) \cup B^+(r) \cup B^-(r) \cup B^{--}(r)$.

Given a program P over $\mathcal{U}$ and a set of atoms $I \subseteq \mathcal{U}$, a so-called *interpretation*, the *reduct* of P w.r.t. I, is defined as $P^I = \{H(r) \leftarrow B^+(r) \mid r \in P, B^-(r) \cap I = \emptyset, B^{--}(r) \subseteq I\}$. An interpretation I is an *answer set* of P if $I \vDash P$, and for each interpretation I' we have: If $I' \vDash P^I$, then $I' \notin I$. The set of all answer sets of P is denoted by $\mathcal{AS}(P)$. We say that two programs P_1, P_2 are *equivalent* if $\mathcal{AS}(P_1) = \mathcal{AS}(P_2)$ and *strongly equivalent*, denoted by $P_1 \equiv P_2$, if $\mathcal{AS}(P_1 \cup R) = \mathcal{AS}(P_2 \cup R)$ for any program R (Lifschitz, Pearce, & Valverde, 2001). Given a set $V \subseteq \mathcal{U}$, the *V-exclusion* of a set of answer sets $\mathcal{M}$, denoted $\mathcal{M}_{\|V}$, is $\{X \setminus V \mid X \in \mathcal{M}\}$.

Example 7. *Double negations within rule-bodies is of particular interest in the context of forgetting. By allowing them, we are able to model nondeterministic 'decisions', whether an atom is true or not, without the use of auxiliary atoms. E.g. without double negation, modelling that the weather might be good, is only possible over a loop using a second atom:* $P_1 := \{\mathit{weather\_good} \leftarrow \mathit{not}\,\mathit{weather\_bad};\ \mathit{weather\_bad} \leftarrow \mathit{not}\,\mathit{weather\_good}\}$. *Then* $\mathcal{AS}(P_1) = \{\{\mathit{weather\_good}\}, \{\mathit{weather\_bad}\}\}$. *On the other hand the same circumstance may be modelled by use of a self-cycle over one atom:* $P_2 = \{\mathit{weather\_good} \leftarrow \mathit{not}\,\mathit{not}\,\mathit{weather\_good}\}$, *where* $\mathcal{AS}(P_2) = \{\emptyset, \{\mathit{weather\_good}\}\}$.

Programs with double negation are strictly more expressive than those without, as they are able to realise sets of answer-sets that are not $\subseteq$*-antichains.*

Forgetting in Logic Programming: Desiderata and Operators Let $\mathcal{P}$ be the set of all logic programs. A *forgetting operator* is a (partial) function $\mathsf{f} : \mathcal{P} \times 2^{\mathcal{U}} \rightarrow \mathcal{P}$. The program $\mathsf{f}(P, V)$ is interpreted as the *result of forgetting about V from P*. Moreover,

3. Within the argumentation community this property is usually referred to as *I-maximality* (Baroni & Giacomin, 2007).

$\mathcal{U}(\mathsf{f}(P,V)) \subseteq \mathcal{U}(P) \setminus V$ is usually required. In the following we introduce some well-known properties for forgetting operators (Gonçalves et al., 2016a).

Strong persistence is arguably the best known forgetting property (Knorr & Alferes, 2014). It requires that the result of forgetting about V from P, $\mathsf{f}(P,V)$, is strongly equivalent to the original program P, modulo the forgotten atoms.

(SP) f satisfies *strong persistence* if, for each program P and each set of atoms V, we have: $\mathcal{AS}(\mathsf{f}(P,V) \cup R) = \mathcal{AS}(P \cup R)_{\|V}$ for all programs R with $\mathcal{U}(R) \subseteq \mathcal{U} \setminus V$.

Strong invariance requires that forgetting is independent of the rules not mentioning the atoms to be forgotten.

(SI) f satisfies *strong invariance* if, for each program P and each set of atoms V, we have: $\mathsf{f}(P,V) \cup R \equiv \mathsf{f}(P \cup R, V)$ for all programs R with $\mathcal{U}(R) \subseteq \mathcal{U} \setminus V$.

Consequence persistence and its two variations are weaker forms of strong persistence dealing with ordinary equivalence only.

(CP) f satisfies *consequence persistence* if, for each P and each set of atoms V: $\mathcal{AS}(\mathsf{f}(P,V)) = \mathcal{AS}(P)_{\|V}$.

(wC) f satisfies *strengthened consequence* if, for each P and each set of atoms V: $\mathcal{AS}(\mathsf{f}(P,V)) \subseteq \mathcal{AS}(P)_{\|V}$.

(sC) f satisfies *weakened consequence* if, for each P and each set of atoms V: $\mathcal{AS}(\mathsf{f}(P,V)) \supseteq \mathcal{AS}(P)_{\|V}$.

Note that the presented desiderata are often considered for certain subclasses like *disjunctive*, *normal* or *Horn programs*. Moreover, forgetting properties can also be considered for single forgetting instances only (Berthold et al., 2019). In this article we will follow the latter interpretation.

4. Desiderata for Forgetting Operations

Given an AF F and a set of arguments $X \subseteq \mathcal{U}$, we use $\mathsf{f}(F,X)$ to denote the *result of forgetting the arguments X in F*. This means, we consider functions $\mathsf{f} : \mathcal{F} \times 2^{\mathcal{U}} \to \mathcal{F}$. We define forgetting operators over AFs, such that sets of arguments are forgotten. Forgetting a single argument x can be modelled via $X = \{x\}$. In the following we collect and define a large number of desiderata for forgetting in abstract argumentation mainly due to (Baumann et al., 2020; Baumann & Berthold, 2022). We say that a forgetting operator f *satisfies desideratum d w.r.t. instance* (F,X) iff $\mathsf{f}(F,X)$ fulfills the specified requirement. Moreover, the forgetting operator f *satisfies desideratum d* iff f satisfies d w.r.t. any instance (F,X). Note that the specification is usually parameterised with a particular semantics σ. Finally, a desideratum d is called *satisfiable* iff there is a forgetting operator f satisfying d.

We now start with presenting different groups of desiderata. We start with requirements that are concerned with purely *syntactical* changes, i.e. they impose conditions on the existence of arguments (and/or attacks) of the resulting AF.

Syntactical Desiderata. Given an AF F, a set of arguments X and a forgetting operator f.

s_1. $A(\mathsf{f}(F,X)) \cap X = \varnothing$, (no arguments from X)

s_2. $A(\mathsf{f}(F,X)) = A(F) \setminus X$, and (precise set of arguments)

s_3. $\mathsf{f}(F,X) = F_X$. (rigid AF)

Desideratum s_1 makes explicit what is often implicitly assumed for forgetting operators in other formalisms, namely that the arguments in X syntactically disappear. Desideratum s_2 requires additionally that no further arguments are removed. Condition s_3 presents the most strict condition as it precisely defines the outcome. Such a syntactical approach was firstly considered in (Bisquert et al., 2011).

The next four desiderata are concerned with the arguments either contained in every extension (sceptical acceptance) or at least in one extension (credulous acceptance).

Reasoning Desiderata. Given an AF F, a set of arguments X, a semantics σ and a forgetting operator f.

r_1. $\bigcap\sigma(\mathsf{f}(F,X)) \cap X = \varnothing$, ($X$ is not sceptically accepted)

r_2. $\bigcup\sigma(\mathsf{f}(F,X)) \cap X = \varnothing$, ($X$ is not credulously accepted)

r_3. $\bigcap\sigma(\mathsf{f}(F,X)) = (\bigcap\sigma(F)) \setminus X$, and (rigid sceptically acceptance)

r_4. $\bigcup\sigma(\mathsf{f}(F,X)) = (\bigcup\sigma(F)) \setminus X$. (rigid credulously acceptance)

Arguably the syntactical and reasoning desiderata either describe too strictly or too loosely what the outcome should be. For instance, in order to satisfy r_1 and r_2 it suffices to syntactically remove all arguments. In contrast, to satisfy r_3 or r_4 the resulting AF must entail a precise set of arguments. As a compromise between them, we suggest the following 'mixed' desiderata, that bridge syntactical and reasoning requirements.

Mixed Syntactical-Reasoning Desiderata. Given two AFs F and H, a set of arguments X, a semantics σ and a forgetting operator f.

m_1. $\bigcap\sigma(\mathsf{f}(F,X)) \subseteq A(F) \setminus X$,
(sceptical acceptance is included in the arguments not forgotten)

m_2. $\bigcup\sigma(\mathsf{f}(F,X)) \subseteq A(F) \setminus X$,
(credulous acceptance is included in the arguments not forgotten)

m_3. $\bigcap\sigma(\mathsf{f}(F,X) \sqcup H) \subseteq (A(H) \cup A(F)) \setminus X$ for all AFs H with $A(H) \subseteq \mathcal{U} \setminus X$, and
(forgotten arguments are never sceptically accepted)

m_4. $\bigcup\sigma(\mathsf{f}(F,X) \sqcup H) \subseteq (A(H) \cup A(F)) \setminus X$ for all AFs H with $A(H) \subseteq \mathcal{U} \setminus X$.
(forgotten arguments are never credulously accepted)

Condition m_1 (resp. m_2) requires that, if there are new arguments added while forgetting, they be irrelevant to sceptical (resp. credulous) reasoning. In other words, that these arguments are purely administrative. Then m_3 (resp. m_4) require new arguments to be irrelevant, even under the addition of new information.

An alternative way of imposing restrictions is to introduce dependencies between former and new extensions. In fact, there are many different ways to relate them.

Extensionality Desiderata. Given an AF F, a set of arguments X, a semantics σ and a forgetting operator f.

e_1. $\sigma(\mathsf{f}(F,X)) = \{E \setminus X \mid E \in \sigma(F)\}$, ($X$-adjusted extension)

$e_{\mathbf{wC}}$. $\sigma(\mathsf{f}(F,X)) \supseteq \{E \setminus X \mid E \in \sigma(F)\}$, (no X-adjusted extension is lost)

$e_{\mathbf{sC}}$. $\sigma(\mathsf{f}(F,X)) \subseteq \{E \setminus X \mid E \in \sigma(F)\}$, (no further extensions are added)

e_2. $\sigma(\mathsf{f}(F,X) \sqcup H) = \{E \setminus X \mid E \in \sigma(F \sqcup H)\}$ for any H with $A(H) \subseteq \mathcal{U} \setminus X$, (delete X even from any future extension)

$e_{3_\subseteq}$. $\sigma(\mathsf{f}(F,X)) = \{T(E) \mid E \in \sigma(F)\}$ with $T : \sigma(F) \to 2^{\mathcal{U}}$ and $E \mapsto T(E) \subseteq E \setminus X$, (subsets of X-adjusted extension)

$e_{3_\supseteq}$. $\sigma(\mathsf{f}(F,X)) = \{T(E) \mid E \in \sigma(F)\}$ with $T : \sigma(F) \to 2^{\mathcal{U}}$ and $E \mapsto T(E) \supseteq E \setminus X$, and (supersets of X-adjusted extension)

e_4. $\sigma(\mathsf{f}(F,X)) = \sigma(F) \setminus \{E \mid E \in \sigma(F), E \cap X \neq \varnothing\}$. (remove X-overlapping extensions)

Desideratum e_1 requires that any former extension has to survive in an adjusted fashion, namely new extensions are obtained from initial ones via deleting the arguments that have to be forgotten. Any subsequent desiderata, apart from e_4, is a variation of e_1. For instance, $e_{\mathbf{wC}}$ allows for outputting additional extensions beyond the adjusted ones. In contrast, $e_{\mathbf{sC}}$ demands that only adjusted extensions are are considered reasonable positions.

Desiderata $e_{3_\subseteq}$ and $e_{3_\supseteq}$ relax e_1 in another direction. They require that for each former extension E the forgetting result has an extension that is a subset (resp. superset) of $E \setminus X$.

Finally, Desideratum e_4 requires the full removal of extensions containing arguments that have to be forgotten.

Note that desiderata $e_{\mathbf{wC}}$ and $e_{\mathbf{sC}}$ correspond to **(wC)** and **(sC)** (as introduced in Section 3.2), hence their name. The desiderata e_1 as well as e_2 could be alternatively renamed as $e_{\mathbf{CP}}$ and $e_{\mathbf{SP}}$, as **(CP)** and **(SP)** are the equivalent forgetting properties in the context of logic programming. However, for continuity we decided to keep the original denomination as introduced in (Baumann et al., 2020).

Next, we postulate desiderata that aim to constrain the amount of change to an AF. In particular, the following *vacuity* desiderata provide conditions under which a given framework does not require any changes.

Vacuity Desiderata. Given an AF F, a set of arguments X, a semantics σ and a forgetting operator f.

v_1. If $\bigcap \sigma(F) \cap X = \varnothing$, then $F = \mathsf{f}(F,X)$, (sceptical vacuity)

v_2. If $\bigcup \sigma(F) \cap X = \varnothing$, then $F = \mathsf{f}(F,X)$, and (credulous vacuity)

v_3. If $A(F) \cap X = \varnothing$, then $F = \mathsf{f}(F,X)$. (argument vacuity)

In particular the initial framework is required to stay unchanged if none of the forgotten arguments X appears in F in case of v_1, or if no argument of X is sceptically (resp. credulously) accepted in case of v_2 (resp. v_3).

It is advantageous to confine the construction of a forgetting result in some way. For comparison, some forgetting operators in logic programming have been shown to be able to disregard rules that do not mention the atoms to be forgotten, i.e. they satisfy the discussed property **(SI)**. Similarly, when forgetting arguments from an AF we could require that parts of the framework that do not stand in (close) contact to the arguments to be forgotten can be left unchanged.

Locality Desiderata. Given an AF F, a set of arguments X, a semantics σ and a forgetting operator f.

l_1. $\mathsf{f}(F,X) \sqcup H \equiv_\sigma \mathsf{f}(F \sqcup H, X)$ for all AFs H with $A(H) \subseteq \mathcal{U} \setminus X$, and (f and $\sqcup$ are compatible)

l_2. $\mathsf{f}(F,X) \sqcup H \equiv_\sigma \mathsf{f}(F \sqcup H, X)$ for all AFs H with $A(H) \subseteq \mathcal{U} \setminus (X \cup \{a \mid \exists x \in X,\ \text{s.t. } (a,x) \in R(F) \text{ or } (x,a) \in R(F)\})$. (less tolerant refinement of compatibility)

As shown in Example 4 there are forgetting operators where the result of forgetting arguments iteratively highly depends on the considered ordering. Moreover, in case of logic programs there are cases where atoms cannot be correctly forgotten iteratively in *any* order (Berthold, 2022), meaning that forgetting the whole set at once is always different from forgetting iteratively. The following desiderata describe either independence of the forgetting order or that there is at least one reconstructing ordering.

Iteration Desiderata. Given an AF F, a set of arguments $X = \{x_1, \dots x_n\}$ and a forgetting operator f.

i_1. for each bijection $b : X \to X$: $\mathsf{f}(F,X)) = \mathsf{f}(\mathsf{f}(\dots \mathsf{f}(F, b(x_1)) \dots, b(x_{n-1})), b(x_n)))$, and (independence of the chosen ordering)

i_2. there is a bijection $b : X \to X$ with $\mathsf{f}(F,X)) = \mathsf{f}(\mathsf{f}(\dots \mathsf{f}(F, b(x_1)) \dots, b(x_{n-1})), b(x_n)))$. (there is a reconstructing forgetting order)

4.1 Relationships between Single Desiderata

We proceed with an analysis of the dependencies between single desiderata. More precisely, we clarify the following question, namely: Does an operator satisfy desideratum c' w.r.t. instance (F,X), if the operator satisfies desideratum c w.r.t. the same instance (F,X).

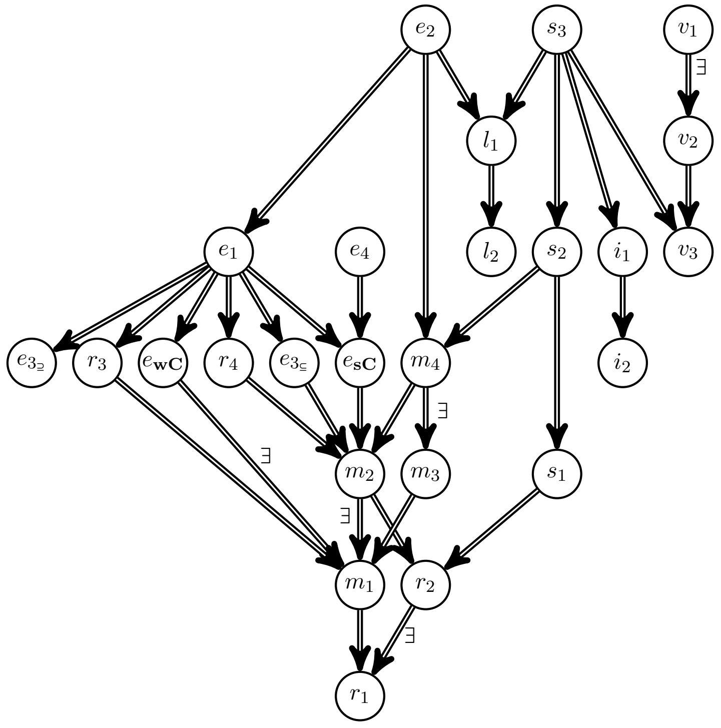


Figure 1: General Relations between Desiderata. Edges are marked with '$\exists$', if the implication requires the existence of at least one extension.

