Limits and Possibilities of Forgetting in Abstract Argumentation

Ringo Baumann and **Matti Berthold**
Leipzig University, Department of Computer Science
{baumann,berthold}@informatik.uni-leipzig.de

Abstract

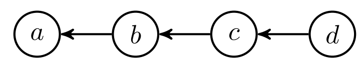
The topic of *forgetting* has been extensively studied in the field of knowledge representation and reasoning for many major formalisms. Quite recently it has been introduced to abstract argumentation. However, many already known as well as essential aspects about forgetting like *strong persistence* or *strong invariance* have been left unconsidered. Moreover, we show that forgetting in abstract argumentation cannot be reduced to forgetting in logic programming. In addition, we deal with the more general problem of forgetting whole sets of arguments and show that iterative application of existing operators for single arguments does not necessarily yield a desirable result as it may not produce an informationally economic argumentation framework. As a consequence we provide a systematic and exhaustive study of forgetting desiderata and associated operations adapted to the intrinsics of abstract argumentation. We show the limits and shed light on the possibilities.

1 Introduction

The notion of *forgetting* has been extensively studied in the field of knowledge representation and reasoning for many major formalisms like classical logic (Lin and Reiter 1994), logic programming (Gonçalves, Knorr, and Leite 2016a; Eiter and Kern-Isberner 2018) and more recently for abstract argumentation (Baumann, Gabbay, and Rodrigues 2020). Roughly speaking, forgetting is about getting rid of some variables, atoms or arguments while keeping as much as possible of the reasoning not concerned with the forgotten. The ability of forgetting is often exploited to make reasoning more efficient. In this paper we want to further elaborate the limits and possibilities of forgetting in abstract argumentation. The latter is a vibrant research area in AI (Simari and Rahwan 2009; Baroni et al. 2018) with Dung-style argumentation frameworks (AFs) and their associated semantics at the heart of this field (Dung 1995) .

In order to obtain reasonable forgetting operators for abstract reasoning we may try to convey ideas from other formalisms. The area of logic programming with its plenty of approaches to forgetting is a good candidate (cf. (Gonçalves, Knorr, and 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 1 (Limits of the Standard Translation)**.** *Consider the following AF F. We observe $stb(F) = \{\{b,d\}\}$. Assume now that we want to forget the argument b. Note that simply deleting b would yield an AF F_b, s.t. $stb(F_b) = \{\{a,d\}\}$. This means, such a syntactical removal would render the previously unaccepted argument a acceptable.*

$F:$


Let us consider instead the standard translation from AFs to LPs (Strass 2013). This yields the following equivalent logic program P.

$$P: \quad a \leftarrow not\, b \qquad b \leftarrow not\, c \qquad c \leftarrow not\, d \qquad d$$

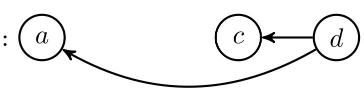
Now we may apply the already defined forgetting operator f_{SP} (Berthold et al. 2019b). More precisely, forgetting b from P results in $\mathsf{f}_{SP}(P,b)$ as given below.

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

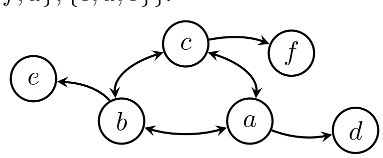
Unfortunalety, $\mathsf{f}_{SP}(P,b)$ is a non-AF-like program. Therefore, it is generally not possible to simply reverse the standard translation. However, in case of $\mathsf{f}_{SP}(P,b)$ we may find an equivalent LP P' which is indeed AF-like.

$$P': \quad a \leftarrow not\, d \qquad c \leftarrow not\, d \qquad d$$

Retranslating P' to the realm of AFs results in F'. Note that $stb(F) = \{\{d\}\}$ as desired.

$F':$


Example 2 (Representational Limits)**.** *Consider now the slightly more involved AF F. We observe $stb(F) = \{\{a,e,f\},\{b,f,d\},\{c,d,e\}\}$.*

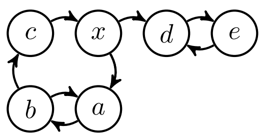
$F:$


Let us assume again that we want to forget the argument b. The favored forgetting result is thus the extension set $D = \{\{a,e,f\},\{f,d\},\{c,d,e\}\}$. Since D forms a ⊆-antichain there is an LP P realizing it (Eiter et al. 2013). However, we will never find an equivalent AF-like LP P' since D does not satisfies so-called tightness (Dunne et al. 2015). In particular, $\{f,d\} \cup \{e\} \notin D$ but $\{e,f\} \subseteq \{a,e,f\}$ and $\{d,e\} \subseteq \{c,d,e\}$.

The final example deals with already existing forgetting operators in abstract argumentation. It shows that forgetting multiple arguments cannot be simply reduced to forgetting single arguments as this does not necessarily yield desirable results.

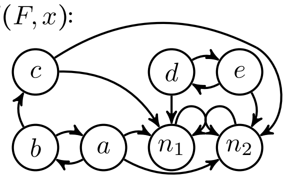
Example 3 (Forgetting Sets vs. Arguments)**.** *Consider the following AF F. We have $stb(F) = \{\{x,b,e\},\{a,c,d\},\{a,c,e\}\}$. Assume that we want to forget a set of arguments, say $\{x,b\}$. One reasonable forgetting result is thus an AF F', s.t. $stb(F') = \{\{a,c,d\},\{a,c,e\}\} = D$.*

F:

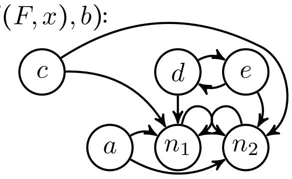


One natural approach for forgetting multiple arguments is to iteratively apply an existing forgetting operator for single arguments. The following frameworks illustrate this procedure for the operator f *firstly presented in (Baumann, Gabbay, and Rodrigues 2020, Algorithm 1, Example 4).*

$\mathsf{f}(F,x)$:

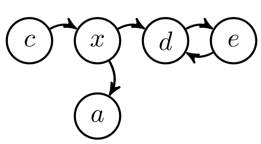


$\mathsf{f}(\mathsf{f}(F,x),b)$:

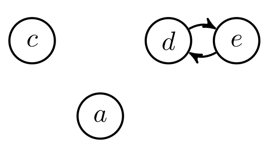


If forgetting b *first and subsequently* x *reveals that this approach is sensitive to the order of forgetting and might not yield an informationally economic result.*

$\mathsf{f}(F,b)$:



$\mathsf{f}(\mathsf{f}(F,b),x)$:



The three examples above show that we need further investigation on how sets of arguments can be forgotten in case of AFs. As a consequence we provide a systematic and comprehensive analysis of forgetting desiderata and associated operations adapted to the intrinsics of abstract argumentation. We hereby draw a lot of inspiration from logic programming. We show the limits and shed light on the possibilities. In particular, we study the relations between desiderata, their individual as well as combined satisfiability and look for promising combinations. Moreover, we consider forgetting under stable semantics as it shows a quite different behaviour regarding the fulfillment of combined desiderata. Finally, we conclude and discuss related work.

