Forgetting an Argument

Ringo Baumann
Department of Computer Science
Leipzig University
Germany

Dov Gabbay
Department of Informatics
King's College London
United Kingdom

Odinaldo Rodrigues
Department of Informatics
King's College London
United Kingdom

Abstract

The notion of *forgetting*, as considered in the famous paper by Lin and Reiter in 1994 has been extensively studied in classical logic and more recently, in non-monotonic formalisms like logic programming. In this paper, we convey the idea of forgetting to another major AI formalism, namely Dung-style argumentation frameworks. Our approach is axiomatic-driven and not limited to any specific semantics: we propose semantical and syntactical desiderata encoding different criteria for what *forgetting an argument* might mean; analyze how these criteria relate to each other; and check whether the criteria can be satisfied in general. The analysis is done for a number of widely used argumentation semantics. Our investigation shows that almost all desiderata are individually satisfiable. However, combinations of semantical and/or syntactical conditions reveal a much more interesting landscape. For instance, we found that the ad hoc approach to forgetting an argument, i.e., by the syntactical removal of the argument and all of its associated attacks, is too restrictive and only compatible with the two weakest semantical desiderata. Amongst the several interesting combinations identified, we showed that one satisfies a notion of minimal change and presented an algorithm that given an AF F and argument x, constructs a suitable AF G satisfying the conditions in the combination.

1 Introduction

Hiding relevant information as well as discarding irrelevant information are tasks which are performed by humans in a straightforward way. The need for a formalisation of the latter was recognised by Lin and Reiter in the context of *reasoning about actions* (Lin and Reiter 1994). Lin and Reiter were concerned about how to update a given knowledge base to remove (or "forget") facts which no longer remained true after actions were executed. Since then the topic, usually referred to as *forgetting* (and also known as *variable elimination*), has been extensively studied in the field of knowledge representation and reasoning for many major formalisms like propositional logic, first order logic and description logics as well as non-monotonic formalisms such as answer set programming (see (Eiter and Kern-Isberner 2018; Delgrande 2017; Gonçalves, Knorr, and Leite 2016a) for recent overviews). Applications of forgetting include *query answering*, *belief update* and *decision making* (see (Lang, Liberatore, and Marquis 2003) for more details).

Forgetting for a logical formalism can be roughly described as follows. Given a knowledge base $\mathcal{K}$ in this formalism and a variable x deemed *irrelevant*, the result of forgetting x in $\mathcal{K}$ is a knowledge base $\mathcal{K}'$, s.t.

LR1. The variable x does not occur in $\mathcal{K}'$.

LR2. All logical consequences of $\mathcal{K}'$ are logical consequences of $\mathcal{K}$.

LR3. All logical consequences of $\mathcal{K}$ that do not contain x are logical consequences of $\mathcal{K}'$.

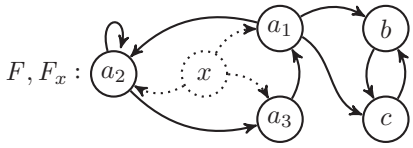
Lin and Reiter showed that (LR1)–(LR3) can be fulfilled via a simple syntactic transformation. For the case of propositional logic, a possible construction dating back to (Boole 1854) involves replacing every formula ϕ in $\mathcal{K}$ with the disjunction of the formulas obtained from ϕ by substituting x by *true* and *false*, respectively.

In this paper we study the notion of forgetting in the realm of Dung's Abstract Argumentation Frameworks (AFs) (Dung 1995), in which one of the chief concerns is the handling of conflicts amongst arguments. A large number of semantics expressing acceptable positions for AFs (so-called *extensions*) have been subsequently introduced (see (Baroni, Caminada, and Giacomin 2018) for a good overview). In recent years, the community started to investigate the effects of changes to an argumentation framework. One extensively studied issue is the so-called *enforcing problem* (Baumann and Brewka 2010; de Saint-Cyr et al. 2016; Wallner, Niskanen, and Järvisalo 2017), which is concerned with how to manipulate an argumentation framework in such a way that a certain desired set of arguments becomes an extension. Another closely related line of research is the principle-based study of *revision operators* (Coste-Marquis et al. 2014; Baumann and Brewka 2015; Diller et al. 2018) based on a reformulation of the famous AGM postulates for belief revision (Alchourrón, Gärdenfors, and Makinson 1985).

Perhaps surprisingly, how to "forget" single arguments has not yet received much attention. The most straightfoward way of forgetting an argument is simply by removing it from the argumentation framework (as considered in (Bisquert et al. 2011)). Whilst such a syntactical approach obviously guarantees that the extensions of the resulting AF

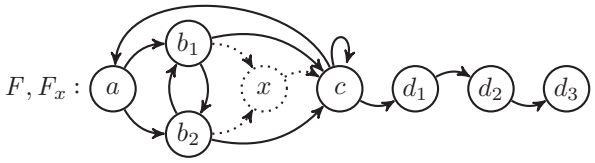
will not contain the argument, it leaves the precise semantical relationship between the input and output frameworks unclear. The following motivating examples show some undesired effects of the syntactical removal of arguments and some possible alternatives. For an overview of argumentation semantics and reasoning modes, see Section 2.

Example 1 (Forgetting Everything Credulously Accepted)**.** *The graph below represents two AFs F and F_x, s.t. x is an argument in F but not in F_x. The preferred extensions of F are $\{x, b\}$ and $\{x, c\}$. Hence, b and c are credulously accepted in F. The syntactical removal of x guarantees the semantical removal of x but has the side-effect that neither the credulous acceptance of b nor that of c "survive" since F_x's sole preferred extension is empty.*

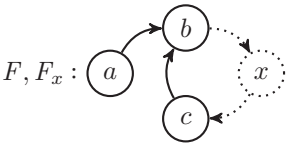


If we were to delete the argument a_1 in addition to x we would preserve the credulous acceptance of b and c since the preferred extensions of $(F_x)_{a_1}$ would be $\{b\}$ and $\{c\}$.

