The Thirty-Second International Florida
Artificial Intelligence Research Society Conference (FLAIRS-32)

Axiomatic Evaluation of Epistemic Forgetting Operators

Gabriele Kern-Isberner, Tanja Bock
Department of Computer Science
TU Dortmund University, Dortmund, Germany

Christoph Beierle, Kai Sauerwald
Department of Computer Science
University of Hagen, Hagen, Germany

Abstract

Forgetting as a knowledge management operation has received much less attention than operations like inference or revision. It was mainly in the area of logic programming that techniques and axiomatic properties have been studied systematically. However, at least from a cognitive view, forgetting plays an important role in restructuring and reorganizing a human's mind, and it is closely related to notions like relevance and independence which are crucial to knowledge representation and reasoning. In this paper, we propose axiomatic properties of (intentional) forgetting for general epistemic frameworks which are inspired by those for logic programming, and we evaluate various forgetting operations which have been proposed recently by Beierle et al. according to them. The general aim of this paper is to advance formal studies of (intentional) forgetting operators while capturing the many facets of forgetting in a unifying framework in which different forgetting operators can be contrasted and distinguished by means of formal properties.

1 Introduction

The term forgetting is used in various frameworks for quite different operations (Eiter and Kern-Isberner 2018). In the AGM framework (Alchourrón, Gärdenfors, and Makinson 1985), forgetting is called contraction and refers to the removal of a part of one's beliefs while in logic programming, forgetting aka variable elimination has been widely used for the removal of middle variables from a knowledge base (Lin and Reiter 1994). Recently, Beierle et al. (Beierle et al. 2019) discussed even more kinds of forgetting collected by findings in the knowledge representation literature as well as by common-sense understanding of forgetting. Each research field set up its own set of properties, or postulates, to distinguish between different forgetting operators. The field of logic programming offers the widest variety of operators and properties which were collected and categorized by Goncalves, Knorr, and Leite (Gonçalves, Knorr, and Leite 2016). We use this as a good starting point for building up a general, unifying framework for forgetting.

In this paper, we bring these different approaches to (intentional) forgetting together and present axiomatic properties of forgetting which are inspired by the postulates having been proposed for logic programming but expressed in a more abstract framework for forgetting operations in epistemic states. The logical object that is to be forgotten can be a variable or a sentence, thus covering both the operations of contraction and of variable elimination. We also consider a form of forgetting that can be achieved by conditionalization. We reinterpret the ASP postulates in the setting of epistemic states by defining formal analogies where, e.g., a logic program corresponds to an epistemic state. To realize substantially different forgetting operators like the ones mentioned above in one logic framework, we need to make use of approaches that provide a sufficiently broad base for operations. Spohn's ranking functions (Spohn 1988) have been used in (Beierle et al. 2019) to exemplify different forgetting operations in one semantic framework, and they also prove to be useful here for evaluating different forgetting operations according to the novel postulates. The results help revealing different features of the operators. More precisely, the main contributions of this paper are:

- We present novel postulates for forgetting sentences or variables from epistemic states, translating the basic ideas of postulates for forgetting in logic programs having been proposed by (Wong 2009; Gonçalves, Knorr, and Leite 2016) to more general frameworks; conditionals play a major role in this translation.
- The forgetting operators *contraction, marginalization* and *conditionalization* from (Beierle et al. 2019) are evaluated according to these postulates.
- The novel postulates prove to be helpful to distinguish different kinds of forgetting, thus at the same time revealing basic differences but also similarities among them.

This paper is organized as follows: In Section 2, we recall the formal preliminaries, to be able to present different properties of forgetting operators in ASP proposed over the years in Section 3. After briefly summoning the different kinds of forgetting in Section 4, we present general postulates for forgetting operators in Section 5. The postulates are used to evaluate the kinds of forgetting in Section 6, and in Section 7 we conclude and point out future work.

2 Preliminaries

Propositional Logic Let $\mathcal{L}_\Sigma$ be a propositional language finitely generated by a signature Σ. We write AB for $A \wedge B$, and $\overline{A}$ instead of $\neg A$. For an atom a, let $\dot{a}$ denote any of its literals $a, \overline{a}$. Ω_Σ denotes the set of possible worlds (propositional interpretations) over Σ. As usual, $\omega \models A$ means that the propositional formula $A \in \mathcal{L}_\Sigma$ holds in the possible world $\omega \in \Omega$, and $Mod(A) = \{\omega \in \Omega \mid \omega \models A\}$ denotes the set of all such possible worlds. We will use ω both for the model and the corresponding complete conjunction containing all atoms either in positive or negative form. The marginalization of a propositional formula A to a signature $\Sigma' \subseteq \Sigma$, $A|_{\Sigma'}$, is defined by iteratively forgetting every $\sigma \in \Sigma \setminus \Sigma'$ with $forget(A, \sigma) = A^+_\sigma \vee A^-_\sigma$, where A^+_σ and A^-_σ are the results of replacing all occurrences of σ in A with $\top$ and $\bot$, respectively. If $A \in \mathcal{L}$ is a formula, then the minimal set of signature elements from Σ needed to represent a formula which is equivalent to A is denoted by Σ_A.