2 Background

Logic Programming

Syntax and Semantics We assume a *propositional signature* $\mathcal{U}$. A *logic program* P over $\mathcal{U}$ (Lifschitz, Tang, and Turner 1999) is a finite set of *rules* of the form $a_1 \vee \ldots \vee a_k \leftarrow b_1, ..., b_l,\ not\, c_1, ..., not\, c_m,\ not\, not\, d_1, ..., not\, not\, d_n$. 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 $P^I = \{H(r) \leftarrow B^+(r) \mid r \in P, B^-(r) \cap I = \varnothing, B^{--}(r) \subseteq I\}$. An interpretation I is an *answer set* of P if $I \models P$, and for each interpretation I' we have: If $I' \models P^I$, 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, and Valverde 2001). Given a set $V \subseteq \mathcal{U}$, the V*-exclusion* of a set of answer sets $\mathcal{M}$, denoted $\mathcal{M}_{\parallel V}$, is $\{X \backslash V \mid X \in \mathcal{M}\}$.

Forgetting: Desiderata and Operators Let $\mathcal{P}$ be the set of all logic programs. A *forgetting operator* is a (partial) function $\mathsf{f} : \mathcal{P} \times 2^{\mathcal{U}} \to \mathcal{P}$ with $(P, V) \mapsto \mathsf{f}(P, V)$. The program $\mathsf{f}(P, V)$ is interpreted as the *result of forgetting about* V *from* P. Moreover, $\mathcal{U}(\mathsf{f}(P, V)) \subseteq \mathcal{U}(P) \smallsetminus V$ is usually required. In the following we introduce some well-known properties for forgetting operators (Gonçalves, Knorr, and Leite 2016a).

Strong persistence is presumbly the best known one (Knorr and Alferes 2014). It requires that the result of forgetting $\mathsf{f}(P, V)$ 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: $\mathcal{AS}(\mathsf{f}(P, V) \cup R) = \mathcal{AS}(P \cup R)_{\parallel V}$ for all programs R with $\mathcal{U}(R) \subseteq \mathcal{U} \backslash V$.

Strong invariance requires that rules not mentioning atoms to be forgotten can be added before or after forgetting.

(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 $\mathcal{U}(R) \subseteq \mathcal{U} \backslash V$.

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: $\mathcal{AS}(\mathsf{f}(P, V)) = \mathcal{AS}(P)_{\parallel V}$.

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

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

Note that the presented desiderata are often considered for certain subclasses like *disjunctive*, *normal* or *Horn programs*. Sometimes forgetting properties are also considered relativized to concrete forgetting instances (Berthold et al. 2019b).

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 of 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 $F = (B, S)$, we use $A(F)$ to refer to the set B and $R(F)$ to refer to the relation S. 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, the restriction of an AF F to a set of arguments $C \subseteq \mathcal{U}$ is defined as $F|_C = (A(F) \cap C, R(F) \cap (C \times C))$.

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 semantics include stage (stg), stable (stb), semi-stable (ss), complete (co), preferred (pr), grounded (gr), ideal (il) and eager (eg). The requirements of each semantics are summarised below. A recent overview of argumentation semantics can be found in (Baroni, Caminada, and 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 any $\mathcal{I} \in co(F)$, $E \subseteq \mathcal{I}$,*
9. *$E \in il(F)$ iff $E \in co(F)$, $E \subseteq \bigcap pr(F)$ and there is no $\mathcal{I} \in co(F)$ satisfying $\mathcal{I} \subseteq \bigcap pr(F)$ s.t. $E \subset \mathcal{I}$,*
10. *$E \in eg(F)$ iff $E \in co(F)$, $E \subseteq \bigcap ss(F)$ and there is no $\mathcal{I} \in co(F)$ 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 stable semantics all considered semantics are universally defined. The grounded, ideal and eager semantics are uniquely defined (cf. (Baumann and Spanring 2015) for an overview).

With respect to the acceptability of arguments, we consider the 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 *skeptically accepted w.r.t.* σ if $\sigma(F) \neq \emptyset$ and $a \in \bigcap \sigma(F)$. In case of stable semantics a collapse is possible, i.e. $stb(F) = \emptyset$ for some F. From basic set theory we know that in this case all arguments $x \in A(F)$ are skeptically accepted. To get around this cornercase we redefine the intersection over the empty extension set as $\bigcap stb(F) = \emptyset$.

We say that a set of sets $\mathcal{E} \subseteq 2^{\mathcal{U}}$ is *realizable w.r.t. a semantics* σ if there is an AF F s.t. $\sigma(F) = \mathcal{E}$. Realizability under stable semantics is given if and only if i) $\mathcal{E}$ forms a $\subseteq$-antichain[1] and ii) $\mathcal{E}$ is *tight* (Dunne et al. 2015). Tightness if 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. $(a, e) \notin \{(b, c) \mid \exists E' \in \mathcal{E} : \{b, c\} \subseteq E'\}$. See Example 2 for an illustration. 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 Desiderata for Forgetting

Given an AF F and a set of arguments $X \subseteq \mathcal{U}$, we use $\mathsf{f}_\sigma(F, X)$ to denote the *result of forgetting the arguments X in F under semantics σ*. This means, we consider a function $\mathsf{f}_\sigma : \mathcal{F} \times 2^{\mathcal{U}} \to \mathcal{F}$ mapping a pair (F, X) to an AF $\mathsf{f}_\sigma(F, X)$. If clear from context or irrelevant we will omit σ.