Proposition 1. *For $\sigma \in \{stg, stb, ss, pr, gr, il, eg\}$ and conditions c and c' in the diagram of Figure 1, a path from c to c' indicates that: Given an AF F and a set of arguments X, any function f satisfying c w.r.t. (F, X) under σ also satisfies c' w.r.t. (F, X) under σ. Moreover, in case of $\sigma \in \{stg, stb, ss, pr\}$ only these relationships between pairs of desiderata hold.*

Proof. Given an AF F and a set of arguments X. We start by proving the valid dependencies. We mention that in case of stable semantics we have to be a bit more careful since it is the only considered semantics not being universally defined. Consequently, familiar properties like sceptical accepted arguments are credulously accepted too are no longer necessarily true (cf. (Baumann & Spanring, 2015) for more information). If providing at least one σ-extension is essential for the relation, we use the shorthand "$\exists_{stb}$". For instance, $X \subseteq_{\exists_{stb}} Y$ means that the subset-relation holds under the assumption that at least one stable extension exists.

- $e_1 \Rightarrow r_3$: $\bigcap \sigma(\mathsf{f}(F, X)) = \bigcap\{E \setminus X \mid E \in \sigma(F)\} = (\bigcap\{E \mid E \in \sigma(F)\}) \setminus X = (\bigcap \sigma(F)) \setminus X$.
- $e_1 \Rightarrow r_4$: $\bigcup \sigma(\mathsf{f}(F, X)) = \bigcup\{E \setminus X \mid E \in \sigma(F)\} = (\bigcup\{E \mid E \in \sigma(F)\}) \setminus X = (\bigcup \sigma(F)) \setminus X$.
- $e_1 \Rightarrow e_{\mathbf{sC}}, e_{\mathbf{wC}}$: Obviously, $\sigma(\mathsf{f}(F, X)) = \{E \setminus X \mid E \in \sigma(F)\}$ implies $\sigma(\mathsf{f}(F, X)) \circ \{E \setminus X \mid E \in \sigma(F)\}$ for each $\circ \in \{\subseteq, \supseteq\}$.

- $e_1 \Rightarrow e_{3_\supseteq}, e_{3_\subseteq}$: $\sigma(\mathsf{f}(F,X)) = \{E \setminus X \mid E \in \sigma(F)\}$ implies $T(E) = E \setminus X$ and thus, $T(E) \circ E \setminus X$ for each $\circ \in \{\subseteq, \supseteq\}$.
- $e_4 \Rightarrow e_{\mathbf{sC}}$: We have $\sigma(F) \setminus \{E \mid E \in \sigma(F), E \cap X \neq \emptyset\} \subseteq \{E \setminus X \mid E \in \sigma(F)\}$.
- $e_{\mathbf{sC}} \Rightarrow m_2$: $\bigcup \sigma(\mathsf{f}(F,X)) \subseteq \bigcup\{E \setminus X \mid E \in \sigma(F)\} \subseteq A(F) \setminus X$ since for each extension $E \in \sigma(F)$, $E \subseteq A(F)$ is implied.
- $e_{3_\subseteq} \Rightarrow m_2$: $\bigcup \sigma(\mathsf{f}(F,X)) \subseteq \bigcup\{T(E) \mid E \in \sigma(F)\}$ with $T : \sigma(F) \to 2^{\mathcal{U}}$ and $E \mapsto T(E) \subseteq E \setminus X$. Consequently, $\bigcup\{T(E) \mid E \in \sigma(F)\} \subseteq \bigcup\{E \setminus X \mid E \in \sigma(F)\} \subseteq A(F) \setminus X$ as argued above.
- $e_2 \Rightarrow e_1$: Consider $H_\emptyset = (\emptyset, \emptyset)$.
- $e_2 \Rightarrow m_4$: Let H be an AF with $A(H) \subseteq \mathcal{U} \setminus X$. We have: $\bigcup \sigma(\mathsf{f}(F,X) \sqcup H) = \bigcup\{E \setminus X \mid E \in \sigma(F \sqcup H)\} \subseteq (A(H) \cup A(F)) \setminus X$.
- $e_2 \Rightarrow l_1$: $\sigma(\mathsf{f}(F,X) \sqcup H) = \{E \setminus X \mid E \in \sigma(F \sqcup H)\} = \{E \setminus X \mid E \in \sigma(F \sqcup H \sqcup H_\emptyset)\} = \sigma(\mathsf{f}(F \sqcup H, X) \sqcup H_\emptyset) = \sigma(\mathsf{f}(F \sqcup H, X))$
- $l_1 \Rightarrow l_2$: Obviously, $\mathcal{U} \setminus (X \cup \{a \mid \exists x \in X, \text{ s.t. } (a,x) \in R \text{ or } (x,a) \in R\}) \subseteq \mathcal{U} \setminus X$
- $r_3 \Rightarrow m_1$: $\bigcap \sigma(\mathsf{f}(F,X)) = (\bigcap \sigma(F)) \setminus X \subseteq A(F) \setminus X$
- $r_4 \Rightarrow m_2$: $\bigcup \sigma(\mathsf{f}(F,X)) = (\bigcup \sigma(F)) \setminus X \subseteq A(F) \setminus X$
- $m_3 \Rightarrow m_1$ and $m_4 \Rightarrow m_2$: Consider $H_\emptyset$.
- $m_4 \Rightarrow m_3$: $\bigcap \sigma(\mathsf{f}(F,X) \sqcup H) \subseteq_{\exists_{stb}} \bigcup \sigma(\mathsf{f}(F,X) \sqcup H) \subseteq (A(H) \cup A(F)) \setminus X$
- $m_2 \Rightarrow m_1$: $\bigcap \sigma(\mathsf{f}(F,X)) \subseteq_{\exists_{stb}} \bigcup \sigma(\mathsf{f}(F,X)) \subseteq A(F) \setminus X$
- $m_1 \Rightarrow r_1$ and $m_2 \Rightarrow r_2$: Obvious since $(A(F) \setminus X) \cap X = \emptyset$.
- $s_3 \Rightarrow s_2$: $A(\mathsf{f}(F,X)) = A(F|_{A(F) \setminus X}) = A(F) \setminus X$.
- $s_3 \Rightarrow v_3$: Let $A(F) \cap X = \emptyset$. Consequently, $\mathsf{f}(F,X) = F|_{A(F) \setminus X} = F$.
- $s_3 \Rightarrow l_1$: We have $\mathsf{f}(F,X) \sqcup H = F|_{A(F) \setminus X} \sqcup H = F|_{A(F) \setminus X} \sqcup H|_{A(H) \setminus X}$ since $A(H) \subseteq \mathcal{U} \setminus X$ is assumed. Consequently, $\mathsf{f}(F,X) \sqcup H = F \sqcup H|_{A(F \sqcup H) \setminus X} = \mathsf{f}(F \sqcup H, X)$ implying $\mathsf{f}(F,X) \sqcup H \equiv \mathsf{f}(F \sqcup H, X)$.
- $s_2 \Rightarrow s_1$: $A(\mathsf{f}(F,X)) \cap X = (A(F) \setminus X) \cap X = \emptyset$.
- $s_2 \Rightarrow m_4$: $\bigcup \sigma(\mathsf{f}(F,X) \sqcup H) \subseteq A(\mathsf{f}(F,X) \sqcup H) = A(\mathsf{f}(F,X)) \cup A(H) = (A(F) \setminus X) \cup A(H) = (A(H) \cup A(F)) \setminus X$ since for the AF H we have $A(H) \subseteq \mathcal{U} \setminus X$.
- $r_2 \Rightarrow r_1$: Obviously, $\bigcup \sigma(\mathsf{f}(F,X)) \cap X = \emptyset \Rightarrow_{\exists_{stb}} \bigcap \sigma(\mathsf{f}(F,X)) \cap X = \emptyset$.
- $s_1 \Rightarrow r_2$: $\bigcup \sigma(\mathsf{f}(F,X)) \cap X \subseteq A(\mathsf{f}(F,X)) \cap X = \emptyset$.
- $v_1 \Rightarrow v_2$: $\bigcup \sigma(F) \cap X = \emptyset \Rightarrow_{\exists_{stb}} \bigcap \sigma(F) \cap X = \emptyset \Rightarrow F = \mathsf{f}(F,X)$.

- $v_2 \Rightarrow v_3$: $A(F) \cap X = \emptyset \Rightarrow \bigcup\sigma(F) \cap X = \emptyset \Rightarrow F = \mathsf{f}(F, X)$.

- $s_3 \Rightarrow i_1$: Desideratum s_3 requires the removal of any argument (together with its corresponding attacks) in X. Clearly, this procedure is independent of the order.

- $i_1 \Rightarrow i_2$: Obviously, since any bijection, and thus resulting order, does the job.

- $e_{\mathbf{wC}} \Rightarrow m_1$. $\bigcap\sigma(\mathsf{f}(F, X)) \subseteq_{\exists_{stb}} E \setminus X$ for some $E \in \sigma(F)$. Since $E \subseteq A(F)$ we are done.

Now we show that the remaining dependencies do not hold. In order to reduce the number of proofs, we make use of the following observation. Let there be four desiderata $d_1, \dots, d_4$, s.t. $d_1 \Rightarrow d_2$ and $d_3 \Rightarrow d_4$. If we are able to show $d_1 \not\Rightarrow d_4$, then we also have that $d_2 \not\Rightarrow d_4$ and $d_1 \not\Rightarrow d_3$. The following proofs suffice to show that the graph in Figure 1 is complete w.r.t. to the semantics $\sigma \in \{stg, stb, ss, pr\}$.

- $e_{\mathbf{wC}} \not\Rightarrow r_2$: Let $\sigma \in \{stg, stb, ss, pr\}$ and $F = (\{a, x\}, \emptyset)$. Consequently, $\sigma(F) = \{\{a, x\}\}$. Consider an operator f s.t. $\mathsf{f}(F, \{x\}) = (\{a, x\}, \{(a, x), (x, a)\})$. Hence, $\sigma(\mathsf{f}(F, \{x\})) = \{\{a\}, \{x\}\}$. This means, f satisfies $e_{\mathbf{wC}}$ w.r.t. $(F, \{x\})$ but not r_2 as x is credulously accepted in $\mathsf{f}(F, \{x\})$.

- $e_{\mathbf{wC}} \not\Rightarrow r_3$: Let $\sigma \in \{stg, stb, ss, pr\}$. Consider the AF $F = (\{a, b, c, x\}, \{(a, x), \{(x, a), (a, c), (c, a))$ and a forgetting operator f, s.t. $\mathsf{f}(F, \{x\}) = (\{a, b, c, x\}, \{(a, x), (x, a), (b, x), (x, b), (c, x), (x, c), (a, c), (c, a)\})$. We have $\sigma(F) = \{\{a, b\}, \{b, c, x\}\}$ and $\sigma(\mathsf{f}(F, \{x\})) = \{\{a, b\}, \{b, c\}, \{x\}\}$. Thus, f satisfies $e_{\mathbf{wC}}$ w.r.t. $(F, \{x\})$ since $\{a, b\} \setminus \{x\} = \{a, b\}$ and $\{b, c, x\} \setminus \{x\} = \{b, c\}$ are contained in $\sigma(\mathsf{f}(F, \{x\}))$. However, r_3 is violated as $\bigcap\sigma(\mathsf{f}(F, \{x\})) = \emptyset \neq \{b\} = (\bigcap\sigma(F)) \setminus \{x\}$.

- $e_{3_{\supseteq}} \not\Rightarrow m_2$: Let $\sigma \in \{stg, stb, ss, pr\}$ and $F = (\{a, x\}, \emptyset)$. Consequently, $\sigma(F) = \{\{a, x\}\}$. Consider an operator f s.t. $\mathsf{f}(F, \{x\}) = (\{a, x\}, \{(a, x), (x, a)\})$. Hence, $\sigma(\mathsf{f}(F, \{x\})) = \{\{a\}, \{x\}\}$. This means, f satisfies $e_{\mathbf{wC}}$ w.r.t. $(F, \{x\})$ but not m_2 as x is credulously accepted in $\mathsf{f}(F, \{x\})$.

- $e_{3_{\supseteq}} \not\Rightarrow m_3$: Let $\sigma \in \{stg, stb, ss, pr\}$ and $F = (\{a, x\}, \emptyset)$. Consequently, $\sigma(F) = \{\{a, x\}\}$. Consider an operator f s.t. $\mathsf{f}(F, \{x\}) = (\{a, x\}, \{(a, x)\})$. Hence, $\sigma(\mathsf{f}(F, \{x\})) = \{\{a\}\}$. This means, f satisfies $e_{\mathbf{wC}}$ w.r.t. $(F, \{x\})$. Consider now $H = (\{a, b\}, \{(b, a)\})$, then $\sigma(\mathsf{f}(F, \{x\}) \sqcup H) = \{b, x\}$, meaning that $\sigma(\mathsf{f}(F, \{x\}) \sqcup H) \cap \{x\} = \{x\} \neq \emptyset$, hence m_3 is not satisfied for $(F, \{x\})$.

- $e_{3_{\supseteq}} \not\Rightarrow r_1$: Let $\sigma \in \{stg, stb, ss, pr, gr, il, eg\}$ and F like above. This means, $\sigma(F) = \{\{a, x\}\}$. Let f be the identity, i.e. $\mathsf{f}(F, \{x\}) = F$. Hence, f satisfies $e_{3_{\supseteq}}$ w.r.t. $(F, \{x\})$ but not r_1 as x sceptically accepted in $\mathsf{f}(F, \{x\}) = F$.

- $l_1 \not\Rightarrow r_1$: Let $\sigma \in \{stg, stb, ss, pr, gr, il, eg\}$, f the identity function, and $F = (\{x\}, \emptyset)$. Then, l_1 is obviously satisfied for $(F, \{x\})$ (even for any instance) but r_1 is not, as x is sceptically accepted in $\mathsf{f}(F, \{x\}) = F$.

- $i_1 \not\Rightarrow r_1$: Let $\sigma \in \{stg, stb, ss, pr, gr, il, eg\}$, f be the identity function, and $F = (\{a, x\}, \emptyset)$. Then f satisfies i_1 w.r.t. $(F, \{x\})$ but not r_1, as x is sceptically accepted in $\mathsf{f}(F, \{x\}) = F$.

- $v_1 \not\Rightarrow r_1$: Let $\sigma \in \{stg, stb, ss, pr, gr, il, eg\}$, f be the identity function, and $F = (\{a, x\}, \varnothing)$. Then f satisfies v_1 w.r.t. $(F, \{x\})$ but not r_1, as x is sceptically accepted in $\mathsf{f}(F, \{x\}) = F$

- $r_3 \not\Rightarrow r_2$: Let $\sigma \in \{stg, stb, ss, pr\}$, $F = (\{a, x\}, \{(a, x), (x, a)\})$ be an AF, and f be a forgetting operator, s.t. $\mathsf{f}(F, \{x\}) = F$. Then f satisfies r_3 for $(F, \{x\})$ but not r_2 as x is credulously accepted in $\mathsf{f}(F, \{x\}) = F$.

- $m_3 \not\Rightarrow r_2$: Let $\sigma \in \{stg, ss, pr\}$. Consider the AF $F = (\{x, y\}, \{(x, y), (y, x)\})$ and a forgetting operator f, s.t. $\mathsf{f}(F, \{x, y\}) = F$. We have $\sigma(F) = \{\{x\}, \{y\}\}$ Please note that any considered semantics σ satisfies so-called non-interference (van der Torre & Vesic, 2017; Baroni, Caminada, & Giacomin, 2011). This means that the semantics of two disjoint AFs F and H is obtained via combining single extensions from F and H. Moreover, the considered semantics σ are universally defined. Consequently, for any H over $\mathcal{U} \smallsetminus \{x, y\}$ we have that $\{x\} \cup E \in \sigma(\mathsf{f}(F, \{x, y\}) \sqcup H)$ and $\{x\} \cup E \in \sigma(\mathsf{f}(F, \{x, y\}) \sqcup H)$ for some $E \in \sigma(H)$. Hence, $\bigcap \sigma(\mathsf{f}(F, \{x, y\}) \sqcup H) = \varnothing$ showing that m_3 is satisfied w.r.t. $(F, \{x, y\})$. In contrast, r_2 is not satisfied, as both x and y are credulously accepted $\sigma(\mathsf{f}(F, \{x, y\}))$

- $s_1 \not\Rightarrow m_1$: Let $\sigma \in \{stg, stb, ss, pr, gr, il, eg\}$, $F = (\{x\}, \varnothing)$, f be a forgetting operator, s.t. $\mathsf{f}(F, \{x\})) = (\{a\}, \varnothing)$. Then f satisfies s_1 for $(F, \{x\})$, as the forgotten arguments are not contained in the result, but not m_1, as new arguments are introduced.