ASP We consider logic programs under the answer set semantics (Gelfond and Leone 2002). A *logic program* P is a set of rules r of the form $h \leftarrow a_1, \ldots, a_l, not\, b_1, \ldots, not\, b_m$ where h and all elements in $\mathcal{B}^+(r) = \{a_1, \ldots, a_l\}$ and $\mathcal{B}^-(r) = \{b_1, \ldots, b_l\}$ are atoms. The set of all atoms of a program is denoted by Σ_P. A set of atoms S is called a model of a program P if for all $r \in P$ if $\mathcal{B}^+(r) \subseteq S$ and $\mathcal{B}^-(r) \cap S = \emptyset$ then $h \in S$. The Gelfond-Lifschitz reduct of a program P relative to a set of atoms S is the program $P^S = \{h \leftarrow \mathcal{B}^+(r) \mid r \in P, \mathcal{B}^-(r) \cap S = \emptyset\}$. S is an *answer set* of a program P if S is a minimal model of P^S. The set of all answer sets of a logical program P will be denoted by $\mathcal{AS}(P)$. The V-exclusion of a set of answer sets $\mathcal{M}$ by a set of atoms $V \subseteq \Sigma$ is $\mathcal{M}_{||V} = \{X \setminus V \mid X \in \mathcal{M}\}$. P_1 and P_2 are called strongly equivalent[1], namely $P_1 \equiv_S P_2$, if $\mathcal{AS}(P_1 \cup P') = \mathcal{AS}(P_2 \cup P')$ for any logic program P' (Lifschitz, Pearce, and Valverde 2001).

Definition 1 (Forgetting operator for ASP (Gonçalves, Knorr, and Leite 2016))**.** *Let $\mathcal{C}$ be a class of logic programs over Σ. A forgetting operator is a function $f : \mathcal{C} \times 2^\Sigma \rightarrow \mathcal{C}$, such that $f(P, V)$ is a logic program over $\Sigma_P \setminus V$. Then, we call $f(P, V)$ the result of forgetting V in P.*

Conditionals By introducing a new binary operator $|$, we obtain the set $(\mathcal{L}_\Sigma|\mathcal{L}_\Sigma) = \{(B|A) \mid A, B \in \mathcal{L}_\Sigma\}$ of conditionals over $\mathcal{L}_\Sigma$. $(B|A)$ formalizes "*if A then usually B*" and establishes a plausible connection between the *antecedent* A and the *consequent* B. Conditionals with tautological antecedents are taken as plausible statements about the world. Following (DeFinetti 1974), a conditional $(B|A)$ can be *verified* (*falsified*) by a possible world ω iff $\omega \models AB$ ($\omega \models A\overline{B}$) and is *not applicable* to ω if $\omega \not\models A$.

As established semantics for conditionals, we use *Ordinal conditional functions (OCFs)*, also called *ranking functions*, $\kappa : \Omega \rightarrow \mathbb{N}$ with $\kappa^{-1}(0) \neq \emptyset$, which were introduced (in a more general form) first by (Spohn 1988). They express degrees of plausibility of propositional formulas A by specifying degrees of disbelief of their negations $\overline{A}$. More formally, we have $\kappa(A) := \min\{\kappa(\omega) \mid \omega \models A\}$, so that $\kappa(A \vee B) = \min\{\kappa(A), \kappa(B)\}$.

A conditional $(B|A)$ is accepted in the epistemic state κ, written as $\kappa \models (B|A)$, iff $\kappa(AB) < \kappa(A\overline{B})$, i.e. iff the verification AB of the conditional is more plausible than its falsification $A\overline{B}$. This transfers easily to a propositional formula A by defining $\kappa \models A$ iff $\kappa \models (A|\top)$, iff $\kappa(\overline{A}) > 0$.

In most approaches to belief change, the new information is given by a propositional formula or by a set of propositional formulas. In the general framework developed by Kern-Isberner in (Kern-Isberner 2001) change operations taking conditionals or sets of conditionals into account are provided, based on the central principle of conditional preservation. We focus in this paper on the special case for a change with a single proposition A.

Definition 2 (c-change by a single proposition (Kern-Isberner et al. 2017))**.** *A belief change from κ to κ° by A fulfils the principle of conditional preservation and is called a* c-change *with A if there exist integers $\kappa_0, \gamma^+, \gamma^-$ such that:*

$$\kappa^\circ(\omega) = -\gamma^- - \kappa(\overline{A}) + \kappa(\omega) + \begin{cases} \gamma^+ & \text{if } \omega \models A \\ \gamma^- & \text{if } \omega \models \overline{A} \end{cases} \quad (1)$$

The abstract definition of a c-change allows for defining the several kinds of forgetting of Section 4, see (Beierle et al. 2019) for the full list of definitions.

3 ASP Postulates for Forgetting

ASP is presumably the discipline with the most properties describing forgetting operators. Whereas the AGM postulates for contraction have been established several years ago, it is still debatable which of the proposed properties for ASP are really necessary or meaningful. We recall here two different works on postulates and properties (Wong 2009; Gonçalves, Knorr, and Leite 2016). Let P, P' be logical programs, V, V' sets of atoms we want to forget, and $f(P, V)$ the result of the forgetting of V in P.