In the following we collect and define a large number of desiderata for forgetting in abstract argumentation. Some of them have been already considered in (Baumann, Gabbay, and Rodrigues 2020) for the case of single arguments. We generalize them to sets of arguments as done in the LP case. Moreover we introduce further important conditions firstly considered in the realm of LPs (Gonçalves, Knorr, and Leite 2016a). We will see that there are many dependencies that are not clear at first glance. Note that desiderata e_1 as well as e_2 could be alternatively renamed as $e_{\mathbf{CP}}$ and $e_{\mathbf{SP}}$ (see Section 2 for more details.) However, we decided to keep in line with the notation chosen in (Baumann, Gabbay, and Rodrigues 2020).

Desiderata 1. *Given an AF F and a set of arguments $X \subseteq \mathcal{U}$. For a forgetting operator f we define:*

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 such 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 $E \mapsto T(E) \supseteq E \setminus X$* *(supersets of X-adjusted extension)*

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

[1] Within the argumentation community this property is usually referred to as *I-maximality* (Baroni and Giacomin 2007).

The next four desiderata are concerned with skeptical and credulous reasoning.

Desiderata 2. *Given an AF F and a set of arguments $X \subseteq \mathcal{U}$. For a forgetting operator* f *we define:*

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

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

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

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

Arguably the presented reasoning desiderata either describe to strictly or too loosely what is skeptically or credulously accepted. For r_1 and r_2 to be satisfied, it suffices to syntactically remove X. 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 desiderata, that bridge semantic and syntatic requirements.

Desiderata 3. *Given two AFs F and H as well as a set of arguments $X \subseteq \mathcal{U}$. For a forgetting operator* f *we define:*

m_1. $\bigcap \sigma(\mathsf{f}(F, X)) \subseteq A(F) \setminus X$
(skept. acceptance is among unforgotten old arguments)

m_2. $\bigcup \sigma(\mathsf{f}(F, X)) \subseteq A(F) \setminus X$
(cred. acceptance is among unforgotten old arguments)

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

Condition m_1 (resp. m_2) requires that, if there are new arguments added while forgetting, they be irrelevant to skeptical (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.

The following three conditions are purely syntactical ones. Desideratum s_1 makes explicit what is often implicitly assumed for forgetting operators in other formalisms. Condition s_3 presents the most straightforward way of forgetting a set of arguments. Such an syntactical approach was firstly considered in (Bisquert et al. 2011).

Desiderata 4. *Given an AF F and a set of arguments $X \subseteq \mathcal{U}$. For a forgetting operator* f *we define:*

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

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

s_3. $\mathsf{f}(F, X) = F|_{A(F) \setminus X}$ *(rigid AF)*

The following vacuity desiderata provide conditions under which a given framework does not require any changes.

Desiderata 5. *Given an AF F and a set of arguments $X \subseteq \mathcal{U}$. For a forgetting operator* f *we define:*

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

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

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

When deriving a forgetting result it would be advantageous to be able to confine the construction in some way. For comparison, some forgetting operators in LP 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 $(\mathbf{SI})$. Similarly, when forgetting arguments from an AF we could require that arguments that do not stand in (close) contact to the arguments to be forgotten can be left unchanged.

Desiderata 6. *Given an AF F and a set of arguments $X \subseteq \mathcal{U}$. For a forgetting operator* f *we define:*

l_0. $\mathsf{f}(F, X) \sqcup H \equiv \mathsf{f}(F \sqcup H, X)$ *for all AFs H with $A(H) \subseteq \mathcal{U} \setminus X$* *($\mathsf{f}$ and $\sqcup$ are compatible)*

l_1. $\mathsf{f}(F, X) \sqcup H \equiv \mathsf{f}(F \sqcup H, X)$ *for all AFs H with $A(H) \subseteq \mathcal{U} \setminus (X \cup \{a \mid \exists x \in X,$ s.t. $(a, x) \in R$ or $(x, a) \in R\})$*
(less tolerant refinement of compatibility)

We proceed with an analysis of their dependencies.

Proposition 1. *For $\sigma \in \{stg, stb, ss, pr, gr, il, eg\}$ and conditions c and c' in the diagram below, a path from c to c' indicates that any function f_σ satisfying c under σ also satisfies c' under σ. Moreover, only these relations hold.*

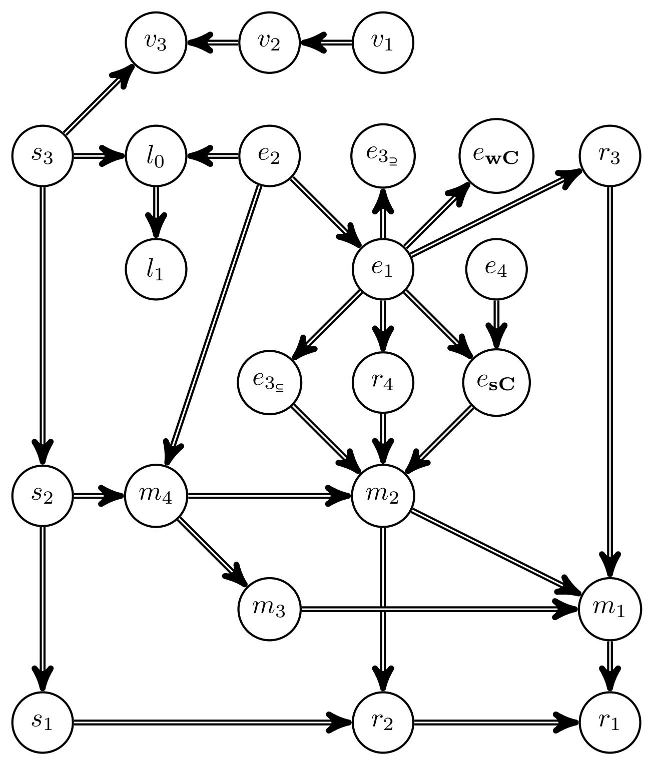


Figure 1: Dependencies

Proof: In the following we show all valid relations.