- $i_1 \not\Rightarrow v_3$: Let f be the constant forgetting operator mapping any input to $G = (\varnothing, \varnothing)$. Consider $F = (\{a\}, \varnothing)$. Then f satisfies i_1 w.r.t. $(F, \{x\})$, since $\{x\}$ can only be forgotten in one order. However v_3 does not hold since F changed, although x does not appear in F.

- $s_2 \not\Rightarrow v_3$: Let $\sigma \in \{stg, stb, ss, pr, gr, il, eg\}$ and f a forgetting operator, s.t. $\mathsf{f}(F, X) = (A(F) \smallsetminus X, \varnothing)$ for any AF F and any set X. Consider $F = (\{a, b\}, \{(a, b)\})$. Obviously, f satisfies s_2 w.r.t. $(F, \{x\})$ as all non-forgotten arguments are still included. However, v_3 is violated since F is changed, although x does not appear in F.

- $e_4 \not\Rightarrow v_3$: Let $\sigma \in \{stg, stb, ss, pr, gr, il, eg\}$. Consider the AF $F = (\{a, b\}, \{(a, b), (b, b)\})$ and a forgetting operator f, s.t. $\mathsf{f}(F, \{x\}) = (\{a\}, \varnothing)$. We have $\sigma(F) = \sigma(\mathsf{f}(F, \{x\})) = \{\{a\}\}$. Hence, f satisfies e_4 for $(F, \{x\})$ as there are no $\{x\}$-overlapping extensions. However, v_3 is violatd as the forgetting result and the initial framework are not syntactically equal.

- $e_2 \not\Rightarrow i_2$: For each semantics $\sigma \in \{stg, stb, ss, pr, gr, il, eg\}$ one may find two syntactically different AFs F and G being strongly equivalent (Oikarinen & Woltran, 2011). The proof idea is to consider such two AFs and a set of two arguments $\{x, y\}$ disjoint from $A(F)$. The forgetting operator now maps F to G if forgetting x and y simultaneously and results in F if forgetting only one single argument. In this way the described operator satisfies e_2 for the instance $(F, \{x, y\})$ but not i_2.
  For the sake of clarity we give a concrete example. Let $\sigma = stb$, $F = (\{a, b\}, \{(a, a), (a, b)\})$. Moreover, the forgetting operator f is defined as $\mathsf{f}(F, \{x, y\}) = (\{a, b\}, \{(a, a)\}) = G$, and $\mathsf{f}(F, \{z\}) = F$ for each $z \in \mathcal{U}$. Please note that $F \equiv^s_{stb} G$ as both have the same

stable kernels (Oikarinen & Woltran, 2011). Consequently, by definition of strong equivalence we have: $\mathit{stb}(\mathsf{f}(F,\{x,y\}) \sqcup H) = \mathit{stb}(G \sqcup H) = \mathit{stb}(F \sqcup H)$ for any AF H. Since $\{x,y\} \cap A(F) = \varnothing$ we have $\{E \smallsetminus \{x,y\} \mid E \in \mathit{stb}(F \sqcup H)\} = \mathit{stb}(F \sqcup H)$ for any H with $A(H) \subseteq \mathcal{U} \smallsetminus \{x,y\}$. Hence, desideratum e_2 is fulfilled for the instance $(F,\{x,y\})$. However, $F \neq G$ and $F = \mathsf{f}(\mathsf{f}(F,\{x\}),\{y\}) = \mathsf{f}(\mathsf{f}(F,\{y\}),\{x\})$ proves that i_2 is violated for $(F,\{x,y\})$.

- $e_4 \not\Rightarrow i_2$: Consider the above case $e_2 \not\Rightarrow i_2$. We will now use the same idea as well as counterexample. We have already shown that i_2 does not hold. Since F and G are strongly equivalent they possess the same extensions. Moreover, $\{x,y\}$ does not intersect with the arguments in F and G. Consequently, no extensions contain x or y. Hence, f satisfies e_4 w.r.t. $(F,\{x,y\})$.

- $e_2 \not\Rightarrow v_3$: Again we use the idea and counterexample from $e_2 \not\Rightarrow i_2$. We have already shown that e_2 is fulfilled. Since $A(F) \cap \{x,y\} = \varnothing$, but $F \neq G$ we obtain that f violates v_3 w.r.t. $(F,\{x,y\})$.

- $s_2 \not\Rightarrow i_2$: The concrete semantics does not matter as we consider syntactical criteria. Consider a forgetting operator f, s.t. $\mathsf{f}(F,\{z\}) = (A(F) \smallsetminus \{z\}, \varnothing)$ for each $z \in \mathcal{U}$. Moreover, if X is not a singleton we have $\mathsf{f}(F,X) = (A(F) \smallsetminus X, R(F) \smallsetminus X \times R(F) \smallsetminus X)$. Consequently, for $F = (\{a,x,y\},\varnothing)$ we have and $\mathsf{f}(F,\{x,y\}) = (\{a\},\{(a,a)\})$ but $\mathsf{f}(\mathsf{f}(F,\{x\}),\{y\}) = \mathsf{f}(\mathsf{f}(F,\{y\}),\{x\}) = (\{a\},\varnothing)$. Hence, s_2 is satisfied for $(F,\{x,y\})$, but i_2 is not.

- $v_1 \not\Rightarrow i_2$: Let $\sigma \in \{\mathit{stg}, \mathit{stb}, \mathit{ss}, \mathit{pr}, \mathit{gr}, \mathit{il}, \mathit{eg}\}$ and f a forgetting operator, s.t. $\mathsf{f}(F,X) = (A(F) \cup \{c\}, R(F) \cup \{(a,c) \mid a \in A(F)\})$ for some fresh argument c, i.e. $c \notin A(F)$. Consider now $F = (\{a\},\varnothing)$. Obviously, $\bigcap \sigma(F) = \{a\}$. Further, we obtain $\mathsf{f}(F,\{a,x\}) = (\{a,c\},\{(a,c)\})$. Since $\bigcap \sigma(F) \cap \{a,x\} = \{a\} \neq \varnothing$ we see that f satisfies v_1 w.r.t. $(F,\{a,x\})$. However, according to the construction both frameworks $\mathsf{f}(\mathsf{f}(F,\{a\}),\{x\})$ and $\mathsf{f}(\mathsf{f}(F,\{x\}),\{a\})$ contain two new arguments which prevents them from coinciding with $\mathsf{f}(F,\{a,x\})$. This means, i_2 is violated.

- $e_{3\subseteq} \not\Rightarrow e_{\mathbf{wC}}$: Let $\sigma \in \{\mathit{stg}, \mathit{stb}, \mathit{ss}, \mathit{pr}, \mathit{gr}, \mathit{il}, \mathit{eg}\}$ and f the constant operator mapping any input to $G = (\varnothing,\varnothing)$. Consider $F = (\{a\},\varnothing)$. Hence, $\sigma(F) = \{\{a\}\}$ and $\sigma(\mathsf{f}(F,\{x\}) = \{\varnothing\}$. Since $\varnothing \subseteq \{a\} \smallsetminus \{x\}$ we have that f satisfies $e_{3\subseteq}$ w.r.t. $(F,\{x\})$. Obviously, $e_{\mathbf{wC}}$ is violated as $\{a\} \smallsetminus \{x\} = \{a\} \notin \{\varnothing\}$.

- $e_{3\supseteq} \not\Rightarrow e_{\mathbf{wC}}$: Let $\sigma \in \{\mathit{stg}, \mathit{stb}, \mathit{ss}, \mathit{pr}, \mathit{gr}, \mathit{il}, \mathit{eg}\}$ and f the identity function mapping any input to itself. Consider $F = (\{x\},\varnothing)$. Consequently, $\sigma(F) = \{\{x\}\} = \sigma(\mathsf{f}(F,\{x\}))$. Since $\{x\} \smallsetminus \{x\} = \varnothing \subseteq \{x\}$ we have that f satisfies $e_{3\supseteq}$ w.r.t. $(F,\{x\})$. However, $e_{\mathbf{wC}}$ is violated as $\{x\} \smallsetminus \{x\} = \varnothing \notin \{\{x\}\}$.

- $r_4 \not\Rightarrow e_{\mathbf{wC}}$: Let us consider the semantics $\sigma \in \{\mathit{stg}, \mathit{stb}, \mathit{ss}, \mathit{pr}\}$ only. Consider the AF $F = (\{a,b,c,x\},\{(a,x),(b,x),(c,x),(x,a),(x,b),(x,c),(b,c),(c,b)\})$ and the forgetting operator f with $\mathsf{f}(F,\{x\}) = (\{a,b,c\},\{(a,b),(a,c),(b,a),(b,c),(c,a),(c,b)\})$. Hence, $\sigma(F) = \{\{a,b\},\{a,c\},\{x\}\}$ and $\sigma(\mathsf{f}(F,\{x\})) = \{\{a\},\{b\},\{c\}\}$. Consequently, f satisfies r_4 w.r.t. $(F,\{x\})$ as $\bigcup \sigma(\mathsf{f}(F,\{x\})) = \{a,b,c\} = \{a,b,c,x\} \smallsetminus \{x\} = (\bigcup \sigma(F)) \smallsetminus$

$\{x\}$. Moreover, we see that $e_{\mathbf{wC}}$ is violated as for instance, $\{a,b\} \setminus \{x\} = \{a,b\} \notin \{\{a\},\{b\},\{c\}\}$.

- $r_3 \not\Rightarrow e_{\mathbf{wC}}$: Let $\sigma \in \{stg, stb, ss, pr\}$. Consider the $F = (\{a,x\},\{(a,x),(x,a)\})$ and a forgetting operator f, s.t. $\mathsf{f}(F,\{x\}) = (\{a,b\},\{(a,b),(b,a)\})$. We have $\sigma(F) = \{\{a\},\{x\}\}$ as well as $\sigma(\mathsf{f}(F,\{x\})) = \{\{a\},\{b\}\}$. This means, f satisfies r_3 w.r.t. $(F,\{x\})$ as $\bigcap \sigma(\mathsf{f}(F,\{x\})) = \varnothing = \varnothing \setminus \{x\} = (\bigcap \sigma(F)) \setminus \{x\}$. However, we see that $e_{\mathbf{wC}}$ is violated as for instance, $\{x\} \setminus \{x\} = \varnothing \notin \{\{a\},\{b\}\}$.

- $e_4 \not\Rightarrow e_{\mathbf{wC}}$: Let $\sigma \in \{stg, stb, ss, pr\}$. Consider the AF $F = (\{a,b,x\},\{(a,b), (a,x), (b,a), (x,a)\})$ and let f be a forgetting operator, s.t. $\mathsf{f}(F,\{x\}) = (\{a\},\varnothing)$. We have $\sigma(F) = \{\{a\},\{x,b\}\}$ and $\sigma(\mathsf{f}(F,\{x\})) = \{\{a\}\}$. Consequently, f satisfies e_4 w.r.t. $(F,\{x\})$ as any extension containing x (i.e. $\{x,b\}$) is removed. However, $e_{\mathbf{wC}}$ is violated as $\{x,b\} \setminus \{x\} = \{b\} \notin \{\{a\}\}$.

- $s_3 \not\Rightarrow e_{\mathbf{wC}}$: Let $\sigma \in \{stg, stb, ss, pr, gr, il, eg\}$. Consider the AF $F = (\{a,x\},\{(x,a)\})$ and let f be a forgetting operator, s.t. $\mathsf{f}(F,\{x\}) = (\{a\},\varnothing)$. Consequently, f satisfies s_3 for $(F,\{x\})$ as $\mathsf{f}(F,\{x\}) = F_{\{x\}}$. Since $\sigma(F) = \{\{x\}\}$ and $\sigma(\mathsf{f}(F,\{x\})) = \{\{a\}\}$ we see that $e_{\mathbf{wC}}$ is violated because $\{x\} \setminus \{x\} = \varnothing \notin \{\{a\}\}$.

- $v_1 \not\Rightarrow e_{\mathbf{wC}}$: Let $\sigma \in \{stg, stb, ss, pr, gr, il, eg\}$, $F = (\{x,a\},\{(x,a)\})$ and $\mathsf{f}(F,\{x\}) = F$. Then $\sigma(\mathsf{f}(F,\{x\})) = \sigma(F) = \{\{x\}\}$. f satisfies v_1 for $(F,\{x\})$, as the forgetting result is equal to the initial framework, but not $e_{\mathbf{wC}}$, as $\varnothing \notin \sigma(\mathsf{f}(F,\{x\}))$.

- $r_3 \not\Rightarrow e_{3_\supseteq}$: Let $\sigma \in \{stg, stb, ss, pr\}$. Consider the AF $F = (\{a,b,c,x\},\{(a,x),(b,x),(c,x),(x,a),(x,b),(x,c),(b,c),(c,b)\})$ and let f be a forgetting operator, s.t. $\mathsf{f}(F,\{x\}) = (\{a,b,c\},\{(a,b), (b,a), (a,c), (c,a), (b,c), (c,b)\})$. Hence, $\sigma(F) = \{\{a,b\},\{a,c\},\{x\}\}$ and $\sigma(\mathsf{f}(F,\{x\})) = \{\{a\},\{b\},\{c\}\}$. Consequently, there are no sceptically accepted arguments in both frameworks. This means, f satisfies r_3 w.r.t. $(F,\{x\})$. However, $e_{3_\supseteq}$ is violated as there are no supersets of $\{a,b\}$ nor of $\{b,c\}$ in $\sigma(\mathsf{f}(F,\{x\}))$.

- $e_{3_\subseteq} \not\Rightarrow e_{3_\supseteq}$: Let $\sigma \in \{stg, stb, ss, pr, gr, il, eg\}$ and f be a constant forgetting operator mapping any input to the empty AF $G = (\varnothing,\varnothing)$. Consider the AF $F = (\{a\},\varnothing)$. Then f satisfies $e_{3_\subseteq}$ w.r.t. $(F,\{x\})$ as $\varnothing \in \sigma(\mathsf{f}(F,\{x\}))$ is a subset of $\{a\}$. However, $e_{3_\supseteq}$ is violated since there is no superset of $\{a\}$ in $\sigma(\mathsf{f}(F,\{x\}))$.

- $e_{\mathbf{wC}} \not\Rightarrow e_{3_\supseteq}$: Let $\sigma \in \{stg, stb, ss, pr\}$, $F = (\{a\},\varnothing)$ and $\mathsf{f}(F,\{x\}) = (\{a,b\},\{(a,b),(b,a)\})$. Then $\sigma(F) = \{\{a\}\}$ and $\sigma(\mathsf{f}(F,\{x\})) = \{\{a\},\{b\}\}$. Consequently, f satisfies $e_{\mathbf{wC}}$ for $(F,\{x\})$, as $\{a\} \setminus \{x\} = \{a\} \in \sigma(\mathsf{f}(F,\{x\}))$. In contrast, $e_{3_\supseteq}$ is violated since $\{b\}$ is no superset of $\{a\} \setminus \{x\}$.

- $r_4 \not\Rightarrow e_{3_\supseteq}$: Let $\sigma \in \{stg, stb, ss, pr\}$. Consider the AF $F = (\{a,b,c,x\},\{(a,x), (b,x), (c,x), (x,a), (x,b), (x,c), (b,c), (c,b)\})$ and let f be a forgetting operator, s.t. $\mathsf{f}(F,\{x\}) = (\{a,b,c\},\{(a,b),(b,a),(a,c),(c,a),(b,c),(c,b)\})$. Hence, $\sigma(F) = \{\{a,b\}, \{a,c\}, \{x\}\}$ and $\sigma(\mathsf{f}(F,\{x\})) = \{\{a\},\{b\},\{c\}\}$. Then f satisfies r_4 w.r.t. $(F,\{x\})$ since $\{a,b,c\} = \{a,b,c,x\} \setminus \{x\}$. In contrast, $e_{3_\supseteq}$ is violated since there is no mapping T, s.t. each set $\{a\}$, $\{b\}$ and $\{c\}$ is the image of exactly one $E \in \sigma(F)$, s.t. $T(E) \supseteq E \setminus \{x\}$. For instance, in case of $\{a,b\} \in \sigma(F)$ neither $\{a\}$, $\{b\}$ nor $\{c\}$ is a superset of $\{a,b\} \setminus \{x\} = \{a,b\}$.