Wong defined properties for logic programs relative to a notion of equivalence, $\equiv$, and called them postulates for ASP logic program forgetting. We recall here only two of his postulates, the rest uses very specific aspects of logic programming, making a generalization difficult. For the entire list see (Wong 2009).

(F0) If $P \equiv P'$ then $f(P, V) \equiv f(P', V)$
(F6) $f(f(P, V'), V) \equiv f(f(P, V), V')$

The forgetting operators collected by Goncalves, Knorr, and Leite (Gonçalves, Knorr, and Leite 2016) are defined by properties of logical programs relative to a notion of equivalence, $\equiv$, and entailment, $\models$ (e.g. defined over HT-models (Lifschitz, Pearce, and Valverde 2001)).

(wC)^{asp} $\mathcal{AS}(P)_{||V} \subseteq \mathcal{AS}(f(P, V))$
(sC)^{asp} $\mathcal{AS}(f(P, V)) \subseteq \mathcal{AS}(P)_{||V}$
(CP)^{asp} $\mathcal{AS}(f(P, V)) = \mathcal{AS}(P)_{||V}$
(wE)^{asp} $(\mathcal{AS}(P) = \mathcal{AS}(P')) \Rightarrow (\mathcal{AS}(f(P, V)) = \mathcal{AS}(f(P', V)))$
(W)^{asp} $P \models f(P, V)$

[1]Some works define strong equivalence over HT-models (Lifschitz, Pearce, and Valverde 2001). Since we only adapt the general idea of equivalence here, we will not go deeper into it.

(SE)^{asp} $(P \equiv P') \Rightarrow (f(P,V) \equiv f(P',V))$

As we will use the same abbreviations for the general properties later we indicate the ASP properties here with the superscript ^{asp}.

4 Kinds of Forgetting

In (Beierle et al. 2019) an abstract model is used in which an agent is equipped with an epistemic state Ψ (also called belief state) and an inference relation $\approx$. They make no further assumptions about how this belief state is represented, except that Ψ makes use of a language $\mathcal{L}$ over a signature Σ. The relation $\Psi \approx A$ holds if an agent with belief state Ψ infers/believes/accepts A with $A \in \mathcal{L}$ or $A \in (\mathcal{L}|\mathcal{L})$. Let $\Psi_{|_{\Sigma'}}$ be the marginalized belief state for a subset Σ' of signatur elements with $\Psi_{|_{\Sigma'}} \approx A$ iff $\Psi \approx A$ for all $A \in \mathcal{L}_{\Sigma'}$. Under the assumption that a conditionalization operator $|$ on Ψ exists, $\Psi|A$ has the intended meaning that Ψ should be interpreted under the assumption that A holds. In particular, we can assume that $\Psi|A \approx A$ holds for every A.

If Ψ is a prior state, then we denote with Ψ°_A the posterior belief state of the agent after forgetting A resp. change operation on A. In this context the object A that we want to forget can be a formula from $\mathcal{L}$, or a variable from Σ. Different notions of forgetting can be specified by the inferences an agent can or can no longer draw after the change operation on A (Beierle et al. 2019); i.e., types of forgetting are characterised by their respective *success condition*. In this paper we focus on the three following major types of forgetting that have been applied in various frameworks:

Contraction	$\Psi^\circ_A \not\approx A$
Marginalization	$\Psi^\circ_A = \Psi\|_{\Sigma \setminus \Sigma_A}$
Conditionalization	$\Psi^\circ_A = \Psi\|\overline{A}$

Contraction refers to the intention to directly give up information A, as it is known from the AGM framework (Alchourrón, Gärdenfors, and Makinson 1985). Marginalization and conditionalization are well-established operators; they are reinterpreted here for defining forgetting operators that take the information A to be forgotten as their argument.

5 General Postulates of Forgetting Operators

We now want to define general postulates for operators based on the postulates/properties for ASP of Section 3. We will use the same abstract framework as described above based on belief states Ψ and will find abstract correspondences for the different components of the properties. A logic program P will be translated into an epistemic state Ψ, and the result of forgetting, in ASP denoted by $f(P,V)$, will be Ψ°_A. The V-exclusion $_{||V}$ corresponds to the marginalization $|_{\Sigma \setminus \Sigma_A}$ on propositional formulas. In order to be able to define abstract postulates independent of a concrete logic, we need a few prerequisites about belief states that go beyond classical logics while obeying general logical traditions. First and most importantly, belief states shall be capable of dealing with conditionals by means of an inference relation (or acceptance relation) $\approx$, i.e., $\Psi \approx (B|A)$ iff Ψ accepts $(B|A)$, and $\Psi \approx A$ iff $\Psi \approx (A|\top)$. The set of all (conditional) inferences will be denoted by $C(\Psi)$ with $C(\Psi) = \{(B|A) \in (\mathcal{L} \mid \mathcal{L}) \mid \Psi \approx (B|A)\}$. We define the entailment relation between two belief states Ψ_1, Ψ_2 by using the conditional inference relation and have $\Psi_1 \approx \Psi_2$ iff $C(\Psi_2) \subseteq C(\Psi_1)$. The equivalence relation over logic programs, $\equiv$, is lifted to an equivalence relation over belief states, $\cong$, as well. The equivalence is defined by the entailment relation $\approx$ on belief states with $\Psi_1 \cong \Psi_2$ iff $\Psi_1 \approx \Psi_2$ and $\Psi_2 \approx \Psi_1$. We use a propositional belief operator $Bel\,(\Psi) \subseteq \mathcal{L}$ representing the most plausible beliefs of an agent. $Bel\,(\Psi)$ then corresponds to $\mathcal{AS}(P)$ and is defined as $Bel\,(\Psi) = \{A \in \mathcal{L} \mid \Psi \approx (A|\top)\}$.