- $e_1 \Rightarrow r_3$: $\bigcap \sigma(\mathsf{f}(F, X)) = \bigcap \{E \setminus X \mid E \in \sigma(F)\} = (\bigcap \{E \mid E \in \sigma(F)\}) \setminus X = (\bigcap \sigma(F)) \setminus X$
- $e_1 \Rightarrow r_4$: $\bigcup \sigma(\mathsf{f}(F, X)) = \bigcup \{E \setminus X \mid E \in \sigma(F)\} = (\bigcup \{E \mid E \in \sigma(F)\}) \setminus X = (\bigcup \sigma(F)) \setminus X$
- $e_1 \Rightarrow e_{\mathbf{sC}}, e_{\mathbf{wC}}$: 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 $\sigma(F) \setminus \{E \mid E \in \sigma(F), E \cap X \neq \varnothing\} \subseteq \{E \setminus X \mid E \in \sigma(F)\}$.
- $e_{\mathbf{sC}} \Rightarrow m_2$: $\bigcup \sigma(\mathsf{f}(F,X)) \subseteq \bigcup\{E \setminus X \mid E \in \sigma(F)\} \subseteq A(F) \setminus X$ since for each extension $E \in \sigma(F)$, $E \subseteq A(F)$ is implied.
- $e_2 \Rightarrow e_1$: Consider $H_\varnothing = (\varnothing, \varnothing)$.
- $e_2 \Rightarrow m_4$: Let H be an AF with $A(H) \subseteq \mathcal{U} \setminus X$. We have: $\bigcup \sigma(\mathsf{f}(F,X) \sqcup H) = \bigcup\{E \setminus X \mid E \in \sigma(F \sqcup H)\} \subseteq (A(H) \cup A(F)) \setminus X$.
- $e_2 \Rightarrow l_0$: $\sigma(\mathsf{f}(F,X) \sqcup H) = \{E \setminus X \mid E \in \sigma(F \sqcup H)\} = \{E \setminus X \mid E \in \sigma(F \sqcup H \sqcup \varnothing)\} = \sigma(\mathsf{f}(F \sqcup H, X) \sqcup \varnothing) = \sigma(\mathsf{f}(F \sqcup H, X))$
- $l_0 \Rightarrow l_1$: 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 $H_\varnothing$.
- $m_4 \Rightarrow m_3$: $\bigcap \sigma(\mathsf{f}(F,X) \sqcup H) \subseteq \bigcup \sigma(\mathsf{f}(F,X) \sqcup H) \subseteq (A(H) \cup A(F)) \setminus X$
- $m_2 \Rightarrow m_1$: $\bigcap \sigma(\mathsf{f}(F,X)) \subseteq \bigcup \sigma(\mathsf{f}(F,X)) \subseteq A(F) \setminus X$
- $m_1 \Rightarrow r_1$ and $m_2 \Rightarrow r_2$: Obvious since $(A(F) \setminus X) \cap X = \varnothing$.
- $s_3 \Rightarrow s_2$: $A(\mathsf{f}(F,X)) = A(F|_{A(F) \setminus X}) = A(F) \setminus X$.
- $s_3 \Rightarrow v_3$: Let $A(F) \cap X = \varnothing$. Consequently, $\mathsf{f}(F,X) = F|_{A(F) \setminus X} = F$.
- $s_3 \Rightarrow l_0$: 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 $\mathsf{f}(F,X) \sqcup H \equiv \mathsf{f}(F \sqcup H, X)$.
- $s_2 \Rightarrow s_1$: $A(\mathsf{f}(F,X)) \cap X = (A(F) \setminus X) \cap X = \varnothing$.
- $s_2 \Rightarrow m_4$: $\bigcup \sigma(\mathsf{f}(F,X) \sqcup H) \subseteq A(\mathsf{f}(F,X) \sqcup H) = A(\mathsf{f}(F,X)) \cup A(H) = (A(F) \setminus X) \cup A(H) = (A(H) \cup A(F)) \setminus X$ since for the AF H we have $A(H) \subseteq \mathcal{U} \setminus X$.
- $r_2 \Rightarrow r_1$: Obviously, $\bigcup \sigma(\mathsf{f}(F,X)) \cap X = \varnothing \Rightarrow \bigcup \sigma(\mathsf{f}(F,X)) \cap X = \varnothing$.
- $s_1 \Rightarrow r_2$: $\bigcup \sigma(\mathsf{f}(F,X)) \cap X \subseteq A(\mathsf{f}(F,X)) \cap X = \varnothing$.
- $v_1 \Rightarrow v_2$: $\bigcup \sigma(F) \cap X = \varnothing \Rightarrow \bigcap \sigma(F) \cap X = \varnothing \Rightarrow F = \mathsf{f}(F,X)$.
- $v_2 \Rightarrow v_3$: $A(F) \cap X = \varnothing \Rightarrow \bigcup \sigma(F) \cap X = \varnothing \Rightarrow F = \mathsf{f}(F,X)$.

In order to argue that the remaining relations do not hold we have to provide counter examples. Due to the limited space we provide two illustrating examples only. The remaining non-relations can be shown in a similar fashion.

- $e_4 \not\Rightarrow e_1$: Towards a contradiction suppose $e_4 \Rightarrow e_1$. Consider the AF $F = (\{a,b\}, \varnothing)$ and $X = \{b\}$. For any considered semantics σ we have, $\sigma(F) = \{\{a,b\}\}$. Let f be a forgetting operator satisfying e_4. Thus, $\sigma(\mathsf{f}(F,X)) = \sigma(F) \setminus \{E \mid E \in \sigma(F), E \cap X \neq \varnothing\} = \varnothing$. On the other hand, since f satisfies e_1 too we derive, $\sigma(\mathsf{f}(F,X)) = \{E \setminus X \mid E \in \sigma(F)\} = \{\{a\}\}$. Contradiction.
- $e_4 \not\Rightarrow s_1$: Consider a forgetting operator f satisfying e_4, i.e. $\sigma(\mathsf{f}(F,X)) = \sigma(F) \setminus \{E \mid E \in \sigma(F), E \cap X \neq \varnothing\}$. Assume that s_1 is satisfied, i.e. $A(\mathsf{f}(F,X)) \cap X = \varnothing$. Pick an argument $x \in X$ and define a new operator g, s.t. $A(\mathsf{g}(F,X)) = A(\mathsf{f}(F,X)) \cup \{x\}$ and $R(\mathsf{g}(F,X)) = R(\mathsf{f}(F,X)) \cup \{(z,x) \mid z \in A(\mathsf{g}(F,X))\}$. Obviously, e_4 is still satisfied by g since $\sigma(\mathsf{f}(F,X)) = \sigma(\mathsf{g}(F,X))$ by construction, but s_1 is not.

■

Apart from the relationships concerning single conditions there are more complex implications. In the realm logic programming it was already shown that $(\mathbf{SP})$ is necessary and sufficient for $(\mathbf{SI})$ and $(\mathbf{CP})$ (Gonçalves, Knorr, and Leite 2016a). Beside other interesting relations we state the analogous result for abstract argumentation in Item 5 of the following proposition. Please note that the proof is astonishingly simple.