Example 2 (Forgetting Everything Sceptically Accepted)**.** *F's sole preferred extension 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 removal of x does not preserve any sceptical acceptance since F_x's sole preferred extension is empty. If we were to delete the argument a instead, we would still be able to forget x semantically as well as to preserve the sceptical acceptance of d_1 and d_3 since the preferred extensions of F_a would be $\{b_1, d_1, d_3\}$ and $\{b_2, d_1, d_3\}$.*



Example 3 (Accepting the Unacceptable)**.** *The unique preferred extension of F is $\{a, x\}$. The syntactical removal of x replaces the former extension with $\{a, c\}$. This means that the removal prevents the acceptance of x but yields the sceptical and credulous acceptance of c (previously unaccepted).*



If we were to find an argument n which attacked both x and c and added it to F, then we would end up with the unique preferred extension $\{a\} = \{a, x\} \setminus \{x\}$ which precisely does the job in terms of both acceptance modes.

The objective of the examples above is not to show how to achieve the syntactical/semantical removal of an argument, as there are multiple possibilities, but rather that we need to reason about possible constraints on the result of the forgetting operation.

The rest of the paper is organized as follows. We provide some background material in Section 2. This is followed by our main contributions, summarised below.

- Inspired by forgetting in other logical formalisms we identify reasonable semantical and syntactical properties for Dung-style AFs, and then show how they relate to each other. (Section 3)
- We then analyze the individual and joint satisfiability of conditions. We consider all 24 suitable combinations of syntactical and semantical criteria and identify that 9 are non-trivial combinations that are simultaneously satisfiable. Moreover, we identify $\subseteq$-minimal sets of conditions that are not simultaneously satisfiable. These unsatisfiable sets represent general restrictions on the simultaneous satisfiablity of forgetting desiderata. (Section 4)
- We then single out one promising combination under one of the most important argumentation semantics, namely the stable semantics, and provide a sound algorithm that given any AF F will compute an updated AF F' satisfying the conditions in the combination as well as a particular notion of minimal change to the AF. (Section 5)

The paper concludes with a discussion, comparisons with related work and directions for future work in Section 6.

2 Background Argumentation Theory

Argumentation Frameworks and Semantics

In what follows, we fix an infinite background set $\mathcal{U}$. An *abstract argumentation framework (AF)* (Dung 1995) is a directed graph $F = (A, R)$ where $A \subseteq \mathcal{U}$ is a set of arguments and $R \subseteq A \times A$ represents attacks between them. This means, for $a, b \in A$, 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 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. $E \subset \mathcal{I}$.*

In this paper, we consider the semantics 4–10, that is, the term *considered semantics* is used as a shorthand for the *stage, stable, semi-stable, preferred, grounded, ideal and eager semantics*.

Definedness, Acceptance and Realizability

If $\sigma(F) \neq \emptyset$ for any $F \in \mathcal{F}$, then we say that the semantics σ is *universally defined*, otherwise we say that σ *collapses*. All considered semantics are universally defined, with the exception of the stable semantics. This means that there are AFs F s.t. $stb(F) = \emptyset$. If $|\sigma(F)| = 1$, for any $F \in \mathcal{F}$, then we say that σ is *uniquely 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 two standard 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 *realizable w.r.t. a semantics σ* if there is an AF F s.t. $\sigma(F) = \mathcal{E}$. In this paper, we frequently use the facts that for any of the considered semantics a realizable set $\mathcal{E}$ has to be a $\subseteq$-antichain and that only the stable semantics may realize $\emptyset$, due its ability to collapse (Dunne et al. 2015).

3 Desiderata for Forgetting

As already mentioned in Section 1, apart from simply removing an argument and all attacks involving it (Bisquert et al. 2011), the notion of *forgetting* in AFs has never been considered in a principled way as done in (Lin and Reiter 1994) and (Eiter and Wang 2008) for classical logic or logic programming, respectively. We start our investigation by proposing reasonable/desirable properties of a forgetting operation. Given an AF F and an argument $x \in \mathcal{U}$, we use $\texttt{forget}_\sigma(F, x)$ to denote the set of AFs representing *"suitable" candidate AFs for the result of forgetting the argument x in F under the semantics σ*. Formally, it is a function $\texttt{forget}_\sigma : \mathcal{F} \times \mathcal{U} \to 2^{\mathcal{F}}$ mapping a pair (F, x) to a subset $\texttt{forget}_\sigma(F, x)$ of $\mathcal{F}$.

Whenever the semantics σ and the AF F are clear from the context, we will omit them and simply speak about *forgetting x*. The notion of *suitability* will be made precise via the following three blocks of desiderata, each considering a different aspect of the problem. We want to emphasize that the following lists do not express any preference amongst the desiderata. In practice, different criteria will apply to different contexts and there may be very good reasons to pick one criterium over another. The first two blocks are semantical in nature: $e1$–$e4$ concern the relationship between the old and the new set of extensions of the AF and $r1$–$r4$ deal with properties regarding sceptical and credulous reasoning. Finally, $s1$–$s3$ are purely syntactical. If a function $\texttt{forget}_\sigma$ always returns at least one suitable AF G w.r.t. desideratum d (i.e. $|\texttt{forget}_\sigma(F, x)| \geq 1$ for any $F \in \mathcal{F}$ and $x \in \mathcal{U}$), we say that $\texttt{forget}_\sigma$ *satisfies* d or that the *desideratum d is satisfiable under σ*.