- $e_4 \not\Rightarrow e_{3_\supseteq}$: Let $\sigma \in \{stg, stb, ss, pr\}$. Consider $F = (\{a,b,x\}, \{(a,x),(x,a),(a,b),(b,a)\})$ and let f be a forgetting operator, s.t. $\mathsf{f}(F,\{x\}) = (\{a\},\emptyset)$. We have $\sigma(F) = \{\{a\},\{x,b\}\}$ and $\sigma(\mathsf{f}(F,\{x\})) = \{\{a\}\}$. Hence, f satisfies e_4 w.r.t. $(F,\{x\})$ since $\{x,b\}$ is eliminated as required. However, $e_{3_\supseteq}$ is violated as the mapping T enforces $T(\{x,b\}) = \{a\}$ and $\{a\} \supseteq \{x,b\} \smallsetminus \{x\} = \{b\}$ does not hold.

- $s_3 \not\Rightarrow e_{3_\supseteq}$: Let $\sigma \in \{stg, stb, ss, pr, gr, il, eg\}$. Consider the AF $F = (\{a,b,x\}, \{(x,a),(a,b)\})$ and $\mathsf{f}(F,\{x\}) = (\{a,b\},\{(a,b)\})$. We have $\sigma(F) = \{\{x,b\}\}$ and $\sigma(\mathsf{f}(F,\{x\})) = \{\{a\}\}$. The operator f satisfies s_3 w.r.t. $(F,\{x\})$ as $\mathsf{f}(F,\{x\}) = F_{\overline{\{x\}}}$. However, $e_{3_\supseteq}$ is violated as the mapping T enforces $T(\{x,b\}) = \{a\}$ and $\{a\} \supseteq \{x,b\} \smallsetminus \{x\} = \{b\}$ does not hold.

- $v_1 \not\Rightarrow e_{3_\supseteq}$: Let $\sigma \in \{stg, stb, ss, pr, gr, il, eg\}$. Consider the AF $F = (\{a,x\},\emptyset)$ and $\mathsf{f}(F,\{x\}) = (\{b,x\},\emptyset)$. We have $\sigma(F) = \{\{a,x\}\}$ and $\sigma(\mathsf{f}(F,\{x\})) = \{\{b,x\}\}$. Consequently, f satisfies v_1 w.r.t. $(F,\{x\})$ trivially, as its precondition is not met (i.e. x is sceptically accepted in F). However, $e_{3_\supseteq}$ is violated as the mapping T enforces $T(\{a,x\}) = \{b,x\}$ and $\{b,x\} \supseteq \{a,x\} \smallsetminus \{x\} = \{a\}$ does not hold.

□

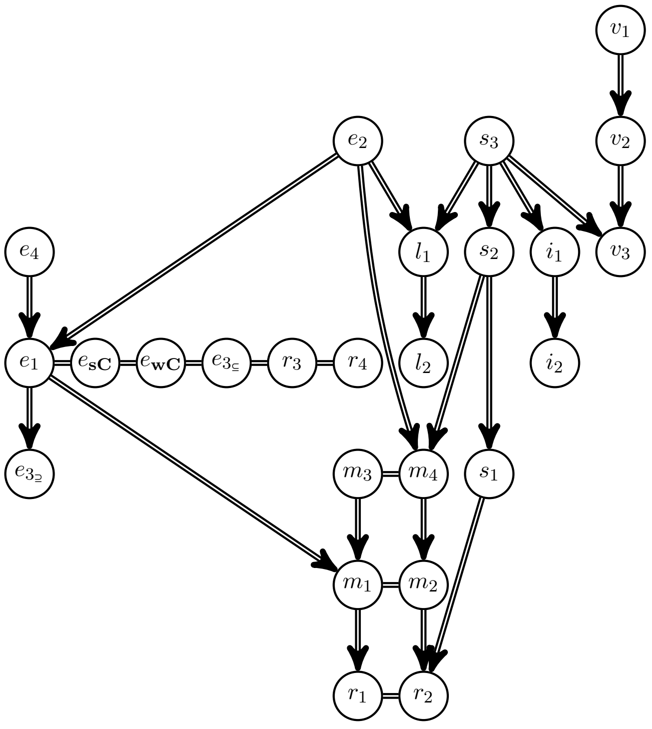


Figure 2: Relationships between desiderata under uniquely defined semantics

Next, we refine the previously established general relations for uniquely defined semantics.

Proposition 2. *For $\sigma \in \{gr, il, eg\}$ and conditions c and c' in the diagram of Figure 2, a path from c to c' indicates that: Given an AF F and a set of arguments X any function f satisfying c w.r.t. (F, X) under σ also satisfies c' w.r.t. (F, X) under σ. Moreover, only these relationships between pairs of desiderata hold.*

Proof. Any new dependency, i.e. $e_{\mathbf{wC}} \Rightarrow e_1$, $e_{\mathbf{sC}} \Rightarrow e_1$, $r_3 \Rightarrow e_1$, $r_4 \Rightarrow e_1$, $r_1 \Rightarrow r_2$, $m_1 \Rightarrow m_2$, and $m_3 \Rightarrow m_4$ hold because the semantics $\sigma \in \{gr, il, eg\}$ are uniquely defined $(*)$. In particular, we can take advantage of the facts that given two unary extension-sets $\{E_1\}$ and $\{E_2\}$, $\{E_1\} \subseteq \{E_2\}$ implies $\{E_1\} = \{E_2\}$, as well as that $\bigcup\{E_1\} = \bigcap\{E_1\} = E_1$. The shown implications together with results depicted in Figure 1 yield that the following pairs of desiderata (m_3, m_4), (m_1, m_2), (r_1, r_2) as well as (x, y) with $x, y \in \{e_{\mathbf{wC}}, e_1, e_{\mathbf{sC}}, e_{3_\subseteq}, r_3, r_4\}$ collapse.

- $e_{\mathbf{wC}} \Rightarrow e_1$: $\sigma(\mathsf{f}(F, X)) \supseteq \{E \setminus X \mid E \in \sigma(F)\} \Rightarrow^{(*)} \sigma(\mathsf{f}(F, X)) = \{E \setminus X \mid E \in \sigma(F)\}$
- $e_{\mathbf{sC}} \Rightarrow e_1$: $\sigma(\mathsf{f}(F, X)) \subseteq \{E \setminus X \mid E \in \sigma(F)\} \Rightarrow^{(*)} \sigma(\mathsf{f}(F, X)) = \{E \setminus X \mid E \in \sigma(F)\}$

In the following two cases we use $\sigma(F) = \{E\}$.

- $r_3 \Rightarrow e_1$: $\bigcap \sigma(\mathsf{f}(F, X)) = (\bigcap \sigma(F)) \setminus X = E \setminus X \Rightarrow^{(*)} \sigma(\mathsf{f}(F, X)) = \{E \setminus X \mid E \in \sigma(F)\}$
- $r_4 \Rightarrow e_1$: $\bigcup \sigma(\mathsf{f}(F, X)) = (\bigcup \sigma(F)) \setminus X = E \setminus X \Rightarrow^{(*)} \sigma(\mathsf{f}(F, X)) = \{E \setminus X \mid E \in \sigma(F)\}$

In the following three cases we use $\bigcap \sigma(\mathsf{f}(F, X)) = \bigcup \sigma(\mathsf{f}(F, X))$ as well as $\bigcap \sigma(\mathsf{f}(F, X) \sqcup H) = \bigcup \sigma(\mathsf{f}(F, X) \sqcup H)$ due to uniquely definedness.

- $r_1 \Rightarrow r_2$: $\bigcap \sigma(\mathsf{f}(F, X)) \cap X = \emptyset \Rightarrow^{(*)} \bigcup \sigma(\mathsf{f}(F, X)) \cap X = \emptyset$
- $m_1 \Rightarrow m_2$: $\bigcap \sigma(\mathsf{f}(F, X)) \subseteq A(F) \setminus X \Rightarrow^{(*)} \bigcup \sigma(\mathsf{f}(F, X)) \subseteq A(F) \setminus X$
- $m_3 \Rightarrow m_4$: $\bigcap \sigma(\mathsf{f}(F, X) \sqcup H) \subseteq (A(H) \cup A(F)) \setminus X$ for all AFs H with $A(H) \subseteq \mathcal{U} \setminus X \Rightarrow^{(*)} \bigcup \sigma(\mathsf{f}(F, X) \sqcup H) \subseteq (A(H) \cup A(F)) \setminus X$ for all AFs H with $A(H) \subseteq \mathcal{U} \setminus X$

All other (non)dependencies are proved in Proposition 1. □

4.2 Relationships involving Multiple Desiderata

Apart from the relationships concerning single conditions there are more complex relationships involving sets of desiderata. In the realm of logic programming it was already shown that **(SP)** is necessary and sufficient for **(SI)** and **(CP)** (Gonçalves et al., 2016a). Beside other interesting relations we will show the analogous result for abstract argumentation (cf. Item 5 of the subsequent proposition). The following assertions are meant in the same way as Proposition 1, that is they consider satisfiability w.r.t. instances. For example, Item 1 states that if an operator simultaneously satisfies desiderata s_2, l_1 and $e_{3_\subseteq}$ w.r.t. a certain instance (F, X), then the operator satisfies desideratum e_2 w.r.t. instance (F, X).

Proposition 3. *For any semantics $\sigma \in \{stg, stb\}$:*

1. *s_2, l_1 and $e_{3_\subseteq}$ imply e_2,*
2. *s_2, l_1 and $e_{3_\supseteq}$ imply e_2.*

Moreover, for any $\tau \in \{stg, stb, ss, pr, gr, il, eg\}$ we have:

3. $e_{3_\subseteq}$ *and* $e_{3_\supseteq}$ *if and only if* e_1*,*

4. $e_{\mathbf{sC}}$ *and* $e_{\mathbf{wC}}$ *if and only if* e_1*,*

5. e_1 *and* l_1 *if and only if* e_2*.*

Proof. 1. Given an AF F and a set of arguments $X \subseteq \mathcal{U}$. Let f satisfy s_2, l_1 and $e_{3_\subseteq}$ w.r.t. (F, X). In order to show desideratum e_2 we will first prove that condition e_1 is implied. Then, applying Item 5 of this Proposition yields e_2. Desideratum e_1 requires $\sigma(\mathsf{f}(F,X)) = \{E \smallsetminus X \mid E \in \sigma(F)\}$. Due to s_2 we have $A(\mathsf{f}(F,X)) = A(F) \smallsetminus X := A$. Define a copy of A, i.e. a set of fresh arguments $A' = \{a' \mid a \in A\}$, s.t. $A' \cap (A(F) \cup X) = \emptyset$. Let us define the AF $H = (A \cup A', \{(a,a') \mid a \in A\})$. By construction any argument $a \in A$ attacks its copy $a' \in A'$. Consequently, for any extension $E \in \sigma(F)$ we have, $E \cup \{a' \in A' \mid a \in A \smallsetminus E\} \in \sigma(F \sqcup H)$. Vice versa, any $E' \in \sigma(F \sqcup H)$ can be uniquely associated with some $E \in \sigma(F)$, s.t. $E' = E \cup \{a' \in A' \mid a \in A \smallsetminus E\}$. Therefore, for both semantics, stable and stage, we deduce for any $a \in A$: either $a \in E'$ or (its copy) $a' \in E'$. The same relation between extensions applies to $\mathsf{f}(F,X)$ and $\mathsf{f}(F,X) \sqcup H$. Furthermore, since $\mathsf{f}(F,X) \sqcup H \equiv \mathsf{f}(F \sqcup H, X)$ due to l_1 we may even conclude the same behaviour regarding arguments $a \in A$ for $\mathsf{f}(F \sqcup H, X)$. More precisely, for any extension $E' \in \sigma(\mathsf{f}(F \sqcup H, X))$, either $a \in E'$ or $a' \in E'$.

($\supseteq$) If $E \in \sigma(F)$, then $E \smallsetminus X \in \sigma(\mathsf{f}(F,X))$.
Since $E \in \sigma(F)$ we deduce $E \cup \{a' \in A' \mid a \in A \smallsetminus E\} \in \sigma(F \sqcup H)$. Due to $e_{3_\subseteq}$ we obtain $T(E \cup \{a' \in A' \mid a \in A \smallsetminus E\}) \subseteq (E \cup \{a' \in A' \mid a \in A \smallsetminus E\}) \smallsetminus X = E \smallsetminus X \cup \{a' \in A' \mid a \in A \smallsetminus E\}$ for some function $T : \sigma(F \sqcup H) \to 2^{\mathcal{U}}$ and $T(E \cup \{a' \in A' \mid a \in A \smallsetminus E\}) \in \sigma(\mathsf{f}(F \sqcup H, X))$. Assuming $T(E \cup \{a' \in A' \mid a \in A \smallsetminus E\}) \subsetneq E \smallsetminus X \cup \{a' \in A' \mid a \in A \smallsetminus E\}$ yields the existence of an $a \in A$, s.t. neither $a \in T(E \cup \{a' \in A' \mid a \in A \smallsetminus E\})$, nor $a' \in T(E \cup \{a' \in A' \mid a \in A \smallsetminus E\})$. Contradiction. Hence, $T(E \cup \{a' \in A' \mid a \in A \smallsetminus E\}) = E \smallsetminus X \cup \{a' \in A' \mid a \in A \smallsetminus E\}$. Moreover, applying l_1 justifies $E \smallsetminus X \cup \{a' \in A' \mid a \in A \smallsetminus E\} \in \sigma(\mathsf{f}(F,X) \sqcup H)$. In consideration of the one-to-one correspondence we deduce $E \smallsetminus X \in \sigma(\mathsf{f}(F,X))$ as claimed.

($\subseteq$) If $E' \in \sigma(\mathsf{f}(F,X))$, then $E' = E \smallsetminus X$ for some $E \in \sigma(F)$.
Due to condition $e_{3_\subseteq}$ we know $\sigma(\mathsf{f}(F,X)) = \{T(E) \mid E \in \sigma(F)\}$ with $T : \sigma(F) \to 2^{\mathcal{U}}$ and $E \mapsto T(E) \subseteq E \smallsetminus X$. This means, there is some $E \in \sigma(F)$, s.t. $E' \subseteq E \smallsetminus X$. Applying the former case ($\supseteq$) yields $E \smallsetminus X \in \sigma(\mathsf{f}(F,X))$. Since stable as well as stage semantics satisfy I-maximality, i.e. $\sigma(\mathsf{f}(F,X))$ has to form a $\subseteq$-antichain we deduce $E' = E \smallsetminus X$ concluding the proof.

2. This proof is analogous to the previous one.

3. ($\Leftarrow$) Confer Proposition 1.
($\Rightarrow$) Let F be an AF and $X \subseteq \mathcal{U}$ a set of arguments. Let f satisfy $e_{3_\supseteq}$ and $e_{3_\subseteq}$ w.r.t. (F, X). Consider now a certain extension $E \in \tau(F)$. Due to $e_{3_\supseteq}$ and $e_{3_\subseteq}$ we deduce that there are two functions $T_1, T_2 : \tau(F) \to 2^{\mathcal{U}}$ s.t. $T_1(E) \supseteq E \smallsetminus X$ and $T_2(E) \subseteq E \smallsetminus X$. Since any considered semantics satisfies I-maximality, i.e. $\tau(\mathsf{f}(F,X))$ has to form a $\subseteq$-antichain we deduce $T_1(E) = E \smallsetminus X = T_2(E)$. Hence, $\tau(\mathsf{f}(F,X)) = \{E \smallsetminus X \mid E \in \tau(F)\}$ as required.

4. Obvious.

5. ($\Leftarrow$) Confer Proposition 1.
($\Rightarrow$) $\tau(\mathsf{f}(F,X) \sqcup H) =^{(l_1)} \tau(\mathsf{f}(F \sqcup H, X)) =^{(e_1)} \{E \setminus X \mid E \in \tau(F \sqcup H)\}$

□

5. Satisfiability and Unsatisfiability of Desiderata

We now turn to the problem of satisfiability in general. More precisely, we tackle the question whether there are forgetting operators satisfying a certain desiderata d. This means, in order to prove an affirmative answer, one has to come up with a concrete operator satisfying d w.r.t. any instance (F,X). Beside some positive results we also reveal intrinsic limits of forgetting operators in abstract argumentation.

5.1 Satisfiability of Individual Desiderata

We first show that the majority of the desiderata are satisfiable. This means, we may find/construct operators satisfying the desideratum in question for any instance. We will consider stable semantics in a seperate proposition as satisfiability sometimes require the existence of extensions.

Proposition 4. *Let* $d \in \{e_{3_\supseteq}, e_{3_\subseteq}, e_{\mathbf{sC}}, r_1, r_2, r_3, r_4, s_1, s_2, s_3, m_1, m_2, m_3, m_4, v_1, v_2, v_3, l_1, l_2, i_1, i_2\}$ *and* $\sigma \in \{stg, ss, pr, gr, il, eg\}$. *Desideratum* d *is satisfiable w.r.t. semantics* σ.