Proposition 2. *For any semantics $\sigma \in \{stg, stb\}$:*

1. s_2, l_0 *and* $e_{3_\subseteq}$ *imply* e_2,
2. s_2, l_0 *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_0 *if and only if* e_2.

Proof:

1. Let f satisfies s_2, l_0 and $e_{3_\subseteq}$. In order to show desideratum e_2 we will first prove that condition e_1 is implied. Then, applying Statement 5 of this Proposition yields e_2. Given an AF F and a set of arguments $X \subseteq \mathcal{U}$. Desideratum e_1 requires $\sigma(\mathsf{f}(F,X)) = \{E \setminus X \mid E \in \sigma(F)\}$.

   Due to s_2 we have $A(\mathsf{f}(F,X)) = A(F) \setminus X := A$. Define a copy of A, i.e. a set of fresh arguments $A' = \{a' \mid a \in A\}$, s.t. $A' \cap (A(F) \cup X) = \varnothing$. 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, $E \cup \{a' \in A' \mid a \in A \setminus E\} \in \sigma(F \sqcup H)$. Vice versa, any $E' \in \sigma(F \sqcup H)$ can be uniquely associated with some $E \in \sigma(F)$, s.t. $E' = E \cup \{a' \in A' \mid a \in A \setminus E\}$. 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_0 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 $E \setminus X \in \sigma(\mathsf{f}(F,X))$.
     Since $E \in \sigma(F)$ we deduce $E \cup \{a' \in A' \mid a \in A \setminus E\} \in \sigma(F \sqcup H)$. Due to $e_{3_\subseteq}$ we obtain $T(E \cup \{a' \in A' \mid a \in A \setminus E\}) \subseteq (E \cup \{a' \in A' \mid a \in A \setminus E\}) \setminus X = E \setminus X \cup \{a' \in$

$A' \mid a \in A \smallsetminus E\}$ 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_0 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 satisfies 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 satisfies $e_{3_\supset}$ and $e_{3_\subset}$. Furthermore, let F be an AF and $X \subseteq \mathcal{U}$ a set of arguments. Consider now a certain extension $E \in \tau(F)$. Due to $e_{3_\supset}$ and $e_{3_\subset}$ 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 satifies 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.
   ($\Rightarrow$) $\tau(\mathsf{f}(F, X) \sqcup H) =^{(l_0)} \tau(\mathsf{f}(F \sqcup H, X)) =^{(e_1)} \{E \smallsetminus X \mid E \in \tau(F \sqcup H)\}$

■

4 Satisfiability and Unsatisfiability

In this section we consider the satisfiability of single conditions as well as whole sets of desiderata. Most of the results underline the intrinsic limits of forgetting in abstract argumentation as they prove unsatisfiability.

Individual Desiderata

We start with a positive result regarding individual satisfiablity. In fact, 19 conditions are satisfiable under any considered semantics if considered in isolation.

Proposition 3. *Desideratum* $d \in \{e_{3_\supseteq}, e_{3_\subseteq}, e_{\mathbf{sC}}, r_1, r_2, r_3, r_4, s_1, s_2, s_3, m_1, m_2, m_3, m_4, v_1, v_2, v_3, l_0, l_1\}$ *is satisfiable under any semantics* $\sigma \in \{stg, stb, ss, pr, gr, il, eg\}$.

Proof: In the following we will only show that each desideratum $d \in \{e_{3_\supseteq}, e_{3_\subseteq}, e_{\mathbf{sC}}, r_3, r_4, v_1\}$ is satisfiable. The satisfiability of the remaining desiderata is implied by Proposition 1.
Given an AF F and a set of arguments X.

- $e_{3_\supseteq}$: If $\sigma(F) = \varnothing$, then set $\mathsf{f}(F, X) = F$. If not, define $\mathsf{f}(F, X) = (\bigcup_{E \in \sigma(F)} E \smallsetminus X, \varnothing)$. Consequently, $\sigma(\mathsf{f}(F, X)) = \{\bigcup_{E \in \sigma(F)} E \smallsetminus X\}$. Thus, the constant function $T : \sigma(F) \to 2^{\mathcal{U}}$ with $E \mapsto T(E) = \bigcup_{E \in \sigma(F)} E \smallsetminus X$ satisfies $T(E) \supseteq E \smallsetminus X$ 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 $T(E) \subseteq E \smallsetminus X$ 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 $\mathsf{f}(F, X) = (E \smallsetminus X, \varnothing)$. Obviously, $\sigma(\mathsf{f}(F, X)) = \{E \smallsetminus X\}$ justifying $\sigma(\mathsf{f}(F, X)) \subseteq \{E \smallsetminus X \mid E \in \sigma(F)\}$.
- r_3: Consider $\mathsf{f}(F, X) = ((\bigcap \sigma(F)) \smallsetminus X, \varnothing)$.
- r_4: Define $\mathsf{f}(F, X) = ((\bigcup \sigma(F)) \smallsetminus X, \varnothing)$.
- v_1: Set $\mathsf{f}(F, X) = F$.

■

The following proposition shows a dividing line between uniquely and universally defined semantics. The subset antichain property of the latter family prevent the satisfiability of e_1 and $e_{\mathbf{wC}}$.

Proposition 4. *Desiderata* e_1 *and* $e_{\mathbf{wC}}$ *are satisfiable under any* $\tau \in \{gr, il, eg\}$, *but not under* $\sigma \in \{stb, stg, ss, pr\}$.

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 universally defined semantics σ and let $F = (\{a, x\}, \{(a, x), (x, a)\})$ be an 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. ■

The following two propositions are mainly due to already shown results in (Baumann, Gabbay, and Rodrigues 2020).

Proposition 5. *Desiderata* e_4 *is satisfiable under stable semantics, but not under any* $\tau \in \{stg, ss, pr, gr, il, eg\}$.

Proof: Consider stable semantics. In (Baumann, Gabbay, and Rodrigues 2020, Algorithm 1, Example 4) an operator f was introduced able to precisely remove a stable extension, whenever it contains a certain argument x. Since we consider finite AFs and thus, finite forgetting sets X we obtain a new operator satisfying e_4 by simply applying f iteratively. Note that forgetting result is sensitive to the order of forgetting (cf. Example 3 for an illustration). Therefore, a predefined order is essential.
The impossibility of satisfying e_4 under τ was already shown for singletons (Baumann, Gabbay, and Rodrigues 2020, Proposition 5). Thus, it does not work for arbitrary sets either.
■

Proposition 6. *Desiderata* e_2 *is unsatisfiable under any semantics* $\sigma \in \{stg, stb, ss, pr, gr, il, eg\}$.