With this framework we are able to define general postulates that can be specialized for concrete logics by only specifying the inference relation $\approx$. To show the connection to ASP we note after each postulate the names of its correspondent ASP postulate/property where possible.

Weakening $\Psi \approx \Psi^\circ_A$ (W)
The posterior belief state has at most the same conditional inferences as the prior belief state.

weakened Consequence $Bel\,(\Psi)|_{\Sigma \setminus \Sigma_A} \subseteq Bel\,(\Psi^\circ_A)$ (wC)
Marginalized beliefs of the prior state are preserved; in particular those that do not contain any forgotten atoms.

strengthened Consequence
$Bel\,(\Psi^\circ_A) \subseteq Bel\,(\Psi)|_{\Sigma \setminus \Sigma_A}$ (sC)
Beliefs of the posterior belief state have to be marginalized beliefs of the prior belief state.

Consequence Persistence $Bel\,(\Psi^\circ_A) = Bel\,(\Psi)|_{\Sigma \setminus \Sigma_A}$ (CP)
Beliefs in the posterior belief state are the same as marginalized beliefs of the prior belief state; in particular, formulas that have none to be forgotten atom are preserved.

weak Equivalence
If $Bel\,(\Psi) \equiv Bel\,(\Phi)$ then $Bel\,(\Psi^\circ_A) \equiv Bel\,(\Phi^\circ_A)$ (wE)
If the belief sets of two belief states are equivalent before, then this equivalence is preserved while forgetting.

(Strong) Equivalence If $\Psi \cong \Phi$, then $(\Psi^\circ_A \cong \Phi^\circ_A)$ (E)
If two belief states are equivalent before the forgetting operation, then the posterior belief states are equivalent as well.

Order Independence $(\Psi^\circ_A)^\circ_B \cong (\Psi^\circ_B)^\circ_A$ (OI)
The result of an iterated forgetting is independent of the order in which the operators are applied.

These postulates make it possible to compare particular forgetting operators within one framework. In the following we will use them to evaluate the three different forgetting operations of Section 4.

6 Evaluation of Different Kinds of Forgetting by Postulates

We concentrate on the concrete instantiation of the operations with ranking functions shown in (Beierle et al. 2019). To apply the postulates to ranking functions we only need to define how the (conditional) inference relation will be interpreted in this framework.

Let κ be a ranking function over Σ. The *conditional inference relation* is defined as $C(\kappa) = \{(B|A) \in (\mathcal{L}|\mathcal{L}) \mid \kappa \approx (B|A)\}$ and a conditional is an inference of a ranking

	(W)	(wC)	(sC)	(CP)	(wE)	(E)	(OI)
Contraction	✗	✗	✗	✗	✗	✗	✗
Marginalization	✓	✓	✓	✓	✓	✓	✓
Conditionalization	✗	✗	✗	✗	✗	✓	✓

Table 1: Fulfilled postulates (✓) of the kinds of forgetting.

function, $\kappa \mathrel{|\!\approx} (B|A)$, iff $\kappa(AB) < \kappa(A\overline{B})$. With this, our notion of equivalence corresponds to the definition of *inferentially equivalent*, $\equiv_{\mid\!\sim}$, ranking functions by Beierle and Kutsch (Beierle and Kutsch 2018).

Definition 3 (inferentially equivalent)**.** *Two ranking functions κ and κ' are* inferentially equivalent*, denoted by $\kappa \equiv_{\mid\!\sim} \kappa'$ iff for all $(B|A)$ it holds that $\kappa \models (B|A)$ iff $\kappa' \models (B|A)$.*

This can be evaluated by comparing the ranks of all worlds.

Proposition 1 ((Beierle and Kutsch 2018))**.** *κ, κ' are inferentially equivalent, $\kappa \equiv_{\mid\!\sim} \kappa'$, iff for all $\omega_1, \omega_2 \in \Omega$ it holds that $\kappa(\omega_1) \leq \kappa(\omega_2)$ iff $\kappa'(\omega_1) \leq \kappa'(\omega_2)$.*

The implementation of each forgetting operator for ranking functions will be recalled from previous works in the respective subsection. All results, as to whether a forgetting operation fulfills certain postulates or not, are summarized in Table 1.

Contraction

c-contraction adapts c-changes as given in Definition 2 to the case of contraction.