Desiderata 1. *Given an AF F and an argument $x \in \mathcal{U}$. For $G \in \texttt{forget}_\sigma(F, x)$ we require:*

- $e1$. $\sigma(G) = \{E \setminus \{x\} \mid E \in \sigma(F)\}$, *(Reiter condition)*
- $e2$. *for any AF H, s.t. $x \notin A(H)$ we have: $\sigma(G \sqcup H) = \{E \setminus \{x\} \mid E \in \sigma(F \sqcup H)\}$. (strong Reiter condition)*
- $e3$. $\sigma(G) = \{T(E) \mid E \in \sigma(F)\}$ *with $T : \sigma(F) \to 2^{\mathcal{U}}$ and $E \mapsto T(E) \subseteq E \setminus \{x\}$ (weak Reiter condition)*
- $e4$. $\sigma(G) = \sigma(F) \setminus \{E \mid E \in \sigma(F), x \in E\}$ *(remove "x-contaminated" extensions)*

$e1$ requires that the only argument to be removed from any of the previous extensions is x itself (if x is in the extension). This so-called *Reiter condition* can be strengthened ($e2$) or weakened ($e3$) as follows. $e2$ stipulates that the Reiter condition carries over to any future expansion of F by an AF H which does not contain x. $e3$ weakens $e1$ in the sense that it allows other arguments besides x to be removed from each of the previous extensions. Finally, $e4$ stipulates that the extensions of the new AF should be exactly those in $\sigma(F)$ that do not contain x.

Desiderata 2. *Given an AF F and an argument $x \in \mathcal{U}$. For $G \in \texttt{forget}_\sigma(F, x)$ we require:*

- $r1$. $x \notin \bigcap \sigma(G)$ *(x is not scept. accepted)*
- $r2$. $x \notin \bigcup \sigma(G)$ *(x is not cred. accepted)*
- $r3$. $\bigcap \sigma(G) = (\bigcap \sigma(F)) \setminus \{x\}$ *(rigid scept. acceptance)*
- $r4$. $\bigcup \sigma(G) = (\bigcup \sigma(F)) \setminus \{x\}$ *(rigid cred. acceptance)*

The first two desiderata $r1$ and $r2$ say that the argument x should not be sceptically (resp., credulously) accepted after it is forgotten. $r3$ and $r4$ strengthen $r1$ and $r2$, resp., in the sense that they precisely determine the resulting set of sceptically or credulously accepted arguments, i.e., except for x, any former argument sceptically (resp., credulously) accepted remains sceptically (resp., credulously) accepted and only those arguments are sceptically (resp., credulously) accepted.

Notice that $e1$–$e4$ and $r1$–$r4$ do not necessarily enforce the removal of the argument to be forgotten from the AF. They are in nature semantical desiderata and even allow the

potential addition of new arguments. The addition of new arguments is completely reasonable in the context of argumentation because it is what normally happens during a debate, where in general arguments do not simply disappear. Instead, opponents aim to obliterate an adversarial argument by putting forward a new argument that directly or indirectly undermines it (this was illustrated in Example 3). Moreover, from a technical point of view, constructing an AF yielding a given set of extensions $\mathcal{E}$ typically requires additional arguments different from those in $\mathcal{E}$ (Baumann et al. 2016). However, in practice, it is the context of the application that will determine what criteria are appropriate and we do not express any preference here.

Finally, we present the syntactical desiderata $s1$–$s3$. As before, $s1$ does not prevent the addition of new arguments, but $s2$ and $s3$ do. All of $s1$–$s3$ require the actual removal of the argument to forget from the argumentation framework. In the same token as the addition of a new argument, removal of an argument could be justifiable for many reasons, e.g., questions about its legitimacy, the reliability of its source, or inconsistency (in the case of arguments expressed in a logical language).

Desiderata 3. *Given an AF F and an argument $x \in \mathcal{U}$. For $G \in \texttt{forget}_\sigma(F, x)$ we require:*

$s1$. $x \notin A(G)$ *(x is not contained)*

$s2$. $A(G) = A(F) \setminus \{x\}$ *(precise set of arguments)*

$s3$. $G = F|_{A(F)\setminus\{x\}}$ *(rigid AF)*

More precisely, $s1$ only requires that x does not belong to the set of arguments of the new AF. $s2$ strengthens $s1$ by requiring that nothing except x is removed from the set of arguments of the previous AF and no new arguments are added. Neither $s1$ nor $s2$ stipulate any conditions on the attack relation. $s3$ strengthens $s2$ further by also requiring the preservation of all attacks not involving x.

The previous syntactical conditions become progressively stronger. Indeed there are further relationships between all introduced conditions as shown next.

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 $\texttt{forget}_\sigma$ satisfying c under σ also satisfies c' under σ. Moreover, only these relations hold.*

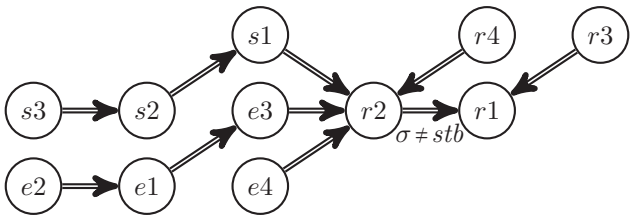


Figure 1: Dependencies

Proof: We start with the valid relations. Since $A\big(F|_{A(F)\setminus\{x\}}\big) = A(F) \setminus \{x\}$ we immediately obtain $s3 \Rightarrow s2$ and moreover, since $x \notin A(F) \setminus \{x\}$, $s2 \Rightarrow s1$ holds. Furthermore, since extensions of G are necessarily subsets of $A(G)$, $x \notin A(G)$ implies that $x \notin \bigcup\sigma(G)$, and hence $s1 \Rightarrow r2$.