Proof: The desideratum e_2 was shown to be unsatisfiable when forgetting single arguments only (Baumann, Gabbay, and Rodrigues 2020, Proposition 6). Hence, unsatisfiability for arbitrary sets is implied. ■

Combined Desiderata

In the following we consider the satisfiability of whole sets of conditions. Most of the results underline the intrinsic limits of forgetting in abstract argumentation.

Proposition 7. *We have the following satisfiability results:*

1. $\{s_2, l_0, e_{3_\subseteq}\}$ *as well as* $\{s_2, l_0, e_{3_\supseteq}\}$ *are unsatisfiable for any semantics* $\mu \in \{stg, stb\}$.
2. *Moreover,* $\{e_{3_\subseteq}, e_{3_\supseteq}\}$ *and* $\{e_{\mathbf{sC}}, e_{\mathbf{wC}}\}$ *are unsatisfiable for any semantics* $\sigma \in \{stg, stb, ss, pr\}$*, but satisfiable for each* $\tau \in \{gr, il, eg\}$.

Proof:

1. Let $\mu \in \{stg, stb\}$. According to Items 1 and 2 of Proposition 2 we have that $\{s_2, l_0, e_{3_\subseteq}\}$ as well as $\{s_2, l_0, e_{3_\supseteq}\}$ imply e_2. The latter is unsatisfiable due to Proposition 6. Hence, both sets are unsatisfiable under 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 $\{e_{\mathbf{sC}}, e_{\mathbf{wC}}\}$ imply e_1 (Items 3 and 4 of Proposition 2). Hence, both are unsatisfiable under σ, but satisfiable under τ (cf. Proposition 4).

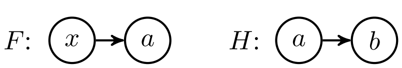
■

The next two results show that stable semantics is somehow exceptional regarding its potential for forgetting.

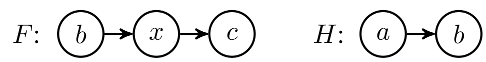
Proposition 8. *The set* $\{l_0, e_{\mathbf{sC}}\}$ *is satisfiable under stable semantics but not under* $\sigma \in \{gr, stg, ss, pr\}$.

Proof:

- The set of desiderata $\{l_0, e_{\mathbf{sC}}\}$ is satisfied under stable semantics by setting $\mathsf{f}(F, X) = (A(F), R(F) \setminus \{(a, x) \mid x \in X\} \cup \{(x, x) \mid x \in X \cap A(F)\})$.
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_0 is fulfilled since $\sigma(\mathsf{f}(F, X) \sqcup H) = \sigma(\mathsf{f}(F \sqcup H, X))$ is implied for any semantics σ.
Moreover, 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_0 and $e_{\mathbf{sC}}$. Consider the following AFs F and H.



For any semantics $\sigma \in \{gr, ss, pr\}$ we have $\sigma(F) = \{\{x\}\}$ as well as $\sigma(F \sqcup H) = \{\{x, b\}\}$. Since all considered semantics are universally defined we deduce $\sigma(\mathsf{f}(F, X)) \neq \emptyset$ for any set of arguments X. Let $X = \{x\}$. Applying condition $e_{\mathbf{sC}}$, i.e. $\sigma(\mathsf{f}(F, X)) \subseteq \{E \setminus X \mid E \in \sigma(F)\}$ yields $\sigma(\mathsf{f}(F, X)) = \{\emptyset\}$ and $\sigma(\mathsf{f}(F \sqcup H, x)) = \{\{b\}\}$, respectively. Due to l_0 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



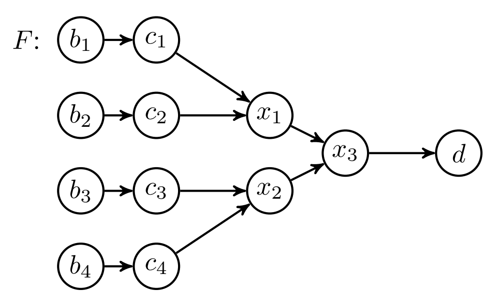
where $a \notin A(\mathsf{f}(F, X))$, i.e. a is an argument that does not appear in the forgetting result $\mathsf{f}(F, X)$. We have $stg(F) = \{\{b, c\}\}$ and $stg(F \sqcup H) = \{\{a, x\}\}$. Since $stg(F)$ is non-empty for any AF, through use of $e_{\mathbf{sC}}$ we can deduce that $stg(\mathsf{f}(F, x)) = \{\{b, c\}\}$, 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 $(c, c) \notin R(\mathsf{f}(F, x) \sqcup H)$. Also $(a, c), (c, a) \notin R(\mathsf{f}(F, x) \sqcup H)$.
Then $\{a, c\}$ is conflict-free and hence, $\{a\}^\oplus \subset \{a, c\}^\oplus$, resulting in $\{a\} \notin stg(\mathsf{f}(F, x) \sqcup H)$. Then $stg(\mathsf{f}(F, x) \sqcup H) \neq stg(\mathsf{f}(F \sqcup H, x)) = \{\{a\}\}$ contradicting l_0.

■

Proposition 9. *The set* $\{l_1, e_{\mathbf{sC}}\}$ *is satisfiable under stable semantics but not under* $\sigma \in \{gr, stg, ss, pr\}$.

Proof:

- The set of desiderata $\{l_1, e_{\mathbf{sC}}\}$ is satisfiable under stable semantics as it is implied by the satisfiable set of conditions $\{l_0, e_{\mathbf{sC}}\}$ (cf. Figure 1 and Proposition 8).
- Let $\sigma \in \{gr, stg, ss, pr\}$. Towards a contradiction suppose that there is a forgetting operator f satisfying l_1 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,
$\sigma(F \sqcup H_1 \sqcup H_3) = \{\{a_1, b_2, a_3, b_4, c_1, c_3, x_3\}\}$,
$\sigma(F \sqcup H_2 \sqcup H_4) = \{\{b_1, a_2, b_3, a_4, c_2, c_4, x_3\}\}$,
$\sigma(F \sqcup H_1 \sqcup H_2) = \{\{a_1, a_2, b_3, b_4, c_1, c_2, x_2, d\}\}$, and
$\sigma(F \sqcup H_3 \sqcup H_4) = \{\{b_1, b_2, a_3, a_4, c_3, c_4, x_1, d\}\}$.