Proof. It suffices to show that each desideratum $d \in \{e_{3_\supseteq}, e_{3_\subseteq}, e_{\mathbf{sC}}, r_3, r_4, v_1, s_3\}$ is satisfiable w.r.t. any instance (F,X). The satisfiability of the remaining desiderata is then implied by Proposition 1. In the following we present concrete forgetting operators handling any instance (F,X) in the desired way.

- $e_{3_\supseteq}$: If $\sigma(F) = \varnothing$, then set $\mathsf{f}(F,X) = F$. If not, define $\mathsf{f}(F,X) = (\bigcup_{E\in\sigma(F)} E \setminus X, \varnothing)$. Consequently, $\sigma(\mathsf{f}(F,X)) = \{\bigcup_{E\in\sigma(F)} E \setminus X\}$. Thus, the constant function $T : \sigma(F) \to 2^{\mathcal{U}}$ with $E \mapsto T(E) = \bigcup_{E\in\sigma(F)} E \setminus X$ satisfies $T(E) \supseteq E \setminus X$ for any $E \in \sigma(F)$ as desired.

- $e_{3_\subseteq}$: If $\sigma(F) = \varnothing$, then set $\mathsf{f}(F,X) = F$. If not, define $\mathsf{f}(F,X) = (\varnothing,\varnothing)$. Thus, the constant function $T : \sigma(F) \to 2^{\mathcal{U}}$ with $E \mapsto T(E) = \varnothing$ satisfies $T(E) \subseteq E \setminus X$ for any $E \in \sigma(F)$ as required.

- $e_{\mathbf{sC}}$: If $\sigma(F) = \varnothing$, then set $\mathsf{f}(F,X) = F$. If not, just pick an arbitrary $E \in \sigma(F)$ and define $\mathsf{f}(F,X) = (E \setminus X, \varnothing)$. Obviously, $\sigma(\mathsf{f}(F,X)) = \{E \setminus X\}$ justifying $\sigma(\mathsf{f}(F,X)) \subseteq \{E \setminus X \mid E \in \sigma(F)\}$.

- r_3: Consider f, s.t. if $\bigcap \sigma(F) = \varnothing$, then $\mathsf{f}(F,X) = \varnothing$, else $\mathsf{f}(F,X) = ((\bigcap \sigma(F)) \setminus X, \varnothing)$.

- r_4: Define $\mathsf{f}(F,X) = ((\bigcup \sigma(F)) \setminus X, \varnothing)$.

- v_1: Set $\mathsf{f}(F,X) = F$.

- s_3: Set $\mathsf{f}(F,X) = F_X$. □

Now we turn to stable semantics as promised.

Proposition 5. *Let $d \in \{e_{3_{\supseteq}}, e_{3_{\subseteq}}, e_{\mathbf{sC}}, r_1, r_2, r_4, s_1, s_2, s_3, m_1, m_2, m_4, v_1, v_2, v_3, l_1, l_2, i_1, i_2\}$. Desideratum d is satisfiable w.r.t. stable semantics.*

Proof. Note that the proofs presented in Proposition 4, namely $d \in \{e_{3_{\supseteq}}, e_{3_{\subseteq}}, e_{\mathbf{sC}}, r_3, r_4, v_1, s_3\}$ is satisfiable, apply to stable semantics too. However, regarding the consequences in case of stable semantics we have to be careful as some implications hold in case of non-collapsing AFs only (cf. Figure 1). This means, some relations require the existence of stable extensions. In the following we show the missing desiderata $d \in \{m_1, r_1, v_2\}$.

- m_1: Let $\mathsf{f}(F, X) = (\varnothing, \varnothing)$ for any instance (F, X). Hence, $\bigcap stb(\mathsf{f}(F, X)) = \varnothing \subseteq A(F) \smallsetminus X$.
- r_1: Using the shown satisfiability of m_1 immediately yields the satisfiability of r_1 w.r.t. stable semantics (cf. Proposition 1).
- v_2: Set $\mathsf{f}(F, X) = F$.

□

In contrast to the other considered semantics we have to give up the satisfiability of r_3 and m_3 in case of stable semantics.

Proposition 6. *Let $d \in \{r_3, m_3\}$. Desideratum d is not satisfiable w.r.t. stable semantics, i.e. there are instances s.t. any operator fails to satisfy d.*

Proof.
- m_3: Given an AF F, a set X and an operator f. Consider further an AF $H = (\{a\}, \{(a, a)\})$ with some fresh argument $a \notin A(\mathsf{f}(F, X))$. Consequently, $stb(\mathsf{f}(F, X) \cup H) = \varnothing$ implying $\bigcap stb(\mathsf{f}(F, X) \cup H) = \mathcal{U}$. This means, $\bigcap stb(\mathsf{f}(F, X) \cup H) \nsubseteq (A(H) \cup A(F)) \smallsetminus X$ which contradicts m_3.
- r_3: This impossibility is mainly due to the finite assumption. Consider the AF $F = (\{x\}, \{(x, x)\})$ and $X = \{x\}$. This means, $stb(F) = \varnothing$ implying $\bigcap stb(F) = \mathcal{U}$. Consequently, there is no forgetting operator f with $\bigcap stb(\mathsf{f}(F, X)) = \mathcal{U} \smallsetminus \{x\}$ as $\bigcap stb(\mathsf{f}(F, X))$ is either $\mathcal{U}$ (in case of no stable extensions) or a finite set (in case of non-collasping) as any extension is a subset of $A(\mathsf{f}(F, X))$.

□

The following assertion shows a dividing line between the considered uniquely defined and universally defined semantics. The I-maximality of the latter prevent the satisfiability of e_1 and $e_{\mathbf{wC}}$ w.r.t. arbitrary inputs.

Proposition 7. *Desiderata e_1 and $e_{\mathbf{wC}}$ are satisfiable w.r.t. semantics $\tau \in \{gr, il, eg\}$. However, there are instances s.t. any operator fails to satisfy e_1 as well as $e_{\mathbf{wC}}$ under semantics $\sigma \in \{stb, stg, ss, pr\}$.*

Proof. Consider the uniquely defined semantics τ. For any AF F we obtain a singleton as extension-set, i.e. $\tau(F) = \{E\}$. Define $\mathsf{f}(F, X) = (E \smallsetminus X, \varnothing)$. Thus, $\tau(\mathsf{f}(F, X)) = \{E \smallsetminus X\}$ proving e_1 and therefore $e_{\mathbf{wC}}$.
Consider now the remaining semantics σ and let $F = (\{a, x\}, \{(a, x), (x, a)\})$ be a specific AF. We obtain $\sigma(F) = \{\{a\}, \{x\}\}$. According to $e_{\mathbf{wC}}$ it must hold $\sigma(\mathsf{f}(F, \{x\})) \supseteq \{\varnothing, \{a\}\}$. This means that $\sigma(\mathsf{f}(F, \{x\}))$ does not form a $\subseteq$-antichain. Hence, no such $\mathsf{f}(F, \{x\})$ can exist. □

We proceed with showing that there are no operators satisfying desideratum e_2. This means, for any chosen operator we may find instances violating the desired property e_2.[4]

Proposition 8. *Desideratum e_2 is not satisfiable w.r.t. semantics $\sigma \in \{stg, stb, ss, pr, gr, il, eg\}$.*

Proof. Let us consider first the universally defined semantics $\sigma \in \{stg, stb, ss, pr\}$. We already know that satisfiability of e_2 implies the satisfiability of e_1 (Proposition 1). Moreover, in Proposition 7 we have shown that e_1 is not satisfiable w.r.t. σ. Therefore, neither is e_2.
Consider now the remaining semantics $\tau \in \{gr, il, eg\}$. We fix the AF $F = (\{a, x\}, \{(x, a)\})$ and let $X = \{x\}$. Striving for a contradiction assume the existence of an AF $G = \mathsf{f}(F, X)$, s.t. for any AF H with $x \notin A(H)$ we have: $\tau(G \sqcup H) = \{E \smallsetminus \{x\} \mid E \in \tau(F \sqcup H)\}$. In order to obtain a contradiction we will augment F with different AFs. Consider first $H_1 = (\{a, b\}, \{(a, b)\})$. We have $\tau(F \sqcup H_1) = \{\{x, b\}\}$. Consequently, by assumption $\tau(G \sqcup H_1) = \{\{b\}\}$. For $\tau = gr$ we already have a contradiction since if the singleton $\{b\}$ is grounded in an AF, then $\{b\}$ has to be unattacked in this AF. Obviously, this is not the case for $G \sqcup H_1$.
Consider now the ideal and eager semantics. Both are based on admissibility. Hence, $\{b\}$ is admissible in $G \sqcup H_1$. This means, $(b, a) \in R(G)$ as b is attacked in $G \sqcup H_1$. Please note that $a, b \in A(G)$ is implied.
Now we do the same procedure with the AF $H_2 = (\{a, c\}, \{(a, c)\})$, i.e. instead of a attacks b we consider a attacks c. Via the same reasoning we finally obtain that $a, c \in A(G)$.
Now, the decisive step. Consider the AF $H_3 = (\{a, b, c\}, \{(a, b)\})$. We already know $a, b, c \in A(G)$. Consequently, we find $G \sqcup H_1 = G \sqcup H_3$. However, $\tau(G \sqcup H_3) = \{\{x, b, c\} \smallsetminus \{x\}\} \neq \{\{b\}\} = \tau(G \sqcup H_1)$. Contradiction! □

The stable semantics is able to precisely cut off undesired extensions. No other considered semantics shares this property.

Proposition 9. *Desideratum e_4 is satisfiable w.r.t. stable semantics, but not under any semantics $\tau \in \{stg, ss, pr, gr, il, eg\}$.*

Proof. The satisfiability of e_4 w.r.t. stable semantics is an immediate consequence of Proposition 13.
Consider now the remaining semantics $\tau \in \{stg, ss, pr, gr, il, eg\}$. In case of the AF $F = (\{x\}, \varnothing)$ we obtain $\tau(F) = \{\{x\}\}$. Hence, desideratum e_4 would require the deletion of $\{x\}$, i.e. $\mathsf{f}(F, x) = \varnothing$. This is impossible due to universal definedness of τ. □

4. We mention that the original proof (cf. (Baumann et al., 2020, Proposition 4)) contained minor errors which have now been corrected.

5.2 Satisfiability of Combinations of Desiderata

In this section we consider the satisfiability of combinations of desiderata, i.e., of whole sets of conditions. Please note that this is not just an academic task, as for certain applications one might be interested in satisfying several desiderata at once or, in case of inherent limits, to satisfy as many desiderata as possible from a set of desired properties. In particular, combinations of syntactical and semantical desiderata are quite natural requirements.

In what follows, we analyse some combinations involving extensionality conditions. Our analysis is not exhaustive, but it serves as an illustration of how the satisfiability of specific combinations can be investigated. We conclude the section with a summary of all satisfiability and unsatisfiability results shown and how they can be used to determine the status of further combinations.

Proposition 10. *We have the following satisfiability results:*

1. *Let $\mu \in \{stg, stb\}$. The sets $\{s_2, l_1, e_{3_\subseteq}\}$ as well as $\{s_2, l_1, e_{3_\supseteq}\}$ are not satisfiable w.r.t. μ.*
2. *The sets $\{e_{3_\subseteq}, e_{3_\supseteq}\}$ and $\{e_{\mathbf{sC}}, e_{\mathbf{wC}}\}$ are satisfiable for each semantics $\tau \in \{gr, il, eg\}$, but not satisfiable w.r.t. semantics $\sigma \in \{stg, stb, ss, pr\}$.*

Proof. 1. Let $\mu \in \{stg, stb\}$. According to Items 1 and 2 of Proposition 3 we have that $\{s_2, l_1, e_{3_\subseteq}\}$ as well as $\{s_2, l_1, e_{3_\supseteq}\}$ imply e_2. The latter requirement is not satisfiable due to Proposition 8. Hence, both sets are not satisfiable under semantics μ.

2. Given $\sigma \in \{stb, stg, ss, pr\}$ and $\tau \in \{gr, il, eg\}$. The sets of desiderata $\{e_{3_\subseteq}, e_{3_\supseteq}\}$ respective $\{e_{\mathbf{sC}}, e_{\mathbf{wC}}\}$ are equivalent to e_1 in terms of satisfiability (Items 3 and 4 of Proposition 3). Consequently, applying Proposition 7 we obtain that both sets are satisfiable w.r.t. τ, but not w.r.t. σ.

$\square$

The next two results once again demonstrate that stable semantics is somehow exceptional regarding its potential for forgetting. More precisely, while individual satisfiability of the subsequent desiderata is guarenteed for any considered semantics, only the stable one exhibits simultaneous satisfiability.

Proposition 11. *The set $\{l_1, e_{\mathbf{sC}}\}$ is satisfiable w.r.t. stable semantics, but not under $\sigma \in \{stg, ss, pr, gr\}$.*

Proof. • Given an AF F and a set X. We define a forgetting operator via $\mathsf{f}(F, X) = (A(F), R(F) \setminus \{(a, x) \mid x \in X\} \cup \{(x, x) \mid x \in X \cap A(F)\})$.
Regarding l_1 please note that $\mathsf{f}(F, X) \sqcup H = \mathsf{f}(F \sqcup H, X)$ if considering AFs H, s.t. $A(H) \cap X = \emptyset$. Consequently, l_1 is fulfilled w.r.t. any semantics as ordinary equivalence of $\mathsf{f}(F, X) \sqcup H$ and $\mathsf{f}(F \sqcup H, X))$ is implied.
We now turn to $e_{\mathbf{sC}}$. If $A(F) \cap X = \emptyset$, then $\mathsf{f}(F, X) = F$ and hence, $\sigma(\mathsf{f}(F, X)) = \sigma(F)$ for any semantics σ. If not, i.e. $A(F) \cap X \neq \emptyset$ we observe that $\mathsf{f}(F, X)$ collapses for stable semantics, i.e. $stb(\mathsf{f}(F, X)) = \emptyset$. In both cases, $stb(\mathsf{f}(F, X)) \subseteq \{E \setminus X \mid E \in stb(F)\}$ yielding $e_{\mathbf{sC}}$.

- Towards a contradiction suppose that there is a forgetting operator f satisfying l_1 and $e_{\mathbf{sC}}$ under $\sigma \in \{ss, pr, gr\}$. Let us consider the instance (F, X) with $F = (\{x, a\}, \{(x, a)\})$ and $X = \{x\}$.

$$\begin{tikzcd} F\colon & x \arrow[r] & a & & H\colon & a \arrow[r] & b \end{tikzcd}$$

We have $\sigma(F) = \{\{x\}\}$ and $\sigma(\mathsf{f}(F, X)) \neq \emptyset$ due to universal definedness. Applying condition $e_{\mathbf{sC}}$, i.e. $\sigma(\mathsf{f}(F, X)) \subseteq \{E \setminus X \mid E \in \sigma(F)\} = \{\{x\} \setminus \{x\}\}$ yields $\sigma(\mathsf{f}(F, X)) = \{\emptyset\}$.

Consider now the AF $H = (\{a, b\}, \{(a, b)\})$. We get $\sigma(F \sqcup H) = \{\{x, b\}\}$. Again due to universal definedness and condition $e_{\mathbf{sC}}$ we obtain $\sigma(\mathsf{f}(F \sqcup H, X)) = \{\{b\}\}$. Due to l_1 we further infer $\sigma(\mathsf{f}(F, X) \sqcup H)) = \{\{b\}\}$.

This already yields a contradiction in case of grounded semantics since $gr(\mathsf{f}(F, X) \sqcup H)) = \{\{b\}\}$ implies b is unattacked in $\mathsf{f}(F, X) \sqcup H$ which is obviously not true.

For preferred and semi-stable semantics we deduce that $\{b\}$ is admissible in $\mathsf{f}(F, X) \sqcup H$. Hence, the AF $H' = (\{a, b\}, \{(b, a)\})$ must be a subframework of $\mathsf{f}(F, X)$. Moreover, whenever b is attacked by some $c \neq a$ in $\mathsf{f}(F, X)$, it has to be counterattacked by b in $\mathsf{f}(F, X)$ because admissibility of $\{b\}$ in $\mathsf{f}(F, X) \sqcup H$ has to be guarenteed. Consequently, $\{b\} \in ad(\mathsf{f}(F, X))$ is implied too. In case of preferred semantics we infer the existence of a set E, s.t. $\{b\} \subseteq E \in pr(\mathsf{f}(F, X))$. For semi-stable we deduce that either $\{b\}$ is already semi-stable in $\mathsf{f}(F, X)$, or there is an admissible set E, s.t. $\{b\}^{\oplus} \subset E^{\oplus}$ with $E \in ss(\mathsf{f}(F, X))$. Note that $E \neq \emptyset$ is implied. For both semantics, $\sigma(\mathsf{f}(F, X)) \neq \{\emptyset\}$. Contradiction!