If for any AF H s.t. $x \notin A(H)$ we have $\sigma(G \sqcup H) = \{E \setminus \{x\} \mid E \in \sigma(F \sqcup H)\}$ ($e2$), then taking $H = (\varnothing, \varnothing)$ gives us $\sigma(G) = \{E \setminus \{x\} \mid E \in \sigma(F)\}$, so clearly $e2 \Rightarrow e1$. Moreover, $e1$ guarantees $e3$, if we take $T(E) = E \setminus \{x\}$. Thus, $e1 \Rightarrow e3$. For any suitable function T, $\bigcup\{T(E) \mid E \in \sigma(F)\} \subseteq \bigcup\{E \setminus \{x\} \mid E \in \sigma(F)\}$ and obviously we have that $x \notin \bigcup\{E \setminus \{x\} \mid E \in \sigma(F)\}$, so $e3 \Rightarrow r2$. Similarly, $x \notin \bigcup\sigma(F) \setminus \{E \mid E \in \sigma(F), x \in E\}$, so $e4 \Rightarrow r2$.

If $\sigma(G)$ is non-empty, then $\bigcap\sigma(G) \subseteq \bigcup\sigma(G)$. Hence, $r2 \Rightarrow r1$ for any of the considered semantics with the exception of stb. Finally, $x \notin (\bigcap\sigma(F)) \setminus \{x\}$ and $x \notin (\bigcup\sigma(F)) \setminus \{x\}$, and hence $r3 \Rightarrow r1$ and $r4 \Rightarrow r2$.

In order to complete the proof, one would need to show that the non-existence of a path from condition c to condition c' implies the existence of a function $\texttt{forget}_\sigma$ satisfying c under σ, but not c'. For example, let $G = (\varnothing, \varnothing)$ and consider the constant function $\texttt{forget}_\sigma(F, x) = \{G\}$. Obviously, $\texttt{forget}_\sigma$ satisfies $s1$ as $x \notin \varnothing = A(G)$. However, the AF $F = (\{y\}, \varnothing)$ shows that $\texttt{forget}_\sigma$ does not satisfy $s1$, since $A(F) \setminus \{x\} = \{y\} \setminus \{x\} = \{y\} \neq \varnothing = A(G)$. For space reasons, we will not show all of these cases in this paper. ■

Whenever there is no path from the condition c to the condition c' in the diagram of Figure 1, we will say that c' *is non-trivial w.r.t. c*. Otherwise we will say that the combination of c and c' is *trivial*.

4 Individual and Combined Satisfiability

Clearly, the *strong Reiter condition* $e2$ is a very desirable property for any forgetting operation. Its logic programming analogue (so-called *strong persistence*) was firstly introduced in (Knorr and Alferes 2014) and further investigated in (Gonçalves, Knorr, and Leite 2016b). They showed that it is not always possible to forget variables from a program while obeying this property. In the same spirit, we start by analysing the individual satisfiability of all conditions from the outset. However, unlike (Gonçalves, Knorr, and Leite 2016b), which concentrated on the study of necessary and sufficient conditions for the satisfiability of strong persistence, we will focus on the simultaneous satisfiability of criteria and possible constructions of forgetting operations.

Satisfiability of Individual Conditions

The following proposition shows that the majority of the desiderata is individually satisfiable under all semantics considered in this paper.

Proposition 2. *Let $\sigma \in \{stg, stb, ss, pr, gr, il, eg\}$ and $d \in \{e3, r1, r2, r3, r4, s1, s2, s3\}$. It holds that desideratum d is satisfiable under σ.*

Proof: In order to prove this assertion it suffices to present a specific function $\texttt{forget}_\sigma$, s.t. for any $F \in \mathcal{F}$ and $x \in \mathcal{U}$, there is an AF $G \in \texttt{forget}_\sigma(F, x)$ fulfilling desideratum d. By Proposition 1, the satisfiability of $s3$ implies the

satisfiability of $s1$, $s2$ and $r2$ and the satisfiability of $r3$ implies the satisfiability of $r1$. Therefore, it suffices to show that desiderata $s3$, $r3$, $r4$ and $e3$ are satisfiable under σ.
($s3$): It is easy to see that the AF $G = F|_{A(F)\setminus\{x\}}$ satisfies $s3$, and there are no conditions on σ, so $s3$ is satisfiable under σ.
($r3$,$r4$): Consider the AFs $G_{r3} = ((\bigcap\sigma(F)) \setminus \{x\}, \varnothing)$ and $G_{r4} = ((\bigcup\sigma(F)) \setminus \{x\}, \varnothing)$. We have that $\sigma(G_{r3}) = \{(\bigcap\sigma(F)) \setminus \{x\}\}$ and $\sigma(G_{r4}) = \{(\bigcup\sigma(F)) \setminus \{x\}\}$. Recall that for any set E, $\bigcap\{E\} = \bigcup\{E\} = E$. Hence, $\bigcap\sigma(G_{r3}) = \bigcap\{(\bigcap\sigma(F)) \setminus \{x\}\} = (\bigcap\sigma(F)) \setminus \{x\}$ as required. Therefore, $G_{r3} \in \texttt{forget}_\sigma(F, x)$, and hence $r3$ is satisfiable under σ. $\texttt{forget}_\sigma(F, x) = \{G_{r4}\}$ can be constructed analogously and proves $r4$'s satisfiability under σ.