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_1 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!

■

Testing the Limits: Promising Combinations

Proposition 10 shows the compatibility of promising combinations of a semantical and syntactical condition. The strongest syntactical desideratum s_3 is incompatible with all considered semantical conditions and can only be trivially combined with r_1 and r_2. In this context, *trivial* means, that already one condition implies the other as shown in Proposition 1. Stable semantics is the only considered semantics able to collapse for certain AFs. This unique property is reflected in its different behaviour regarding the fulfillment of combined desiderata (see Figure 2).

Proposition 10. *Figure 2 summarizes the compatibility under semantics $\sigma \in \{stg, stb, ss, pr, gr, il, eg\}$. A "✓"/"×" in cell (l,c) indicates whether or not the conditions in line l and column c are simultaneously satisfiable under σ. The symbol "τ" restricts the satisfiability to the semantics gr, il and eg, the symbol "stb" to stable semantics and the symbol "$\neg$stb" to all semantics but stable respectively. The combinations in a dark background are trivial .*

	s_1	s_2	s_3	m_1	m_2	m_3	m_4
r_1	✓	✓	✓	✓	✓	✓	✓
r_2	✓	✓	✓	✓	✓	✓	✓
r_3	✓	✓	×	✓	✓	✓	✓
r_4	✓	✓	×	✓	✓	✓	✓
e_1	τ	τ	×	τ	τ	τ	τ
e_2	×	×	×	×	×	×	×
$e_{3_\subseteq}$	✓	$\neg stb$	×	✓	✓	✓	✓
$e_{3_\supseteq}$	✓	$\neg stb$	×	✓	✓	✓	✓
e_4	stb	×	×	stb	stb	stb	stb
$e_{\mathbf{sC}}$	✓	$\neg stb$	×	✓	✓	✓	✓
$e_{\mathbf{wC}}$	τ	τ	×	τ	τ	τ	τ

Figure 2: Compatibility of syntactical/semantical conditions

Proof: The proof involves 77 combinations of desiderata which has to be checked with respect to 7 semantics. It takes more than 3 pages and is omitted here due to the limited space. ■

5 Forgetting under Stable Semantics

Let us reflect on the proposed extension-based conditions as listed in Desidirata 1. At first we observe that 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 fashion, namely new extensions are obtained from initial ones via deleting the arguments which has to be forgotten ($\sigma(\mathsf{f}(F, X)) = \{E \setminus X \mid E \in \sigma(F)\}$). In contrast, Condition e_4 requires to delete any extension carrying arguments which has to be forgotten ($\sigma(\mathsf{f}(F, X)) = \sigma(F) \setminus \{E \mid E \in \sigma(F), E \cap X \neq \emptyset\}$). Both interpretations of forgetting are independent as shown in 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 more detail. We restrict ourselves to stable semantics and left the consideration of other semantics for future work.

Forgetting via e_4 Quite recently, an e_4-operator f for forgetting single arguments was presented (Baumann, Gabbay, and Rodrigues 2020, Algorithm 1). In Example 3 of the introductory part we have seen that applying this operator f iteratively does not necessarily produce a desirable outcome.

Moreover, this procedure is sensitive to the order of forgetting. How to adapt the existing procedures for multiple arguments?

The former construction consists of two steps. Given an AF F and an argument x. First, remove x and its related attacks, i.e. consider $F_x \coloneqq F|_{A(F)\setminus\{x\}}$. 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'.

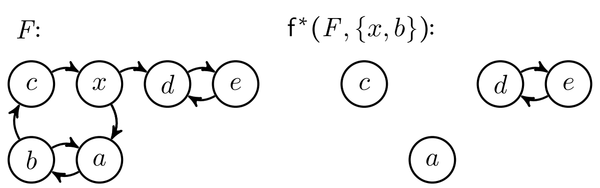
This procedure can be more or less directly applied to forget a whole set of arguments X. In the first step we simply restrict the initial framework to $A(F) \setminus X$, i.e. we consider $F_{|A(F)\setminus X}$. Thus, any former extension containing arguments from X is not stable anymore but any other survives. And secondly, we eliminate any unwanted extensions via the addition of self-attacking arguments. This yields the following algorithm.

Algorithm 1: Construct $G = \mathsf{f}^*(F, X)$

Input : AF F; arguments $X \subseteq \mathcal{U}$
Output : AF G satisfying $\{s_1, e_4, v_3\}$

Function `compute_G(F,X)`
  if $X \cap A(F) = \emptyset$ **then** $G \leftarrow F$;
  else
    $G_0 \leftarrow F|_{A(F)\setminus X}$;
    $A = A(G_0)$; $R \leftarrow R(G_0)$;
    foreach $E_i \in stb(G_0) \setminus stb(F)$ **do**
      Let a_i be a fresh argument s.t. $a_i \notin A(F) \cup A$;
      $A \leftarrow A \cup \{a_i\}$; $R \leftarrow R \cup \{(a_i, a_i)\}$
      foreach $y \in \bigcup stb(G_0) \setminus E_i$ **do**
        $R \leftarrow R \cup \{(y, a_i)\}$;
    $G \leftarrow (A, R)$;
  return G;

Example 4 (Example 3 cont.)**.** *Consider again AF F. We have $stb(F) = \{\{x,b,e\},\{a,c,d\},\{a,c,e\}\}$. Let $X = \{x,b\}$. Applying Algorithm 1 immediately yields $\mathsf{f}^*(F,X) = F|_{A(F)\setminus X}$ as $stb(F|_{A(F)\setminus X}) = \{\{a,c,d\},\{a,c,e\}\}$.*



The attentive reader may have already noticed that $\mathsf{f}^*(F,\{x,b\}) = \mathsf{f}(\mathsf{f}(F,b),x)$. This means, forgetting x and b simultaneously yields the more compact outcome if applying the former operator f iteratively ($\mathsf{f}(\mathsf{f}(F,b),x)$ vs. $\mathsf{f}(\mathsf{f}(F,x),b)$). In fact, it can be shown that forgetting through f^* necessarily yields a smaller AF, than iterating f no matter which order is chosen.