Definition 4 (c-contraction by a single proposition (Kern-Isberner et al. 2017))**.** *A change from κ to κ° is called a c-contraction with A, if there exist integers γ^+, γ^- such that Equation 1 holds and the following condition is satisfied:*

$$\gamma^- - \gamma^+ \leq \kappa(A) - \kappa(\overline{A}) \qquad (2)$$

Proposition 2. *c-contraction does not fulfill (W), (wC), (sC), (CP) (wE), (E), and (OI).*

Proof. For (W) let κ be a ranking function over $\Sigma = \{a, b\}$ with $\kappa(ab) = \kappa(a\overline{b}) = 0$, and $\kappa(\overline{a}b) = \kappa(\overline{a}\,\overline{b}) = 1$. By contracting κ with a and $\gamma^+ = 2$, $\gamma^- = 0$, the two levels flip. It is $\kappa^\circ_a(\overline{a}b) = \kappa(\overline{a}\,\overline{b}) = 0$ and $\kappa^\circ_a(ab) = \kappa(a\overline{b}) = 1$, which leads to $(\overline{a}|b) \in C(\kappa^\circ_a)$ but $(\overline{a}|b) \notin C(\kappa)$.

c-contraction does not fulfill (wC). Let κ be a ranking function over $\Sigma = \{a, b\}$ with $\kappa(ab) = 0$, $\kappa(\overline{a}\,\overline{b}) = 1$, and $\kappa(a\overline{b}) = \kappa(\overline{a}b) = 2$. The contraction of κ with a and $\gamma^+ = 0$, $\gamma^- = 1$ leads to $\kappa^\circ_a(ab) = \kappa(\overline{a}\,\overline{b}) = 0$, $\kappa^\circ_a(\overline{a}b) = 1$, and $\kappa(a\overline{b}) = 2$. So it is $b \in \mathit{Bel}(\kappa)|_{\{b\}} = \mathit{Cn}(ab)|_{\{b\}} = \mathit{Cn}(b)$ but $b \notin \mathit{Bel}(\kappa^\circ_a) = \mathit{Cn}(\overline{a}\overline{b} \vee ab)$.

The counterexample for (W) is also a counterexample for (sC). We have $\overline{a} \in \mathit{Bel}(\kappa^\circ_a) = \mathit{Cn}(\overline{a})$, but $\overline{a} \notin \mathit{Bel}(\kappa)|_{\{b\}} = \mathit{Cn}(a)|_{\{b\}} = \mathit{Cn}(\top)$. c-contraction does not fulfill (wC) nor (sC), therefore (CP) cannot be fulfilled.

c-contraction does not fulfill (wE). In Figure 1 we have two ranking functions, κ_1, κ_2, with the same worlds in the lowermost layer. So, the prior belief sets are the same, i.e., $\mathit{Bel}(\kappa_1) = \mathit{Bel}(\kappa_2) = \mathit{Cn}(a)$. After the contraction with a and $\gamma^+ = 1, \gamma^- = 0$ this leads to two different belief sets, $\mathit{Bel}((\kappa_1)^\circ_a) = \mathit{Cn}(b) \neq \mathit{Bel}((\kappa_2)^\circ_a) = \mathit{Cn}(ab \vee \overline{a}b)$.

c-contraction does not fulfill (E). Two equivalent ranking functions are shown in Figure 2, κ_1 and κ_2, whereas κ_2 has an empty layer. This leads to different total pre-orders after forgetting a with c-contractions and $\gamma^+ = 2, \gamma^- = 0$, e.g we have $(a|\overline{b}) \notin C((\kappa_1)^\circ_a))$ but $(a|\overline{b}) \in C((\kappa_2)^\circ_a))$.

For (OI) consider the case where we have individual impacts for the contraction with a, namely $\gamma^+_a = 2, \gamma^-_a = 0$, and for the contraction with b: $\gamma^+_b = 1, \gamma^-_b = 0$. These impacts are fixed for both application orderings. The contraction first by a and then by b leads to the ranking function $(\kappa^\circ_a)^\circ_b$ shown on the left of Figure 3, which is different from the ranking function $(\kappa^\circ_b)^\circ_a$ shown on the right, which is the result of first forgetting b and then a, e.g it is $(\overline{a}|\overline{b}) \in C((\kappa^\circ_b)^\circ_a)$ and $(\overline{a}|\overline{b}) \notin C((\kappa^\circ_a)^\circ_b)$. □

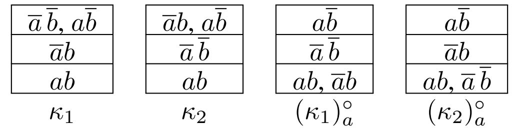

Figure 1: Counterexample for (wE) for Contraction

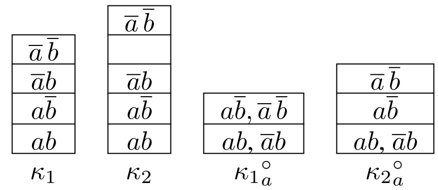

Figure 2: Counterexample for (E) for Contraction

Marginalization

Definition 5 (marginalization of κ to Σ' (Beierle et al. 2019))**.** *Let κ be a ranking function over Σ and $\Sigma' \subseteq \Sigma$. The marginalization of κ to Σ', denoted by $\kappa|_{\Sigma'} : \Omega_{\Sigma'} \to \mathbb{N}$, is given by*

$$\kappa|_{\Sigma'}(\omega') = \min\{\kappa(\omega) \mid \omega \in \Omega_\Sigma \text{ and } \omega \models \omega'\}. \qquad (3)$$