($e3$): If $\sigma(F) = \varnothing$ we define $F = G$ and there is nothing to show. Otherwise, $\sigma(F) = \{E_1, ..., E_n\}$ for some natural number n. Define $G_{e3} = (\bigcap_{1\le i\le n} E_i \setminus \{x\}, \varnothing)$. We have $\sigma(G_{e3}) = \{\bigcap_{1\le i\le n} E_i \setminus \{x\}\}$. This means, we consider the constant function $T(E) = \bigcap_{1\le i\le n} E_i \setminus \{x\}$. Obviously, $\bigcap_{1\le i\le n} E_i \setminus \{x\} \subseteq E_j \setminus \{x\}$ for any $1 \le j \le n$ proving the satisfiability of $e3$ under σ. ■

Condition $e1$ can only be satisfied under certain semantics.

Proposition 3. *Given $\sigma \in \{gr, il, eg\}$, $\tau \in \{stb, stg, ss, pr\}$. Desideratum $e1$ is satisfiable under σ, but not under τ.*

Proof: The satisfiability of $e1$ under σ is quite straigthforward. The semantics gr, il and eg have a unique extension. So let $\sigma(F) = \{E\}$. Then the function $\texttt{forget}_\sigma(F, x) = \{(E \setminus \{x\}, \varnothing)\}$ does the job.

In order to show $e1$'s unsatisfiability under τ we use the representational limits of the considered argumentation semantics τ. Consider the AF $F = (\{a, b, x\}, \{(a, x), (x, a)\})$, with $\tau(F) = \{\{a, b\}, \{x, b\}\}$. For the Reiter condition $e1$ to be satisfied, a candidate AF G would have to have $\tau(G) = \{\{a, b\}, \{b\}\}$, but this is not possible, since $\tau(G)$ does not happen to be a $\subseteq$-antichain and hence cannot be realized w.r.t. τ. Therefore, $\texttt{forget}_\tau(F, x) = \varnothing$, so $e1$ is not satisfiable under τ. ■

Proposition 4 shows that the strong Reiter condition $e2$ is unsatisfiable for any of the considered semantics, and it is hence arguably too strong.

Proposition 4. *Let $\sigma \in \{stg, stb, ss, pr, gr, il, eg\}$. Condition $e2$ is unsatisfiable under σ.*

Proof: From Proposition 1, we know that the satisfiability of $e2$ under semantics $\sigma \in \{stg, stb, ss, pr\}$ implies the satisfiability of $e1$ under semantics $\sigma \in \{stg, stb, ss, pr\}$. But by Proposition 3, $e1$ is not satisfiable under semantics $\sigma \in \{stg, stb, ss, pr\}$. Therefore, $e2$ is also not satisfiable under semantics $\sigma \in \{stg, stb, ss, pr\}$.

Consider a semantics $\sigma \in \{gr, il, eg\}$ and the AF $F = (\{a, x\}, \{(x, a)\})$. Striving for a contradiction we assume the existence of an AF G, s.t. for any AF H with $x \notin A(H)$ we have: $\sigma(G \sqcup H) = \{E \setminus \{x\} \mid E \in \sigma(F \sqcup H)\}$. Let us consider first $H_1 = (\{a, b\}, \{(a, b)\})$. $\sigma(F \sqcup H_1) = \{\{x, b\}\}$. Consequently, $\sigma(G \sqcup H_1) = \{\{b\}\}$. Since $\{b\}$ has to be admissible in $G \sqcup H_1$, we have that $(b, b) \notin R(G)$ (due to conflict-freeness) and moreover, $(b, a) \in R(G)$ (due to defence). Furthermore, $a, b \in A(G)$ follows.
Consider now $H_2 = (\{a\}, \{(a, a)\})$. Thus, $\sigma(F \sqcup H_2) = \{\{x\}\}$ implying that $\sigma(G \sqcup H_2) = \{\varnothing\}$. Since $\{b\}$ is not admissible in $G \sqcup H_2$ we deduce that there is a further argument c attacking b without being counterattacked. Note that the argument c cannot coincide with a since we already know that $(b, a) \in R(G)$. This means, we further conclude $c \in A(G)$, $(c, b) \in R(G)$ (attacker) and $(b, c) \notin R(G)$ (not counterattacked).
Finally, let us consider again H_1. We already know that $\sigma(G \sqcup H_1) = \{\{b\}\}$ which cannot be true since $\{b\}$ does not defend itself against c. This is a contradiction! ■

The last proposition of this section is about the removal of extensions. It turns out that the elimination of extensions containing a given argument can be only successfully accomplished under the stable semantics.

Proposition 5. *Given $\sigma \in \{stb\}$, $\tau \in \{stg, ss, pr, gr, il, eg\}$. Condition $e4$ is satisfiable under σ, but not under τ.*

Proof: The satisfiability of $e4$ under the stable semantics is an immediate consequence of Proposition 8 (see its proof for more details). All of the other semantics τ do not possess witnessing functions $\texttt{forget}_\tau$ because they are all universally defined. In case of the AF $F = (\{x\}, \varnothing)$ the condition $e4$ would require the realizability of the empty set of extensions. Consequently, $\texttt{forget}_\tau(F, x) = \varnothing$ proving the unsatisfiability of $e4$ under τ. ■

Due to limitations in space we will omit all subsequent proofs with the exception of the proof of Proposition 8, which is required by the above proof.