Forgetting via e_1 The main reason for the impossibility to find an operator satisfying e_1 under stable semantics is an intrinsic one, namely realizability. 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, one reasonable strategy is to look for forgetting operators satisfying e_1 whenever possible, and trying to satisfy a certain relaxation if not. Natural candidates 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 et al. 2017).

The question which relaxation to choose has no clear answer. First of all, according to Figure 1 each proposed desideratum is independent of the 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 5. *Consider again AF F presented in Example 2. Let $X = \{a,b,c\}$. As $stb(F) = \{\{a,e,f\},\{b,f,d\},\{c,d,e\}\}$ we obtain $\{E \setminus X \mid E \in stb(F)\} = \{\{d,e\},\{d,f\},\{e,f\}\}$. 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 realizable 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 is left for future work.

6 Discussion and Conclusion

The paper sheds more light on forgetting in abstract argumentation. One central motivation was to convey ideas and desiderata from recent studies of forgetting in the realm of logic programming (Knorr and Alferes 2014; Gonçalves, Knorr, and Leite 2016a; Berthold et al. 2019a; 2019b). We redefined several principles and provided a comprehensive study regarding satisfiability and relations. Moreover, we demonstrated that already existing forgetting operators from logic programming cannot be unconditionally applied to abstract argumentation. The two main reasons are: First, the use of such an operator does not guarantee to stay within the AF-fragment and secondly (as well as more importantly), there are essential differences in the expressibility between both formalisms. Finally, we presented a specific forgetting operator for a particular combination of conditions inspired by an algorithm introduced in (Baumann, Gabbay, and Rodrigues 2020).

One future line of research is the study of forgetting regarding labelling-based semantics (Baroni, Caminada, and Giacomin 2018). These kind of semantics provide some more information than their extension-based counterparts. In contrast to the latter they allow to distinguish explicitly two different kinds of not being accepted, namely *out* (attacked by an accepted argument) and *undec* (not attacked by an accepted argument). Consequently, more differentiated desiderata can be formalized regarding the acceptance status of the arguments to be forgotten. A further related work in this context is (Rienstra et al. 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 desidered properties.

Finally, the paper can be seen as one central part of a much broader investigation on how properties of forgetting on the abstract and structured level are related. One interesting agenda might be to consider *rationality postulates* for forgetting (Caminada 2017).

Acknowledgements

This work was supported by the German Federal Ministry of Education and Research (BMBF, 1IS18026A) by funding the competence center for Big Data and AI "ScaDS.AI" Dresden/Leipzig.

References

Baroni, P., and Giacomin, M. 2007. On principle-based evaluation of extension-based argumentation semantics. *Artif. Intell.* 171:675–700.

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

Baroni, P.; Caminada, M.; and Giacomin, M. 2018. Abstract argumentation frameworks and their semantics. In Baroni, P.; Gabbay, D.; Giacomin, M.; and van der Torre, L., eds., *Handbook of Formal Argumentation*. College Publications. chapter 4.

Baumann, R., and 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*, volume 9060, 281–295. Springer.

Baumann, R.; Gabbay, D. M.; and Rodrigues, O. 2020. Forgetting an argument. In *The Thirty-Fourth AAAI Conference on Artif. Intell., AAAI 2020, The Thirty-Second Innovative Applications of Artif. Intell. Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artif. Intell., EAAI 2020, New York, NY, USA, February 7-12, 2020*, 2750–2757. AAAI Press.

Baumann, R. 2018. On the nature of argumentation semantics: Existence and uniqueness, expressibility, and replaceability. In Baroni, P.; Gabbay, D.; Giacomin, M.; and van der Torre, L., eds., *Handbook of Formal Argumentation*. College Publications. chapter 18. also appears in IfCoLog Journal of Logics and their Applications 4(8):2779-2886.

Berthold, M.; Gonçalves, R.; Knorr, M.; and Leite, J. 2019a. Forgetting in answer set programming with anonymous cycles. In Moura Oliveira, P.; Novais, P.; and Reis, L. P., eds., *Progress in Artif. Intell.*, 552–565. Cham: Springer International Publishing.

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

Bisquert, P.; Cayrol, C.; de Saint-Cyr, F. D.; and Lagasquie-Schiex, M. 2011. Change in argumentation systems: Exploring the interest of removing an argument. In *Scalable Uncertainty Management - 5th International Conference, SUM 2011*, 275–288.

Caminada, M. 2017. Rationality postulates: Applying argumentation theory for non-monotonic reasoning. *FLAP* 4(8).

Dung, P. M. 1995. On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games. *Artif. Intell.* 77(2):321–357.

Dunne, P. E.; Dvořák, W.; Linsbichler, T.; and Woltran, S. 2015. Characteristics of multiple viewpoints in abstract argumentation. *Artif. Intell.* 228:153–178.

Eiter, T., and Kern-Isberner, G. 2018. A brief survey on forgetting from a knowledge representation and reasoning perspective. *KI - Künstliche Intelligenz*.

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

Gonçalves, R.; Knorr, M.; Leite, J.; and 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.

Gonçalves, R.; Knorr, M.; and Leite, J. 2016a. The ultimate guide to forgetting in answer set programming. In *Principles of Knowledge Representation and Reasoning: Proceedings of the Fifteenth International Conference, KR 2016*, 135–144.

Gonçalves, R.; Knorr, M.; and Leite, J. 2016b. You can't always forget what you want: On the limits of forgetting in answer set programming. In *ECAI 2016 - 22nd European Conference on Artif. Intell.*, 957–965.

Knorr, M., and Alferes, J. J. 2014. Preserving strong equivalence while forgetting. In *Proceedings of the 14th European Conference on Logics in Artif. Intell. (JELIA-14)*, 412–425.

Lifschitz, V.; Pearce, D.; and Valverde, A. 2001. Strongly equivalent logic programs. *ACM Transactions on Computational Logic* 2(4):526–541.

Lifschitz, V.; Tang, L. R.; and Turner, H. 1999. Nested expressions in logic programs. *Annals of Mathematics and Artif. Intell.* 25(3-4):369–389.

Lin, F., and Reiter, R. 1994. Forget it. In *Working Notes of AAAI Fall Symposium on Relevance*, 154–159.

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

Simari, G. R., and Rahwan, I., eds. 2009. *Argumentation in Artif. Intell.* Springer.

Strass, H. 2013. Approximating operators and semantics for abstract dialectical frameworks. *Artif. Intell.* 205:39–70.