Applying this to implement marginalization as a forgetting operation according to Section 4, we obtain

$$\kappa^\circ_A = \kappa|_{\Sigma \setminus \Sigma_A}.$$

Proposition 3. *Marginalization fulfills (W), (wC), (sC), (CP), (wE), (E), and (OI).*

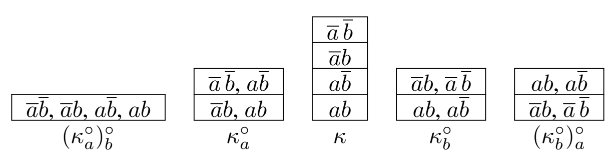

Figure 3: Counterexample for (OI) for Contraction

κ	$\kappa_b^\circ = \kappa\vert\overline{b}$
$a\overline{b}c, \overline{a}\,\overline{b}c, \overline{a}\,\overline{b}\,\overline{c}$	
$a\overline{b}\overline{c}$	$a\overline{b}c, \overline{a}\,\overline{b}c, \overline{a}\,\overline{b}\,\overline{c}$
$abc, ab\overline{c}, \overline{a}bc, \overline{a}b\overline{c}$	$a\overline{b}\,\overline{c}$

Figure 4: Counterexample for (W) for Conditionalization

κ_1	κ_2	$\kappa_1\vert a$	$\kappa_2\vert a$
$a\overline{b}$	ab		
ab	$a\overline{b}$	$a\overline{b}$	ab
$\overline{a}b, \overline{a}\,\overline{b}$	$\overline{a}b, \overline{a}\,\overline{b}$	ab	$a\overline{b}$

Figure 5: Counterexample for (wE) for Conditionalization

Proof. We show all properties for general marginalizations $\kappa|_{\Sigma'}$, this yields immediately the statement of the proposition for the forgetting operator as a special case of marginalization. (W) is clear because of $\kappa|_{\Sigma'}(AB) = \kappa(AB)$ for $A, B \in \mathcal{L}_{\Sigma'}$.

For (wC) fix an arbitrary $B \in \mathit{Bel}(\kappa)|_{\Sigma'}$. By definition of marginalization there exists a $C \in \mathit{Bel}(\kappa)$ such that $B = C|_{\Sigma'} = C^+_{\Sigma'} \vee C^-_{\Sigma'}$. It follows that $\kappa(\overline{C}) > 0$ and $C \models B$ holds. Then, we obtain $\kappa(\overline{B}) \geq \kappa(\overline{C})$ from the contra position of $C \models B$ and thus $\kappa(\overline{B}) > 0$. Because $B \in \mathit{Bel}(\kappa)$ and $\Sigma_B \subseteq \Sigma'$, it follows that $\kappa(\overline{B}) = \kappa|_{\Sigma'}(\overline{B}) > 0$ and we immediately get the result $B \in \mathit{Bel}(\kappa|_{\Sigma'})$.

For (sC) assume $B \in \mathit{Bel}(\kappa|_{\Sigma'})$, which directly implies that $B \in \mathcal{L}_{\Sigma'}$ and $0 < \kappa|_{\Sigma'}(\overline{B}) = \kappa(\overline{B})$. We can conclude that $B \in \mathit{Bel}(\kappa)$ and by $B \in \mathcal{L}_{\Sigma'}$ we get that B is equivalent to $B|_{\Sigma'}$ with respect to the models in $\Omega_{\Sigma'}$. This leads to the result $B \in \mathit{Bel}(\kappa)|_{\Sigma'}$. By (wC) and (sC) we have $\mathit{Bel}(\kappa)|_{\Sigma'} = \mathit{Bel}(\kappa|_{\Sigma'})$, therefore, (CP) is fulfilled.

Marginalization fullfills (wE) because $\mathit{Bel}(\kappa_1) = \mathit{Bel}(\kappa_2)$ means, that for all $\omega \in \Omega$ it holds that $\kappa_1(\omega) = 0$ if and only if $\kappa_2(\omega) = 0$ holds. $\kappa_1|_{\Sigma'}(\omega') = 0$ if there exists a world ω with $\omega \models \omega'$ and $\kappa_1(\omega) = 0$ and these are the same worlds for κ_2, which leads to $\mathit{Bel}(\kappa_1|_{\Sigma'}) = \mathit{Bel}(\kappa_2|_{\Sigma'})$.

For (E) let $\kappa_1 \cong \kappa_2$, and let $B, C \in \mathcal{L}_{\Sigma'}$. It is $(C|B) \in C(\kappa_1|_{\Sigma'})$ iff $\kappa_1|_{\Sigma'}(BC) < \kappa_1|_{\Sigma'}(B\overline{C})$ iff $\kappa_1(BC) < \kappa_1(B\overline{C})$ due to $B, C \in \mathcal{L}_{\Sigma'}$. With $\kappa_1 \cong \kappa_2$, this holds iff $\kappa_2(BC) < \kappa_2(B\overline{C})$ iff $\kappa_2|_{\Sigma'}(BC) < \kappa_2|_{\Sigma'}(B\overline{C})$, i.e., $(C|B) \in C(\kappa_2|_{\Sigma'})$. Therefore, $C(\kappa_1|_{\Sigma'}) = C(\kappa_2|_{\Sigma'})$, and hence $\kappa_1|_{\Sigma'} \cong \kappa_2|_{\Sigma'}$.