Let us turn now to stage semantics. Consider therefore the following two AFs

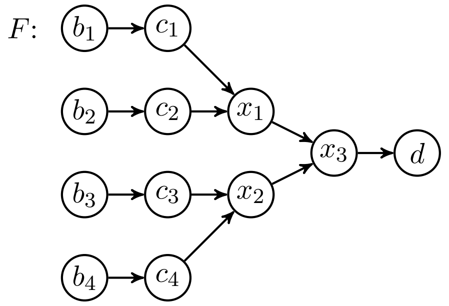
$$\begin{tikzcd} F\colon & b \arrow[r] & x \arrow[r] & c & & H\colon & a \arrow[r] & b \end{tikzcd}$$

with $a \notin A(\mathsf{f}(F, X))$, i.e. a is a fresh argument not appearing in $\mathsf{f}(F, X)$. We have $stg(F) = \{\{b, c\}\}$ and $stg(F \sqcup H) = \{\{a, x\}\}$. Since stage semantics is universally defined we deduce via $e_{\mathbf{sC}}$ that $stg(\mathsf{f}(F, x)) = \{\{b, c\}\}$, and respectively $stg(\mathsf{f}(F \sqcup H, x)) = \{\{a\}\}$. Since c appears in an extension of $\mathsf{f}(F, x)$, we have $(c, c) \notin R(\mathsf{f}(F, x))$, hence $(c, c) \notin R(\mathsf{f}(F, x) \sqcup H)$. Moreover, since we assumed $a \notin A(\mathsf{f}(F, X))$ we deduce $(a, c), (c, a) \notin R(\mathsf{f}(F, x))$. Furthermore, $(a, c), (c, a) \notin R(H)$ as c is not contained in H. This means, $(a, c), (c, a) \notin R(\mathsf{f}(F, x)) \cup R(H) = R(\mathsf{f}(F, x) \sqcup H)$. In other words, $\{a, c\}$ is conflict-free in $\mathsf{f}(F, x) \sqcup H$ and obviously, $\{a\}^{\oplus} \subset \{a, c\}^{\oplus}$. By definition of stage semantics we conclude $\{a\} \notin stg(\mathsf{f}(F, x) \sqcup H)$. Hence, $stg(\mathsf{f}(F, x) \sqcup H) \neq \{\{a\}\} = stg(\mathsf{f}(F \sqcup H, x))$ contradicting l_1.

□

Proposition 12. *The set $\{l_2, e_{\mathbf{sC}}\}$ is satisfiable under stable semantics but not under $\sigma \in \{stg, ss, pr, gr\}$.*

Proof.
- First, the positive result regarding stable semantics. In Proposition 11 we have shown that the set of desiderata $\{l_1, e_{\mathbf{sC}}\}$ is satisfiable under stable semantics. As satisfiability of l_1 w.r.t. a certain instance implies satisfiability of l_2 w.r.t. to the same instance (see Figure 1) the simultaneous satisfiability of the set $\{l_1, e_{\mathbf{sC}}\}$ is shown.
- Let $\sigma \in \{stg, ss, pr, gr\}$. Towards a contradiction suppose that there is a forgetting operator f satisfying l_2 and $e_{\mathbf{sC}}$ under σ. Consider the following AF F and let $X = \{x_1, x_2, x_3\}$.



Obviously, $\sigma(F) = \{\{b_1, b_2, b_3, b_4, x_1, x_2, d\}\}$. Let $\mathsf{f}(F, X)$ be the forgetting result. Let further a_1, a_2, a_3 and a_4 be arguments not contained in $A(\mathsf{f}(F, X))$. For each $1 \leq i \leq 4$ we define $H_i = (\{a_i, b_i\}, \{(a_i, b_i)\})$. Hence,

$\sigma(F \sqcup H_1 \sqcup H_3) = \{\{a_1, b_2, a_3, b_4, c_1, c_3, x_3\}\}$,
$\sigma(F \sqcup H_2 \sqcup H_4) = \{\{b_1, a_2, b_3, a_4, c_2, c_4, x_3\}\}$,
$\sigma(F \sqcup H_1 \sqcup H_2) = \{\{a_1, a_2, b_3, b_4, c_1, c_2, x_2, d\}\}$,
$\sigma(F \sqcup H_3 \sqcup H_4) = \{\{b_1, b_2, a_3, a_4, c_3, c_4, x_1, d\}\}$.

Using the universal definedness of any considered semantics σ together with condition $e_{\mathbf{sC}}$ yields:

$\sigma(\mathsf{f}(F \sqcup H_1 \sqcup H_3, X)) = \{\{a_1, b_2, a_3, b_4, c_1, c_3\}\}$,
$\sigma(\mathsf{f}(F \sqcup H_2 \sqcup H_4, X)) = \{\{b_1, a_2, b_3, a_4, c_2, c_4\}\}$,
$\sigma(\mathsf{f}(F \sqcup H_1 \sqcup H_2, X)) = \{\{a_1, a_2, b_3, b_4, c_1, c_2, d\}\}$,
$\sigma(\mathsf{f}(F \sqcup H_3 \sqcup H_4, X)) = \{\{b_1, b_2, a_3, a_4, c_3, c_4, d\}\}$.

Applying l_2 justifies:

$\sigma(\mathsf{f}(F, X) \sqcup H_1 \sqcup H_3) = \{\{a_1, b_2, a_3, b_4, c_1, c_3\}\}$,
$\sigma(\mathsf{f}(F, X) \sqcup H_2 \sqcup H_4) = \{\{b_1, a_2, b_3, a_4, c_2, c_4\}\}$,
$\sigma(\mathsf{f}(F, X) \sqcup H_1 \sqcup H_2) = \{\{a_1, a_2, b_3, b_4, c_1, c_2, d\}\}$,
$\sigma(\mathsf{f}(F, X) \sqcup H_3 \sqcup H_4) = \{\{b_1, b_2, a_3, a_4, c_3, c_4, d\}\}$.

The last two lines show that any a_i, b_i as well as c_i appears together with d in at least one extension. Consequently, the forgetting result $\mathsf{f}(F, X)$ neither contains attacks between a_i and d, nor b_i and d, nor c_i and d.
Hence, $\{a_1, b_2, a_3, b_4, c_1, c_3\} \in cf(\mathsf{f}(F, X) \sqcup H_1 \sqcup H_3)$, implies $\{a_1, b_2, a_3, b_4, c_1, c_3, d\} \in cf(\mathsf{f}(F, X) \sqcup H_1 \sqcup H_3)$. Moreover, $\{a_1, b_2, a_3, b_4, c_1, c_3\}^{\oplus} \subset \{a_1, b_2, a_3, b_4, c_1, c_3, d\}^{\oplus}$ which contradicts $\{a_1, b_2, a_3, b_4, c_1, c_3\} \in stg(\mathsf{f}(F, X) \sqcup H_1 \sqcup H_3)$.

Let us consider the remaining semantics, i.e. $\sigma \in \{gr, pr, ss\}$. Since $d \notin \{a_1, b_2, a_3, b_4, c_1, c_3\} \in \sigma(\mathsf{f}(F, X) \sqcup H_1 \sqcup H_2)$ as well as $d \notin \{b_1, a_2, b_3, a_4, c_2, c_4\} \in \sigma(\mathsf{f}(F, X) \sqcup H_3 \sqcup H_4)$ we conclude that d must be attacked by an argument $e \notin \{a_1, \dots, a_4, b_1, \dots b_4, c_1, \dots, c_4\} = A$ not being counterattacked by any $a \in A$. If so, we deduce $\{a_1, a_2, b_3, b_4, c_1, c_2, d\} \notin ad(\mathsf{f}(F, X) \sqcup H_1 \sqcup H_2)$ as well as $\{b_1, b_2, a_3, a_4, c_2, c_4, d\} \notin ad(\mathsf{f}(F, X) \sqcup H_3 \sqcup H_4)$. Thus, $\{a_1, a_2, b_3, b_4, c_1, c_2, d\} \notin \sigma(\mathsf{f}(F, X) \sqcup H_1 \sqcup H_2)$ and $\{b_1, b_2, a_3, a_4, c_2, c_4, d\} \notin \sigma(\mathsf{f}(F, X) \sqcup H_3 \sqcup H_4)$. Contradiction!

□

The presented unsatisfiability results reveal intrinsic limitations that hinder the practical implementation of potentially desirable combinations (or even single desiderata). In contrast, satisfiability results demonstrate the theoretical possibility of such implementations. However, a satisfiable set of desiderata does not necessarily uniquely determine a forgetting operator. See Section 6.1 for examples.

Tables 1 and 2 summarise the results proved. There are simply too many combinations to check all combinations exhaustively. From the 25 desiderata introduced, there are 300 distinct combinations of two desiderata, 2300 of three desiderata, and so forth. However, taking into account the dependencies between single desiderata (Propositions 1 and 2) the status of many other combinations become immediately clear. For example, we showed that Desideratum s_3 is satisfiable under stable semantics (Proposition 5). Moreover, any function f satisfying s_3 under stable semantics will also satisfy s_2, s_1 and r_1 (Figure 1). Hence, the two-element sets $\{s_3, s_2\}$, $\{s_3, s_1\}$ and $\{s_3, r_1\}$ are satisfiable too. In contrast, any superset of an unsatisfiable set of combinations will stay unsatisfiable. For instance, since Desideratum e_2 is unsatisfiable under any considered semantics (cf. Proposition 8), we infer that any combination that involves e_2, e.g., $\{e_2, s_2\}$, etc, is unsatisfiable too.

Combination		Semantics	Proposition
$\{d\}$,	$d \in \{e_{3_\supseteq}, e_{3_\subseteq}, e_{\mathbf{sC}}, r_1, r_2, r_4, s_1, s_2, s_3, m_1, m_2, m_4, v_1, v_2, v_3, l_1, l_2, i_1, i_2\}$	all	4+5
$\{d\}$,	$d \in \{r_3, m_3\}$	all, except *stb*	4+6
$\{d\}$,	$d \in \{e_1, e_{\mathbf{wC}}\}$	*gr*, *il*, *eg*	7
$\{e_4\}$		*stb*	9
$\{e_{3_\subseteq}, e_{3_\supseteq}\}$		*gr*, *il*, *eg*	10
$\{e_{\mathbf{sC}}, e_{\mathbf{wC}}\}$		*gr*, *il*, *eg*	10
$\{l_1, e_{\mathbf{sC}}\}$		*stb*	11
$\{l_2, e_{\mathbf{sC}}\}$		*stb*	12

Table 1: Summary of shown Compatibilities of Desiderata

6. Forgetting under Stable Semantics

Let us reflect on the proposed extensional desiderata listed in Section 4. At first we observe that conditions e_1 and e_4 represent two opposing philosophies about the concept of forgetting. Desideratum e_1 requires that any former extension has to survive in an adjusted

Combination		Semantics	Proposition
$\{d\}$	$d \in \{r_3, m_3\}$	*stb*	6
$\{d\}$	$d \in \{e_1, e_{\mathbf{wC}}\}$	*stg*, *stb*, *ss*, *pr*	7
$\{e_2\}$		all	8
$\{e_4\}$		all, except *stb*	9
$\{s_2, l_1, e_{3_\subseteq}\}$		*stg*, *stb*	10
$\{s_2, l_1, e_{3_\supseteq}\}$		*stg*, *stb*	10
$\{e_{3_\subseteq}, e_{3_\supseteq}\}$		*stg*, *stb*, *ss*, *pr*	10
$\{e_{\mathbf{sC}}, e_{\mathbf{wC}}\}$		*stg*, *stb*, *ss*, *pr*	10
$\{l_1, e_{\mathbf{sC}}\}$		*stg*, *ss*, *pr*, *gr*	11
$\{l_2, e_{\mathbf{sC}}\}$		*stg*, *ss*, *pr*, *gr*	12

Table 2: Summary of shown Incompatibilities of Desiderata

fashion, namely new extensions are obtained from initial ones via deleting the arguments that have to be forgotten ($\sigma(\mathsf{f}(F, X)) = \{E \smallsetminus X \mid E \in \sigma(F)\}$). In contrast, desideratum e_4 requires the full removal of extensions containing arguments that have to be forgotten ($\sigma(\mathsf{f}(F, X)) = \sigma(F) \smallsetminus \{E \mid E \in \sigma(F), E \cap X \neq \varnothing\}$). These interpretations of forgetting are independent as shown by Proposition 1. Any other considered extension-based condition is either a relaxation of e_1, or a lifting of this interpretation to the level of strong equivalence. In the following we will consider these two main desiderata in case of the most prominent semantics, namely the stable semantics.

6.1 Forgetting via e_4

In (Baumann et al., 2020, Algorithm 1) the very first operator satisfying e_4 w.r.t. forgetting single arguments was introduced. In the motivation section we observed that applying this operator iteratively does not necessarily produce a desirable outcome and furthermore, the procedure is highly sensitive to the order of forgetting (cf. Example 4). In the following we show how to modify the existing procedure for multiple arguments, thereby also covering the case of forgetting a single argument.

The former construction consists of two steps. Given an AF F and an argument x we first remove x and its associated attacks, i.e. we consider F_x. Any stable extension E of F not containing x remains stable in F_x. However, new extensions may arise. Now, in a second step, each new (undesired) extension E' is removed via adding a self-defeating argument not attacked by E'.

This procedure can be more or less directly applied to forget a whole set of arguments X. In the first step we simply restrict the initial framework to $A(F) \smallsetminus X$, i.e. we consider F_X. Thus, any former extension containing arguments from X is not stable anymore but any other survives. And secondly, we eliminate any unwanted extensions via the addition of self-attacking arguments. This process results in the AF returned by the function $\mathsf{f}_1(F, X)$ whose construction is shown in Construction 1. Proposition 13 shows the desiderata that

$\mathsf{f}_1(F, X)$ satisfies. In order to simplify the definition we assume that the set of forgetting arguments X is a subset of the arguments appearing in the framework.[5]

Construction 1. *Given an AF $F = (A, R)$ and $X \subseteq A$. Let $B = \{a_E \mid E \in stb(F_X) \smallsetminus stb(F)\}$, s.t. $B \cap A = \varnothing$. Then, $\mathsf{f}_1(F, X) = (A', R')$, where:*

$$\begin{aligned} A' &= A(F_X) \cup B, \\ R' &= R(F_X) \cup \{(b, b) \mid b \in B\} \\ &\quad \cup \bigcup_{E \in stb(F_X) \smallsetminus stb(F)} \{(y, a_E) \mid y \in \bigcup stb(F_X) \smallsetminus E\}. \end{aligned}$$

Proposition 13. *Let $G = \mathsf{f}_1(F, X)$. Then G satisfies conditions v_3, s_1, m_4, e_4 under the stable semantics.*

Proof. Condition s_1 is satisfied by G as Construction 1, removes all arguments X, and only adds arguments B, where B and X are disjoint. Moreover, condition v_3 is ensured by the fact that if $A(F)$ and X are disjoint, then $F_X = F$ and $B = \varnothing$. Desideratum m_4 is ensured as first, G does not contain arguments in X anymore and secondly, newly introduced arguments are self-attacking. This means, for any H with $A(H) \subseteq \mathcal{U} \smallsetminus X$ we have: $\bigcup \sigma(\mathsf{f}(F, X) \sqcup H) \subseteq (A(H) \cup A(F)) \smallsetminus X$.

The proof that G also satisfies e_4, i.e., $stb(G) = \{E \mid E \in stb(F), E \cap X = \varnothing\}$ mainly relies on two well-known properties listed below.[6] Let us consider the disjoint union $stb(F) = \mathcal{E}_{\bar{X}} \cup \mathcal{E}_X$, where $\mathcal{E}_{\bar{X}} = \{E \in stb(F) \mid E \cap X = \varnothing\}$ and $\mathcal{E}_X = \{E \in stb(F) \mid E \cap X \neq \varnothing\}$.

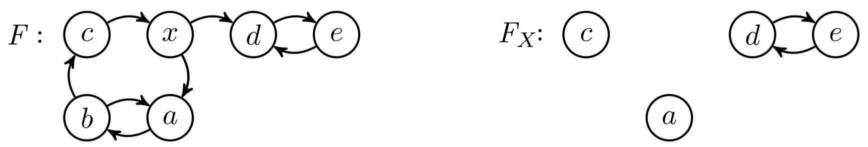
1. Preservation of extensions not containing any $x \in X$. Given an AF F and its restriction $F|_{A(F) \smallsetminus X}$, we have $\mathcal{E}_{\bar{X}} \subseteq stb\left(F|_{A(F) \smallsetminus X}\right)$.

2. Elimination of single extensions. For any H, any fresh argument a, i.e. $a \notin A(H)$, and any $E \in stb(H)$ we have: $stb(H') = stb(H) \smallsetminus \{E\}$ where $A(H') = A(H) \cup \{a\}$ and $R(H') = R(H) \cup \{(y, a) \mid y \in (\bigcup stb(H) \cup \{a\}) \smallsetminus E\}$.