Combining Syntactical and Semantical Conditions

We have seen from Figure 1 that the satisfiability of some conditions imply the satisfiability of others. For example, the satisfiability of $s3$ implies the satisfiability of $s2$. This means that the combination $\{s2, s3\}$ is also satisfiable (recall we called these combinations trivial). We now consider the satisfiability of some *non-trivial* combinations.

Proposition 6 shows the compatibility of all combinations of a semantical and syntactical condition. The strongest syntactical condition $s3$ is incompatible with most of the semantical conditions and can only be trivially combined with $r1$ and $r2$ (for forgetting x under sceptical and credulous reasoning, respectively).

Its weaker counterpart $s2$, which unlike $s3$ does not constrain the attack relation, can however be combined with all reasoning conditions $r1$–$r4$. The weakest syntactical condition $s1$ can in addition satisfy the weak Reiter condition $e3$ under any of the considered semantics as well as condition $e4$ (the forgetting of extensions containing the argument to forget) under the stable semantics. Under the grounded, ideal and eager semantics, $s1$ and $s2$ can also be combined with $e1$, and $s2$ can be combined with $e3$.

Proposition 6. *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 and the symbol stb to the stb semantics only, respectively. The combinations in a dark background are trivial.

	$r1$	$r2$	$r3$	$r4$	$e1$	$e2$	$e3$	$e4$
$s1$	✓	✓	✓	✓	τ	×	✓	stb
$s2$	✓	✓	✓	✓	τ	×	τ	×
$s3$	✓	✓	×	×	×	×	×	×

Figure 2: Compatibility of syntactical/semantical conditions

Limits of Non-trivial Combinations

A subsequent question is whether one may add further criteria to a compatible syntactical and semantical combination taken from Figure 2. It turns out that there are serious inherent restrictions applying to any semantics. More precisely, there are minimal sets of criteria that cannot be simultaneously satisfied.

Proposition 7. *No set of conditions in the set* UNSAT $= \{\{e2\}, \{e4, e1\}, \{e4, e3\}, \{e4, r3\}, \{e4, r4\}, \{r3, r4\}\}$ *is satisfiable under any of the considered semantics.*

We already know from Proposition 6 that $e4$ is not compatible with $s2$ or $s3$ and it is only compatible with $s1$ under the stable semantics. The proposition above goes one step further to state that $e4$ is not compatible with any form of the Reiter conditions $e1$, $e2$ and $e3$ ($e2$ being itself independently unsatisfiable, by Proposition 4). In addition, the proposition states that $e4$ is not compatible with the rigid forms of sceptical and credulous acceptance $r3$ and $r4$, respectively. Finally, the proposition also states that in the forgetting of an argument x, it is not possible to simultaneously preserve both the sceptical and credulous acceptances of all other arguments. As a corollary we have that no superset of any of the sets of conditions in UNSAT is satisfiable either.

5 Construction Method

The choice of the combination of the conditions depends on the particular application in mind. In this section we illustrate how to construct a desired forgetting operation for one selected combination dealing with the popular stable semantics. The investigation of the construction of other valid combinations is left for future work. Before showing the construction method, we discuss the principle of *vacuity* which is a well-known concept in belief contraction (Alchourrón, Gärdenfors, and Makinson 1985). Roughly speaking, the axiom of vacuity requires that a knowledge base is left untouched by any contraction by a belief that is not included in it. In the context of the forget operation, the vacuity principle requires that the intial AF F is not changed if the argument x to forget *is irrelevant* to the imposed conditions. This of course may mean different things. We consider the following principles: that x is not sceptically (resp. credulously) accepted in F; and that x is not contained in $A(F)$ at all.

Desiderata 4. *Given an AF F and an argument $x \in \mathcal{U}$. For $G \in \texttt{forget}_\sigma(F, x)$ we require:*

$v1$. *If $x \notin \bigcap \sigma(F)$, then $F = G$.* *(scept. vacuity)*
$v2$. *If $x \notin \bigcup \sigma(F)$, then $F = G$.* *(cred. vacuity)*
$v3$. *If $x \notin A(F)$, then $F = G$.* *(argument vacuity)*

We have already shown that Desiderata $s1$ and $e4$ are simultaneously satisfiable under the stable semantics (Proposition 6). On top of $s1$ and $e4$, we could choose to add one of the vacuity conditions $v1$–$v3$. However, it turns out that $v1$ and $v2$ are too strong. This can be seen as follows: If $x \in A(F)$, but it is not sceptically or credulously accepted in F, then $v1$ and $v2$ would force $F = G$, and hence $x \in A(G)$, violating $s1$. This means, the only viable option would be $v3$. Although we do not explore this issue further in this paper, $v1$ and $v2$ are indeed compatible with some other non-trivial combinations.

Algorithm 1 shows how a function $\texttt{forget}_{stb}$ that simultaneously satisfies $s1$, $e4$ and $v3$ can be constructed.

Algorithm 1: Construct $G \in \texttt{forget}_{stb}(F, x)$

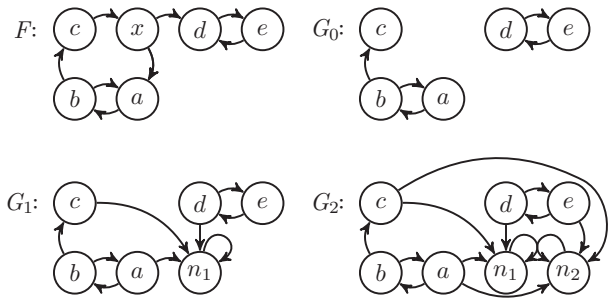
Input : AF F; argument $x \in \mathcal{U}$
Output: AF G satisfying $\{s1, e4, v3\}$

```
1  Function compute_G(F, x)
2      if x ∉ A(F) then G ← F;
3      else
4          G_0 ← F|_{A(F)\{x}};
5          A = A(G_0); R ← R(G_0);
6          foreach E_i ∈ stb(G_0) \ stb(F) do
7              Let a_i be a fresh argument s.t.
                 a_i ∉ A(F) ∪ A;
8              A ← A ∪ {a_i}; R ← R ∪ {(a_i, a_i)}
9              foreach y ∈ ⋃ stb(G_0) \ E_i do
10                 R ← R ∪ {(y, a_i)};
11         G ← (A, R);
12     return G;
```

Before formally proving that Algorithm 1 indeed does what promised we start with an exemplifying run.