Marginalization fulfills (OI) because the marginalization first to A and then to B is the same as the marginalization to $\{A, B\}$. In both cases the rank of a world is determined by taking the minimum rank of a world over the reduced signature $\Sigma \setminus (\Sigma_A \cup \Sigma_B)$. □

Conditionalization

Definition 6 (conditionalization of κ by A (Spohn 1988))**.** *Let κ be a ranking function and A a proposition, then the conditionalization of κ by A is the ranking function $\kappa|A : Mod(A) \to \mathbb{N}$, defined on the models of A as follows:*

$$\kappa|A(\omega) = \kappa(\omega) - \kappa(A) \tag{4}$$

According to Section 4, this yields the forgetting operation

$$\kappa^\circ_A = \kappa|\overline{A}.$$

Proposition 4. *Conditionalization fulfills (E) and (OI) but does not fulfill (W), (wC), (sC), (CP), and (wE).*

Proof. For ease of reading, we show the properties to hold for general conditionalizations $\kappa|A$. Conditionalization fulfills (E). $\kappa_1 \cong \kappa_2$ iff $C(\kappa_1) = C(\kappa_2)$. This is the case iff for all $(B|C)$ it holds $\kappa_1 \models\!\approx (B|C)$ iff $\kappa_2 \models\!\approx (B|C)$. From Proposition 1 if follows that for all $\omega_1, \omega_2 \in \Omega$ we have $\kappa_1(\omega_1) \leq \kappa_1(\omega_2)$ iff $\kappa_2(\omega_1) \leq \kappa_2(\omega_2)$. We now only consider the models of A. The condition is still true which leads to $\kappa_1(\omega_1') \leq \kappa_1(\omega_2')$ iff $\kappa_2(\omega_1') \leq \kappa_2(\omega_2')$ for all $\omega_1', \omega_2' \in \mathit{Mod}(A)$. Since the subtraction of $\kappa_1(A)$ resp. $\kappa_2(A)$ does not change anything, for all $\omega_1', \omega_2' \in \mathit{Mod}(A)$ it is the case that $\kappa_1(\omega_1') - \kappa_1(A) \leq \kappa_1(\omega_2') - \kappa_1(A)$ iff $\kappa_2(\omega_1') - \kappa_2(A) \leq \kappa_2(\omega_2') - \kappa_2(A)$. Because we only consider models of A at this point, for $\omega_1', \omega_2' \in \mathit{Mod}(A)$ we get $\kappa_1|A(\omega_1') \leq \kappa_1|A(\omega_2')$ iff $\kappa_2|A(\omega_1') \leq \kappa_2|A(\omega_2')$. This is the case if $\kappa_1|A \models\!\approx (B|C)$ iff $\kappa_2|A \models\!\approx (B|C)$, leading to $C(\kappa_1|A) = C(\kappa_2|A)$ and hence to $\kappa_1|A \cong \kappa_2|A$.

For (OI) let κ' be the ranking function that is obtained by the conditionalization of κ by A, thus $\kappa'(\omega) = \kappa|A(\omega) = \kappa(\omega) - \kappa(A)$. By conditioning κ' with B we get $(\kappa|A)|B(\omega) = \kappa'(\omega) - \kappa'(B) = \kappa(\omega) - \kappa(A) - \kappa'(B)$ for models of AB. With $\kappa'' = \kappa|B$ we have $(\kappa|B)|A(\omega) = \kappa''(\omega) - \kappa''(A) = \kappa(\omega) - \kappa(B) - \kappa''(A)$ for models of AB. Since κ' is a ranking function over the models of A it is $\kappa'(B) = \min_{\{\omega \in \mathit{Mod}(A) | \omega \models B\}} \kappa(\omega) - \kappa(A) = \kappa(AB) - \kappa(A)$. This leads to $(\kappa|A)|B(\omega) = \kappa(\omega) - \kappa(A) - \kappa'(B) = \kappa(\omega) - \kappa(A) - \kappa(AB) + \kappa(A) = \kappa(\omega) - \kappa(AB)$ for models of AB. With an analogous calculation we get $\kappa''(A) = \kappa(AB) - \kappa(B)$ and $(\kappa|B)|A(\omega) = \kappa(\omega) - \kappa(AB)$ for models of AB. Thus, we get $(\kappa|A)|B(\omega) = (\kappa|B)|A(\omega)$ for all $\omega \models AB$.

Let κ be a ranking function as shown in Figure 4. The forgetting of b leads to the conditionalization of κ to $\overline{b}$, $\kappa^\circ_b = \kappa|\overline{b}$, and to a removal of all the most plausible worlds. As a result we get $(\overline{c}|a) \in C(\kappa^\circ_b)$ but $(\overline{c}|a) \notin C(\kappa)$ so that (W) is not fulfilled.