Given these two points, it is easy to see that *i)* any additional occurring stable extension E of F_X, i.e. an E that is not stable in F will be removed (as it does not attack a_E) and *ii)* every stable extension in $\mathcal{E}_{\bar{X}}$ is preserved as a stable extension of G. This is because $\mathcal{E}_{\bar{X}} \subseteq stb(F_X)$ and for every addition of a new argument a_E, there exists an attack from every argument $y \in \bigcup stb(F_X) \smallsetminus E$ into a_E. Consequently, since $stb(F_X)$ forms a $\subsetneq$-antichain it is guaranteed that every extension of $\mathcal{E}_{\bar{X}}$ contains an argument attacking a_E in G. □

Example 8 (Example 4 cont.)**.** *Consider again the AF F from the beginning. We have $stb(F) = \{\{x, b, e\}, \{a, c, d\}, \{a, c, e\}\}$. Let $X = \{x, b\}$. Applying Construction 1 immediately yields $\mathsf{f}_1(F, X) = F_X$ with $stb(F_X) = \{\{a, c, d\}, \{a, c, e\}\}$. This means, there is no need to eliminate unwanted extensions.*

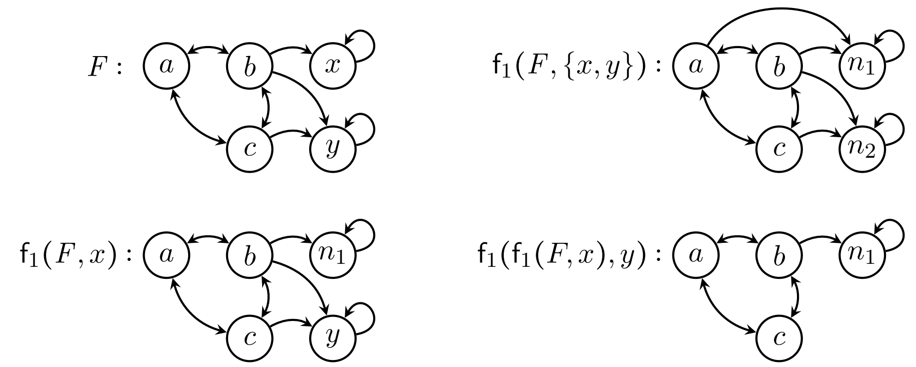
5. If not, simply consider $\mathsf{f}_1(F, X')$ with $X' = X \cap A(F)$. The same applies to Constructions 2 and 3.
6. A procedure to eliminate unwanted stable extensions was first presented in (Dunne et al., 2015) and studied in a principled way in (Baumann & Brewka, 2019).



Please note that $F_X = \mathsf{f}_1(F, \{b, x\}) = \mathsf{f}_1(\mathsf{f}_1(F, b), x)$. *In other words, forgetting* x *and* b *simultaneously yields the more compact outcome of the two potential orders of iterative applications of the operator* f, *i.e.,* $\mathsf{f}_1(\mathsf{f}_1(F, b), x)$ *vs.* $\mathsf{f}_1(\mathsf{f}_1(F, x), b)$ *(cf. Example 4).*

Unfortunately, the above observation does not hold in general. This means, there are cases where iterative forgetting is strictly better than set forgetting. Consider therefore the following illustrating example.

Example 9. *Consider the AF* F *and let* $X = \{x, y\}$. *We have one stable extension, namely* $stb(F) = \{\{b\}\}$. *Furthermore, the initial framework modulo the forgotten arguments has three, i.e.* $stb\left(F_{\{x,y\}}\right) = \{\{a\}, \{b\}, \{c\}\}$. *Hence, the result of simultaneously forgetting* x *and* y, *i.e. the AF* $\mathsf{f}_1(F, \{x, y\})$, *contains two new self-attackers* n_1 *and* n_2, *to eliminate the stable extensions* $\{a\}$ *and* $\{c\}$.



On the other hand, the initial framework modulo x *has two stable extensions,* $stb\left(F_{\{x\}}\right) = \{\{b\}, \{c\}\}$, *so only one argument* n_1 *is introduced in* $\mathsf{f}_1(F, x)$ *to eliminate* $\{c\}$. *This result modulo* y *already has the correct stable models,* $stb\left(\mathsf{f}_1(F, x)_{\{y\}}\right) = \{\{b\}\}$, *therefore no more self-attacker are introduced.*

Now lets turn to the question of whether e_4 can be satisfied when removing arguments is impossible, unintended or just undesired for some reason. This means, only expanding the initial framework F is allowed. The AF returned by the function $\mathsf{f}_2(F, X)$ in Construction 2 gives an affirmative answer. The simple idea is to manipulate F as follows. For any argument $x \in X$ we add a self-attacking argument a_x to F which is attacked by each attacker of x. In doing so, any former extension E not containing x remains stable, but former extensions E' with $x \in E'$ do not survive as they do not attack a_x. It is important to note that the presented construction relies on syntactic information only as it does not require the computation of any extension.

Construction 2. *Given an AF $F = (A, R)$ and $X \subseteq A$. Let $X' = \{a_x \mid x \in X\}$, s.t. $X' \cap A = \varnothing$. Then, $\mathsf{f}_2 := (A', R')$, where:*

$$\begin{aligned} A' &= A \cup X', \\ R' &= R \cup \{(x', x') \mid x' \in X'\} \\ &\quad \cup \bigcup_{x \in X} \{(b, a_x) \mid b \in A \setminus X, (b, x) \in R\}. \end{aligned}$$

Proposition 14. *Let $G = \mathsf{f}_2(F, X)$. Then G satisfies conditions v_3 and e_4 under the stable semantics.*

Proof. Condition v_3 is ensured by the Construction 2, as if $X \cap A(F) = \varnothing$ no new arguments, nor attacks are added.

Now, e_4 holds, since:

1. Former extensions not overlapping X are preserved. Given a stable extension E of F with $E \cap X = \varnothing$. First, conflict-freeness in F immediately transfer to conflict-freeness in G. Secondly, by definition of stable semantics E attacks all $a \in A(F) \setminus E$. In particular, E attacks any $x \in X$. Thus, by construction it also attacks any $x' \in X'$ proving stability in G.

2. Former extensions overlapping X are eliminated. Given a stable extension E of F with $E \cap X \neq \varnothing$. Let $x \in E \cap X$ and $x' \in X'$ the associated new argument. By construction we have E attacks x' in G if and only if E attacks x in F contradicting its conflict-freeness in F.

3. Finally, no new extension $E \in stb(G) \setminus stb(F)$ is created. This can be seen as follows: Newly introduced arguments are self-attacking. This means, $E \subseteq A(F)$. As E attacks any argument in $A(G) \setminus E$ it also attacks any argument in $A(F) \setminus E$. Thus, $E \in stb(F)$ contradicting the assumption and concluding the proof.

□

Construction 3 is a very simple syntactical manipulation. Here, instead of adding new arguments which are self-attacking and attacked by certain former arguments, we add self-loops to any x occurring in F. The downside of this simple implementation is that in contrast to both former constructions, Construction 3 modifies former relations between initial arguments which might be undesired in certain applications.

Construction 3. *Given an AF $F = (A, R)$ and $X \subseteq A$. Then, $\mathsf{f}_3 := (A', R')$, where:*

$$\begin{aligned} A' &= A, \\ R' &= R \cup \{(x, x) \mid x \in X\}. \end{aligned}$$

Proposition 15. *Let $G = \mathsf{f}_3(F, X)$. Then G satisfies conditions v_3, m_4 and e_4 under the stable semantics.*

Proof. Condition v_3 is ensured by the fact that the construction does not do anything if $A(F) \cap X = \emptyset$. Desideratum m_4 is ensured as first, G does not contain newly introduced arguments and secondly, arguments in X are self-attacking.
Now, e_4 holds, since

1. Former extensions not overlapping X are preserved. Given a stable extension E of F with $E \cap X = \emptyset$. Consequently, E remains conflict-free in G and moreover, as E attacks any argument $a \in A(F) \setminus E$ we deduce that E is stable in G since $A(G) = A(F)$.

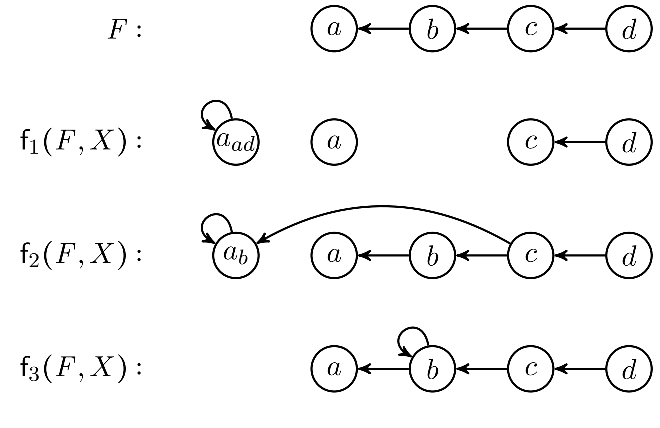
2. Former extensions overlapping X are eliminated. Given a stable extension E of F with $E \cap X \neq \emptyset$. Let $x \in E \cap X$. Hence, x is self-defeating in G and thus, E is not conflict-free and therefore not stable in G.

3. No new extension is created. If $E \in stb(G) \setminus stb(F)$, then E is conflict-free in G. Since $A(G) = A(F)$ and arguments in $A(F) \cap X$ became self-attacking we deduce $E \subseteq A(F) \setminus X$. Moreover, by definition of stable semantics E attacks any argument in $A(F) \setminus E$. Consequently, $E \in stb(F)$ contradicting the assumption.

□

Finally, we provide a simple example illustrating the differences between the three constructions presented. Future work will include fine-tuning these constructions and conducting experiments on runtimes.

Example 10 (Example 5 cont.). *Consider again AF F. We have* $stb(F) = \{\{b, d\}\}$. *Let* $X = \{b\}$. *Note that Desideratum* e_4 *forces the collapse of stable extensions. Indeed, neither* $\mathsf{f}_1(F, X)$, $\mathsf{f}_2(F, X)$, *nor* $\mathsf{f}_3(F, X)$ *provide reasonable positions.*



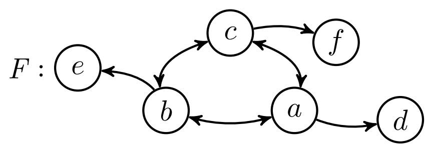
6.2 Forgetting via e_1

The main reason for the impossibility to find an operator satisfying e_1 under the stable semantics is an intrinsic one, namely realisability. More precisely, certain instances, i.e. an initial framework F and a set X of arguments would enforce a framework F' with a set of stable extensions violating the $\subseteq$-antichain property or tightness. Consequently, a reasonable strategy is to look for forgetting operators satisfying e_1 whenever possible, or satisfying a relaxation of the condition, when it is not.

Regarding possibility it would be interesting to find semantical or syntactical conditions for $stb(F)$ or F, respectively, s.t. $\{E \smallsetminus X \mid E \in stb(F)\}$ is realisable. This means, how to guarantee that $\{E \smallsetminus X \mid E \in stb(F)\}$ is tight and forms a $\subseteq$-antichain. Natural candidates for the relaxation of e_1 would be $e_{3_\subseteq}$, $e_{3_\supseteq}$ or $e_{\mathbf{sC}}$. A similar procedure was suggested and also implemented for *strong persistence* in the realm of logic programming (Gonçalves, Knorr, Leite, & Woltran, 2017).

The question which relaxation to choose has no clear answer. First of all, according to Figure 1, $e_{3_\subseteq}$, $e_{3_\supseteq}$ and $e_{\mathbf{sC}}$ are all independent of each other. Moreover, each relaxation has its particular advantages and drawbacks. It does not make sense to express general preferences among the desiderata as the specific application will determine which criterion is most suitable. Furthermore, even for a particular chosen relaxation, the precise result of forgetting might not be clear. This is demonstrated by the following example.

Example 11. *Consider again AF F presented in Example 6:*



Let $X = \{a, b, c\}$. As $stb(F) = \{\{a, e, f\}, \{b, f, d\}, \{c, d, e\}\}$ we obtain $\{E \smallsetminus X \mid E \in stb(F)\} = \{\{d, e\}, \{d, f\}, \{e, f\}\}$. This set is not tight implying that e_1 is impossible. The relaxation $e_{\mathbf{sC}}$ is satisfied by any AF F' with $stb(F') \subseteq \{\{d, e\}, \{d, f\}, \{e, f\}\}$. Giving up one of these three extensions would result in a realisable set. However, without further information there is no reason to prefer one set over the other.

In summary, that means both the choice of how to relax e_1 as well as its particular implementation depends on the application in mind. A more thorough study on this issue including certain preferences is left for future work.

7. Discussion and Conclusion

This article presents an in-depth analysis of forgetting in the realm of abstract argumentation. We considered 25 desirable syntactical and/or semantical properties of possible forgetting operators for a representative set of semantics, including stage, stable, semi-stable, preferred, grounded, ideal, and eager semantics. We excluded other semantics, such as weak admissibility-based semantics (Baumann, Brewka, & Ulbricht, 2020), because their precise characterisation of realisability is still lacking, and complete semantics, because the high complexity of its characterisation (Linsbichler, 2018). In our analysis, we included conditions well-known from logic programming, such as strong persistence and strong invariance (Knorr & Alferes, 2014; Gonçalves et al., 2016a). We would like to note that we did not express any preferences among the given desiderata. We believe that specific applications will favour certain conditions while excluding others.

One main axis of the article is the theoretical analysis of the given desiderata in terms of individual and combined satisfiability. While individual conditions can mostly be easily satisfied, it quickly becomes difficult or even impossible to find operators that fulfil multiple conditions. Moreover, we considered the question whether simultaneous or iterative forgetting always yield a more compact result. Unfortunately, it turns out that no clear statement can be made, as sometimes simultaneous and sometimes iterative forgetting yields a better result. A further analysis of this topic, e.g. identifying conditions which guarantee more compact forgetting results, will be part of future work. Moreover, we showed that already established forgetting operators from logic programming (Berthold et al., 2019) cannot be unconditionally applied to abstract argumentation. The two main reasons are that we cannot translate back the output LP as an AF in a straightforward manner, and secondly as well as more importantly, due to essential differences in the expressibility between both formalisms (Eiter et al., 2013; Dunne et al., 2015). However, we want to emphasise that there may still be instances where the retranslation works perfectly, making former operators potentially useful. A deeper investigation of this issue will be part of our future work.

A second contribution of this article is a more practical one, namely providing concrete constructions and discussing potential forgetting operators for selected conditions. In particular, we focused on the prominent stable semantics (leaving others for future work) and the central extensionality desiderata e_1 and e_4. Desideratum e_1 requires that any former extension must be cleaned of the arguments to be forgotten. In contrast, desideratum e_4 requires the full removal of extensions (cf. (Baumann & Brewka, 2019) for an axiomatic approach) containing arguments that have to be forgotten. We provided three concrete constructions satisfying e_4. Of particular note is that although the criterion is a semantical one, two of the constructions are purely syntactical in nature. We discussed desideratum e_1 only as realisability issues prevent the existence of general operators. We further showed that a lot of non-technical decisions have to be made even if we stick to conditional satisfiability. This underscores that external factors will play a significant role in determining the selection of desiderata.

Our approach follows the line of most formalisms for belief change in which there is an implicit assumption that the primary objective of the belief change is known. For example, in belief revision operations, the belief revising the old belief set is given. Indeed, this is also the case in the multiple belief change operations (see (Rodrigues et al., 2011) for an overview), originally introduced by Furhmann (Fuhrmann, 1988), where the belief set is modified in response to a *set* of formulas, e.g., explanation-based belief revision (Falappa, Kern-Isberner, & Simari, 2002), set revision (Zhang, 1996), package contraction (Fuhrmann & ove Hansson, 1994), etc. However, it is worth mentioning that an important consideration is that of *which* arguments trigger the forgetting operation. In general, one can mention the motivations or external reasons for an agent to want to forget certain arguments. Here are some possibilities:

- Persuasion in Debates: Arguments that an agent want to be defeated in the debate of a given topic.
- Changed Context: Arguments that are no longer applicable because the context has evolved.
- Shifted Values: Arguments that have become irrelevant due to changes in ethical or moral standards.

- Laws and Regulations: Arguments that are obsolete due to updates in legal requirements, such as the recent change in the minimum required passage width from 2.80 metres to 3.05 metres (German law concerning the respective interests of neighbours).