Example 4. *Consider the AF F whose restriction to $\{a, b, c, d, e\}$ results in G_0. $stb(F) = \{\{b, x, e\}, \{a, c, d\}, \{a, c, e\}\}$ and $stb(G_0) = \{\{b, e\}, \{b, d\}, \{a, c, d\}, \{a, c, e\}\}$. Since $stb(G_0) \setminus stb(F) = \{\{b, e\}, \{b, d\}\} = \{E_1, E_2\}$ the algorithm goes through the loop in lines 6–10 twice.*

Let n_1 be the fresh argument chosen for E_1 (line 7). We obtain $A = \{a, b, c, d, e, n_1\}$ and $R = R(G_0) \cup \{(n_1, n_1)\} \cup \{(a, n_1), (c, n_1), (d, n_1)\}$ (lines 8 and 9–10), depicted in G_1. Proceeding to E_2, we take the new argument n_2. As before, this cycle sets $A = \{a, b, c, d, e, n_1, n_2\}$ and $R = R \cup \{(n_2, n_2), (a, n_2), (c, n_2), (e, n_2)\}$ as shown in $G_2 = (A, R)$ which is returned by the function as the AF G (lines 11 and 12). Note that $stb(G) = \{\{a, c, d\}, \{a, c, e\}\}$ as required.



Proposition 8. *Let G = compute_G(F, x). Then G satisfies conditions $s1$, $e4$ and $v3$ under the stable semantics.*

Proof: $s1$ is satisfied by the construction of G in Algorithm 1, lines 4–11. Moreover, condition $v3$ is ensured by line 2. The proof that G also satisfies $e4$, i.e., $stb(G) = stb(F) \setminus \{E \mid E \in stb(F), x \in E\}$ mainly relies on two well-known properties listed below.[1] Let us consider the disjoint union $stb(F) = \mathcal{E}_{\bar{x}} \cup \mathcal{E}_x$, where $\mathcal{E}_{\bar{x}} = \{E \in stb(F) \mid x \notin E\}$ and $\mathcal{E}_x = \{E \in stb(F) \mid x \notin E\}$.

1. Preservation of extensions not containing x. Given an AF F and its restriction $F|_{A(F)\setminus\{x\}}$. We have $\mathcal{E}_{\bar{x}} \subseteq stb\left(F|_{A(F)\setminus\{x\}}\right)$.
2. Elimination of single extensions. For any H and any $E \in stb(H)$ we have $stb(H') = stb(H) \setminus \{E\}$ where $A(H') = A(H) \cup \{n\}$ and $R(H') = R(H) \cup \{(a, n) | a \in (\bigcup stb(H) \cup \{n\}) \setminus E\}$ for a fresh argument n, i.e. $n \notin A(H)$.

Given these two points, it is easy to see that *i)* any additional occurring stable extension E_i of G_0, i.e. an E_i that is not stable in F will be removed (lines 6–10) and *ii)* every stable extension in $\mathcal{E}_{\bar{x}}$ is preserved as a stable extension of G. This is because $\mathcal{E}_{\bar{x}} \subseteq stb(G_0)$ and for every addition of a new argument n_i, there exists an attack from every argument $y \in \bigcup stb(G_0) \setminus E_i$ into n_i. Consequently, since $stb(G_0)$ forms a $\subseteq$-antichain it is guaranteed that every extension of $\mathcal{E}_{\bar{x}}$ contains an argument attacking n_i in G (lines 9–10). ■

6 Discussion and Conclusion

The paper takes previous work on forgetting in propositional logic and logic programming as a basis and studies the notion of forgetting in the context of abstract argumentation in general. In particular, we introduced several semantical and syntactical desiderata for a forgetting operation, provided a rigorous investigation of the satisfiability of combinations of these conditions as well as some impossibility results, and illustrated how a forgetting operation for a particular combination of conditions and semantics can be constructed. We considered seven well-known argumentation semantics and showed that many of our results hold independently of the particular semantics employed. In addition, we investigated a number of relevant issues arising in specific semantics.

[1] A procedure to eliminate unwanted stable extensions was first presented in (Dunne et al. 2015) and studied in a principled way in (Baumann and Brewka 2019).