A future line of research is to include specific conditions with respect to the status of arguments that are *not* accepted, i.e., whether we insist on a forgotten argument to be defeated (i.e., attacked by an accepted argument) or whether we are content with its status being merely "undecided" (i.e., not attacked by an accepted argument). Such information about the specific status of all arguments is more readily available in the so-called *labelling-based approach* to argumentation semantics (Baroni et al., 2018). Please note that although the extension-based versions and labelling-based ones are intimately connected, as for most of them we have a 1-to-1 correspondence, intrinsic properties like strong equivalence (Baumann, 2016) or realisability (Baumann & Heine, 2023, 2024) often differ significantly. Hence, we believe that analysing forgetting desiderata fine-tuned to the specific desired status of forgotten arguments will be a challenging task.

In recent times the argumentation community realised that the limited expressive capability of AFs, namely the option of single attacks only, reduces their suitability as sound target systems for more complex applications (Atkinson, Baroni, Giacomin, Hunter, Prakken, Reed, Simari, Thimm, & Villata, 2017). Consequently, a number of additional functionalities were introduced including preferences, values, collective attacks, attacks on attacks as well as support relations between arguments (Amgoud & Vesic, 2011; Bench-Capon & Atkinson, 2009; Nielsen & Parsons, 2006; Baroni, Cerutti, Giacomin, & Guida, 2009; Cayrol & Lagasquie-Schiex, 2009). *Abstract Dialectical Frameworks* (ADFs) are one of the most powerful generalisations of Dung AFs, yet staying on the abstract level (Brewka & Woltran, 2010; Brewka, Polberg, & Woltran, 2014). The additional expressive power is achieved by adding acceptance conditions to the arguments which allow for the specification of more complex relationships between them. Of particular interest might be the subclass of *bipolar ADFs* (BADFs) which are as complex as AFs while arguably offering more modelling capabilities (Brewka, Ellmauthaler, Strass, Wallner, & Woltran, 2017; Straß & Wallner, 2015; Baumann & Heinrich, 2023). It is one highly relevant future task to investigate notions of forgetting in these more expressive argumentation formalisms.

Similarly, it is worthwhile to study forgetting in logic-based or structured argumentation formalisms (cf. (Besnard & Hunter, 2008; Arieli, Borg, Heyninck, & Straßer, 2021) for excellent overviews). These approaches take the (logical) structure of arguments into account and define notions such as attack, undercut, and defensibility, among others, in terms of the (logical) properties of the chosen argument structures. A first study in this direction was recently conducted for *assumption-based argumentation* (ABA) (Bondarenko, Toni, & Kowalski, 1993). As expected, advancing research from abstract to structured argumentation is a challenging endeavour, as the underlying formalisms may restrict the types of changes possible at the abstract level (Prakken, 2023; Wallner, 2020). Another emerging issue in this context is the need to study expressibility specifically tailored to the considered formalism (Berthold, Rapberger, & Ulbricht, 2023a, 2023b).

One highly relevant work is (Rienstra, Sakama, van der Torre, & Liao, 2020) dealing with so-called *robustness* principles. The paper studies the question to which extent old labellings persist/new labellings arise if a certain change of the initial AF is performed. Such results are highly relevant for the theory of forgetting as they can be used to show the satisfiability/unsatisfiability of desired properties. Similarly, the analysis of *implicit conflicts*, i.e. arguments that do not occur jointly in any extension, albeit there is no attack between them, as well as *compact AFs*, i.e. frameworks where each argument is credulously accepted, may be used for more fine-grained/optimised constructions (Baumann, Dvorák, Linsbichler, Spanring, Strass, & Woltran, 2016).

Within the scope of belief change operations (already mentioned above), the study of belief contraction operations — which aim to remove a belief from a belief set — is very relevant (Alchourrón et al., 1985; Gärdenfors, 1988; Gabbay, Rodrigues, & Russo, 2010; Rodrigues et al., 2011). Contraction operations were originally defined for single formulas, but later extended to *sets* of formulas too (Fuhrmann & ove Hansson, 1994; Falappa et al., 2002; Zhang, 1996; Zhang, Chen, Zhu, & Chen, 1997; Zhang & Foo, 2001) in the spirit of the analysis done in this paper. There is a potentially interesting area of investigation of the relationship between postulates for these operations and the desiderata we proposed here.

Finally, towards a more general operation of forgetting, not necessarily restricted to the realm of abstract argumentation: Forgetting a set X can also be combined with protecting a set Y. This might be important as Y may guarantee that the overall behaviour of the considered system is not affected. For instance, in AGM belief contraction (Alchourrón et al., 1985), the Y part is *implicit*. Contractions and revisions must protect tautologies and all the logical consequences of whatever is not removed during the process to ensure that the set remains closed under logical consequence. In contrast, in the framework of *revision by translation* (Gabbay, Rodrigues, & Russo, 2000), the translation machinery was codified as formulas and included in the belief set. These formulas needed to be protected from removal during revisions. Clearly, our focus was on the removal aspect and we did not considered Y explicitly. Nevertheless some desiderata implicitly touch this topic like the vacuity desiderata as well as some syntactical desiderata. We believe that forgetting combined with protecting will be one central topic in the future.

Acknowledgements

The authors acknowledge the financial support by the Federal Ministry of Education and Research of Germany and by Sächsische Staatsministerium für Wissenschaft, Kultur und Tourismus in the programme Center of Excellence for AI-research “Center for Scalable Data Analytics and Artificial Intelligence Dresden/Leipzig”, project identification number: ScaDS.AI

References

Alchourrón, C. E., Gärdenfors, P., & Makinson, D. (1985). On the logic of theory change: Partial meet contraction and revision functions. *The Journal of Symbolic Logic*, *50*(2), 510—-530.

Amgoud, L., & Vesic, S. (2011). A new approach for preference-based argumentation frameworks. *Annals of Mathematics and Artificial Intelligence*, 149–183.

Arieli, O., Borg, A., Heyninck, J., & Straßer, C. (2021). Logic-based approaches to formal argumentation. *IfCoLog Journal of Logics and their Applications*, *8*(6), 1793–1898.

Atkinson, K., Baroni, P., Giacomin, M., Hunter, A., Prakken, H., Reed, C., Simari, G. R., Thimm, M., & Villata, S. (2017). Towards artificial argumentation. *AI Magazine*, 25–36.

Baroni, P., Gabbay, D., Giacomin, M., & van der Torre, L. (2018). *Handbook of Formal Argumentation*. College Publications.

Baroni, P., Caminada, M., & Giacomin, M. (2011). An introduction to argumentation semantics. *The Knowledge Engineering Review*, *26*(4), 365–410.

Baroni, P., Caminada, M., & Giacomin, M. (2018). Abstract argumentation frameworks and their semantics. In *Handbook of Formal Argumentation*, pp. 159–236. College Publications.

Baroni, P., Cerutti, F., Giacomin, M., & Guida, G. (2009). Encompassing attacks to attacks in abstract argumentation frameworks. In *Proceedings of (ECSQARU-09)*, pp. 83–94.

Baroni, P., & Giacomin, M. (2007). On principle-based evaluation of extension-based argumentation semantics. *Artificial Intelligence*, *171*, 675–700.

Baumann, R. (2012). Normal and strong expansion equivalence for argumentation frameworks. *Artificial Intelligence*, *193*, 18—-44.

Baumann, R. (2016). Characterizing equivalence notions for labelling-based semantics. In *Proceedings of (KR-16)*, pp. 22–32.

Baumann, R. (2018). On the nature of argumentation semantics: Existence and uniqueness, expressibility, and replaceability. In *Handbook of Formal Argumentation*. College Publications. also appears in IfCoLog Journal of Logics and their Applications.

Baumann, R., & Berthold, M. (2022). Limits and possibilities of forgetting in abstract argumentation. In *Proceedings of (IJCAI-22)*, pp. 2539–2545.

Baumann, R., & Brewka, G. (2019). Extension removal in abstract argumentation - an axiomatic approach. In *Proceedings of (AAAI-19)*, Vol. 33, pp. 2670–2677.

Baumann, R., Brewka, G., & Ulbricht, M. (2020). Comparing weak admissibility semantics to their dung-style counterparts - reduct, modularization, and strong equivalence in abstract argumentation. In *Proceedings of (KR-20)*, pp. 79–88.

Baumann, R., Dvorák, W., Linsbichler, T., Spanring, C., Strass, H., & Woltran, S. (2016). On rejected arguments and implicit conflicts: The hidden power of argumentation semantics. *Artificial Intelligence*, *241*, 244–284.

Baumann, R., Gabbay, D. M., & Rodrigues, O. (2020). Forgetting an argument. In *Proceedings of (AAAI-20)*, Vol. 34, pp. 2750–2757.

Baumann, R., & Heine, A. (2023). On conflict-free labellings - realizability, construction and patterns of redundancy. In *Proceedings of (KR-23)*, pp. 720–725.

Baumann, R., & Heine, A. (2024). On naive labellings - realizability, construction and patterns of redundancy. In *Proceedings of (FoIKS-24)*, pp. 125–143.

Baumann, R., & Heinrich, M. (2023). Bipolar abstract dialectical frameworks are covered by kleene's three-valued logic. In *Proceedings of (IJCAI-23)*, pp. 3123–3131.

Baumann, R., & Spanring, C. (2015). Infinite argumentation frameworks - On the existence and uniqueness of extensions. In *Essays Dedicated to Gerhard Brewka on the Occasion of His 60th Birthday*, Vol. 9060, pp. 281–295.

Bench-Capon, T. J. M., & Atkinson, K. (2009). Abstract argumentation and values. In *Argumentation in Artificial Intelligence*, pp. 45–64.

Berthold, M. (2022). On syntactic forgetting with strong persistence. In *Proceedings of (KR-22)*, pp. 43–52.

Berthold, M., Gonçalves, R., Knorr, M., & Leite, J. (2019). A syntactic operator for forgetting that satisfies strong persistence. *Theory and Practice of Logic Programming*, *19*(5–6), 1038–1055.

Berthold, M., Rapberger, A., & Ulbricht, M. (2023a). Forgetting aspects in assumption-based argumentation. In *Proceedings of (KR-23)*, pp. 86–96.

Berthold, M., Rapberger, A., & Ulbricht, M. (2023b). On the expressive power of assumption-based argumentation. In *Proceedings of (JELIA-23)*, pp. 145–160.

Besnard, P., & Hunter, A. (2008). *Elements of Argumentation.*

Bisquert, P., Cayrol, C., de Saint-Cyr, F. D., & Lagasquie-Schiex, M. (2011). Change in argumentation systems: Exploring the interest of removing an argument. In *Proceedings of (SUM-11)*, pp. 275–288.

Bondarenko, A., Toni, F., & Kowalski, R. A. (1993). An assumption-based framework for non-monotonic reasoning. In *Proceedings of (LPNMR-93).*

Brewka, G., Ellmauthaler, S., Strass, H., Wallner, J. P., & Woltran, S. (2017). Abstract dialectical frameworks. an overview. *The IfCoLog Journal of Logics and their Applications*, *4*(8), 2263–2317.

Brewka, G., Polberg, S., & Woltran, S. (2014). Generalizations of dung frameworks and their role in formal argumentation. *IEEE Intelligent Systems*, *29*(1), 30–38.

Brewka, G., & Woltran, S. (2010). Abstract dialectical frameworks. In *Proceedings of (KR-10).*

Caminada, M. (2007). Comparing two unique extension semantics for formal argumentation: Ideal and eager. In *Proceedings of (BNAIC-07).*

Cayrol, C., & Lagasquie-Schiex, M. (2009). Bipolar abstract argumentation systems. In *Argumentation in Artificial Intelligence*, pp. 65–84.

Delgrande, J. P. (2017). A knowledge level account of forgetting. *Journal of Artificial Intelligence Research*, *60*, 1165–1213.

Dung, P. M. (1995). On the acceptability of arguments and its fundamental role in non-monotonic reasoning, logic programming and n-person games. *Artificial Intelligence*, *77*(2), 321–357.

Dung, P. M., Mancarella, P., & Toni, F. (2007). Computing ideal sceptical argumentation. *Artificial Intelligence*, *171*(10), 642–674.

Dunne, P. E., Dvořák, W., Linsbichler, T., & Woltran, S. (2015). Characteristics of multiple viewpoints in abstract argumentation. *Artificial Intelligence*, *228*, 153–178.

Eiter, T., Fink, M., Pührer, J., Tompits, H., & Woltran, S. (2013). Model-based recasting in answer-set programming. *Journal of Applied Non-Classical Logics*, *23*(1-2), 75–104.

Eiter, T., & Kern-Isberner, G. (2019). A brief survey on forgetting from a knowledge representation and reasoning perspective. *KI - Künstliche Intelligenz*, 9–33.

Falappa, M. A., Kern-Isberner, G., & Simari, G. R. (2002). Explanations, belief revision and defeasible reasoning. *Artificial Intelligence*, *141*(1), 1–28.

Fuhrmann, A. (1988). *Relevant logics, modal logics and theory change*. Ph.D. thesis, Australian National University.

Fuhrmann, A., & ove Hansson, S. (1994). A survey of multiple contractions. *Journal of Logic, Language, and Information*, *3*(1), 39–75.

Gabbay, D., Rodrigues, O., & Russo, A. (2010). *Revision, Acceptability and Context: Theoretic and Algorithmic Aspects*. Springer Verlag.

Gabbay, D., Rodrigues, O., & Russo, A. (2000). *Revision by Translation*, pp. 3–31. Springer US.

Gärdenfors, P. (1988). *Knowledge in Flux. Modelling the Dymanics of Epistemic States*. Cambridge, Mass.: MIT Press.

Gonçalves, R., Knorr, M., & Leite, J. (2016a). The ultimate guide to forgetting in answer set programming. In *Proceedings of (KR-16)*, pp. 135–144. AAAI Press.

Gonçalves, R., Knorr, M., & Leite, J. (2016b). You can't always forget what you want: On the limits of forgetting in answer set programming. In *Proceedings of (ECAI-16)*, pp. 957–965.

Gonçalves, R., Knorr, M., Leite, J., & Woltran, S. (2017). When you must forget: Beyond strong persistence when forgetting in answer set programming. *Theory and Practice of Logic Programming*, *17*(5–6), 837–854.

Knorr, M., & Alferes, J. J. (2014). Preserving strong equivalence while forgetting. In *Proceedings of (JELIA-14)*, Vol. 8761, pp. 412–425.

Lang, J., Liberatore, P., & Marquis, P. (2003). Propositional independence: Formula-variable independence and forgetting. *Journal of Artificial Intelligence Research*, *18*(1), 391—-443.

Lifschitz, V., Pearce, D., & Valverde, A. (2001). Strongly equivalent logic programs. *ACM Transactions on Computational Logic*, *2*(4), 526—-541.

Lifschitz, V., Tang, L. R., & Turner, H. (1999). Nested expressions in logic programs. *Annals of Mathematics and Artificial Intelligence*, *25*, 369–389.

Lin, F., & Reiter, R. (1994). Forget it!. In *Proceedings of AAAI Fall Symposium on Relevance*, pp. 154–159.

Linsbichler, T. (2018). Characteristics of multiple viewpoints in abstract argumentation under complete semantics. Tech. rep. DBAI-TR-2018-113, TU Wien.

Nielsen, S. H., & Parsons, S. (2006). A generalization of dung's abstract framework for argumentation: Arguing with sets of attacking arguments. In *Proceedings of (ArgMAS-06)*, pp. 54–73.

Oikarinen, E., & Woltran, S. (2011). Characterizing strong equivalence for argumentation frameworks. *Artificial Intelligence*, *175*(14), 1985–2009.

Prakken, H. (2023). Relating abstract and structured accounts of argumentation dynamics: the case of expansions. In *Proceedings of (KR-23)*, pp. 562–571.

Rienstra, T., Sakama, C., van der Torre, L., & Liao, B. (2020). A principle-based robustness analysis of admissibility-based argumentation semantics. *Argument & Computation*, *11*(3), 305–339.

Rodrigues, O., Gabbay, D., & Russo, A. (2011). Belief revision. In *Handbook of Philosophical Logic*, Vol. 16, pp. 1–114. Springer, Dordrecht.

Strass, H. (2013). Approximating operators and semantics for abstract dialectical frameworks. *Artificial Intelligence*, *205*, 39–70.

Straß, H., & Wallner, J. P. (2015). Analyzing the computational complexity of abstract dialectical frameworks via approximation fixpoint theory. *Artificial Intelligence*, *226*.

van der Torre, L., & Vesic, S. (2017). The principle-based approach to abstract argumentation semantics. *IfCoLog Journal of Logics and their Applications*, 629–639.

Wallner, J. P. (2020). Structural constraints for dynamic operators in abstract argumentation. *Argument & Computation*, *11*(1–2), 151–190.

Zhang, D. (1996). Belief revision by sets of sentences. *Journal of Computer Science and Technology*, *11*, 108–125.

Zhang, D., Chen, S., Zhu, W., & Chen, Z. (1997). Representation theorems for multiple belief changes. In *Proceedings of (IJCAI-97)*, Vol. 1.

Zhang, D., & Foo, N. (2001). Infinitary belief revision. *Journal of Philosophical Logic*, *30*, 525–570.