In the area of logic programming many approaches to forgetting have been proposed. In (Zhang and Foo 2006) forgetting is implemented by simply removing from a logic program all rules containing the atom to be forgotten. This is purely syntactical in nature, and hence similar to our condition $s3$. (Eiter and Wang 2008) provided a comprehensive and principled study of forgetting in answer set programming, proposing five desirable properties covering semantical and syntactical aspects of the operation as well as different construction methods. Eiter and Wang observed that the Reiter condition $e1$ is unsatisfiable in the context of ASP and proposed the use of the so-called *minimal answer sets*. This approach guarantees the antichain property of the resulting set which leads to the realizability under the stable model semantics. However, this idea is not directly applicable to Dung-style AFs. For example, the set $\{\{a_1, a_2, x\}, \{a_1, a_4, a_5\}, \{a_2, a_3, a_5\}\}$ under $e1$ becomes $\{\{a_1, a_2\}, \{a_1, a_4, a_5\}, \{a_2, a_3, a_5\}\}$, which is realizable for ASP, but not under the stable semantics, because it violates the so-called *tightness* (Dunne et al. 2015). In (Knorr and Alferes 2014), the notion of *strong persistence* was introduced. As mentioned in Section 4, strong persistence is analogous to our condition $e2$ in the context of logic programming. A good overview of the study of the forgetting in the area of Answer Set Programming is given in (Gonçalves, Knorr, and Leite 2016b). The same authors showed that strong persistence is not satisfiable in general in (Gonçalves, Knorr, and Leite 2016a), and discussed some alternatives.

In the context of abstract argumentation, the most similar work to ours is that of (Bisquert et al. 2011). There, the so-called *expansive* and *narrowing* changes were proposed that result from the purely syntactical removal of an argument and its corresponding attacks (this corresponds to our desideratum $s3$). Although the authors provided a preliminary investigation of these changes, the investigation did not consider any further semantical desiderata nor any syntactical strengthenings of $s3$. Our paper therefore provides the first systematic and comprehensive account of the study of the forgetting operation in abstract argumentation.

Our function forget$_\sigma(F, x)$ returns a set of suitable candidates for the result of forgetting the argument x in F. This is akin to the famous postulates in belief revision, for example, which dictate the behaviour of revision operations in general without uniquely defining any particular operation (Alchourrón, Gärdenfors, and Makinson 1985). In the same spirit, our desiderata do not determine one unique forgetting operation. A specific forgetting operation may be defined in several ways, the most obvious of which is the "hard encoding" of the operation via the addition of an axiom enforcing uniqueness. Alternatively, one could simply redefine the range of the forgetting function or consider a two-step procedure whereby some of the conditions we introduced would lay the foundations of the operation and a selection function guided by specific desirable criteria, e.g., some notion of minimality, etc, would pick exactly one AF from the candidate options.

A further natural question to ask is how to extend our results to the forgetting of *sets* of arguments. Obviously,

one could iteratively forget each argument in the set, but this would not necessarily constitute an efficient or informationally economical way to perform the operation. Finally, a further in-depth analysis of construction methods for interesting combinations of forgetting conditions is also worthwhile. (Baumann et al. 2017) recently proposed the notion of *relativized equivalence* which allows one to specify *untouchable* arguments, i.e., arguments that cannot be disputed, and ask for simplifications. This seems to be an interesting feature that would allow the forgetting of arguments while simultaneously protecting others.

Acknowledgements

This work was supported by a postdoc fellowship of the German Academic Exchange Service (DAAD 57407370).

References

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

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 Brewka, G. 2010. Expanding argumentation frameworks: Enforcing and monotonicity results. In *Computational Models of Argument: Proceedings of COMMA 2010, Desenzano del Garda, Italy, September 8-10, 2010.*, 75–86.

Baumann, R., and Brewka, G. 2015. AGM meets abstract argumentation: Expansion and revision for Dung frameworks. In *Proceedings of the 24th International Conference on Artificial Intelligence*, IJCAI'15, 2734–2740. AAAI Press.

Baumann, R., and Brewka, G. 2019. Extension removal in abstract argumentation - An axiomatic approach. In *In Proceedings of the Thirty-Third Conference on Artificial Intelligence, AAAI 2019*, 2670–2677.

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

Baumann, R.; Dvorák, W.; Linsbichler, T.; and Woltran, S. 2017. A general notion of equivalence for abstract argumentation. In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence (IJCAI -17)*, 800–806.

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.

Boole, G. 1854. *An Investigation of The Laws of Thought on Which are Founded the Mathematical Theories of Logic and Probabilities*. London: Macmillan.

Coste-Marquis, S.; Konieczny, S.; Mailly, J.; and Marquis, P. 2014. On the revision of argumentation systems: Minimal change of arguments statuses. In *Principles of Knowledge Representation and Reasoning: Proceedings of the Fourteenth International Conference, KR 2014, Vienna, Austria, July 20-24, 2014*.

de Saint-Cyr, F. D.; Bisquert, P.; Cayrol, C.; and Lagasquie-Schiex, M. 2016. Argumentation update in YALLA (yet another logic language for argumentation). *International Journal of Approximate Reasoning* 75:57–92.

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

Diller, M.; Haret, A.; Linsbichler, T.; Rümmele, S.; and Woltran, S. 2018. An extension-based approach to belief revision in abstract argumentation. *International Journal of Approximate Reasoning* 93:395–423.

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

Dunne, P. E.; Dvořák, W.; Linsbichler, T.; and Woltran, S. 2015. Characteristics of multiple viewpoints in abstract argumentation. *Artificial Intelligence* 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., and Wang, K. 2008. Semantic forgetting in answer set programming. *Artificial Intelligence* 172(14):1644–1672.

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 Artificial Intelligence*, 957–965.

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

Lang, J.; Liberatore, P.; and Marquis, P. 2003. Propositional independence: Formula-variable independence and forgetting. *Journal of Artificial Intelligence Research* 18:391–443.

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

Wallner, J. P.; Niskanen, A.; and Järvisalo, M. 2017. Complexity results and algorithms for extension enforcement in abstract argumentation. *Journal of Artificial Intelligence Research* 60:1–40.

Zhang, Y., and Foo, N. Y. 2006. Solving logic program conflict through strong and weak forgettings. *Artificial Intelligence* 170(8-9):739–778.