Conditionalization does not fulfill (wC). Let κ be a ranking function over $\Sigma = \{a, b\}$ with $\kappa(\overline{a}b) = 0$, $\kappa(a\overline{b}) = \kappa(ab) = 1$ and $\kappa(\overline{a}\,\overline{b}) = 2$. By conditionalizing κ to a we get $\kappa|a = \kappa^\circ_{\overline{a}}$ and $\kappa^\circ_{\overline{a}}(ab) = \kappa^\circ_{\overline{a}}(a\overline{b}) = 0$. This leads to a loss of the belief in b, because it is $b \in \mathit{Bel}(\kappa)|_{\{b\}} = \mathit{Cn}(\overline{a}b)|_{\{b\}} = \mathit{Cn}(b)$, but $b \notin \mathit{Bel}(\kappa^\circ_{\overline{a}}) = \mathit{Cn}(a)$.

The previous example also shows that conditionalization does not fulfill (sC). The conditionalization of κ to a leads to the belief in a, $a \in \mathit{Bel}(\kappa^\circ_{\overline{a}}) = \mathit{Cn}(a)$, whereas this belief is not present in the prior ranking function, $a \notin \mathit{Bel}(\kappa)|_{\{b\}} = \mathit{Cn}(\overline{a}b)|_{\{b\}} = \mathit{Cn}(b)$. Conditionalization does not fulfill (CP) because neither (wC) nor (sC) is fulfilled.

Figure 5 shows that conditionalization does not fulfill (wE). It is $\mathit{Bel}(\kappa_1) = \mathit{Bel}(\kappa_2) = \mathit{Cn}(\overline{a})$. By conditionalization of κ_1, κ_2 to a all worlds of the most plausible level are deleted. The worlds of the second plausibility level form the new beliefs, yielding $b \in Bel\,(\kappa_1|a) = Cn(ab)$, but $b \notin Bel\,(\kappa_2|a) = Cn(a\bar{b})$. □

7 Conclusion and Future Work

In this paper, we presented general axiomatic properties for forgetting operators on epistemic states that are inspired by postulates having been proposed in the framework of answer set programming (ASP). In order to capture both the formal setting and the intuition behind the ASP postulates suitably, we reinterpreted logical notions and their roles for ASP reasoning in a generic epistemic framework and used these formal correspondences to translate the postulates. Then, we chose three central forgetting operators presented in (Beierle et al. 2019) and evaluated them according to the epistemic postulates. The results of this evaluation (see Table 1) show that forgetting in ASP is very similar to marginalization that fulfills all the postulates, but very different from contraction which rather follow the ideas of AGM theory (Alchourrón, Gärdenfors, and Makinson 1985) and do not comply with any postulate. The similarity between ASP forgetting and marginalization is not surprising because both implement variable elimination. However, while marginalization occurs as a very simple and natural operation in (semi-) quantitative frameworks like probabilistics and ranking theory, forgetting in qualitative logical environments seems to demand for comprehensive investigations and theories (see, e.g., (Delgrande 2017)). General axiomatic properties as the ones we propose in this paper may be very helpful for making subtle differences between specific forgetting operators apparent, therefore we will continue our axiomatic evaluations on more forgetting operators.

The non-compliance of contraction with the postulates suggests that AGM-like changes are not suitable to implement forgetting in the ASP sense. Nevertheless, as Delgrande states (Delgrande 2017), multiple and more complex types of belief changes could achieve results analogous to forgetting in the sense of syntax reduction. This leaves much room to bring together the multiple kinds of forgetting in the common-sense understanding into one theory. It is part of our ongoing work to broaden the axiomatic framework of forgetting by also taking AGM-inspired postulates into account, and to evaluate more forgetting operators (e.g., from (Beierle et al. 2019)) according to it.

Conditionalization shows that there are more forgetting operators beyond variable elimination and contraction. Usually looked upon as a means to perform revision by assuming sentences A to hold, the forgetting aspects of conditionalization, i.e., its irreversible blinding out of all cases where A does not hold, are often overlooked. Our results show that forgetting by conditionalization is similar to both variable elimination and contraction, but with different respects.

The forgetting operators that we evaluated in this paper are all based on ranking functions (Spohn 1988), but our epistemic postulates are applicable to more general notions of epistemic states. The next step will be to study forgetting operators on epistemic states that are equipped with total preorders to implement plausibility relations. This epistemic setting is particularly useful in the areas of nonmonotonic reasoning and belief revision.

References

Alchourrón, C.; 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.

Beierle, C., and Kutsch, S. 2018. Computation and comparison of nonmonotonic skeptical inference relations induced by sets of ranking models for the realization of intelligent agents. *Applied Intelligence*. (to appear; online first 2018, June 23).

Beierle, C.; Kern-Isberner, G.; Sauerwald, K.; Bock, T.; and Ragni, M. 2019. Towards a general framework for kinds of forgetting in common-sense belief management. *KI - Künstliche Intelligenz*. (to appear; First Online 8 December 2018).

DeFinetti, B. 1974. *Theory of Probability*, volume 1,2. New York: John Wiley and Sons.

Delgrande, J. P. 2017. A knowledge level account of forgetting. *J. Artif. Intell. Res.* 60:1165–1213.

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

Gelfond, M., and Leone, N. 2002. Logic programming and knowledge representation – the A-prolog perspective. *Artificial Intelligence* 138:3–38.

Gonçalves, R.; Knorr, M.; and Leite, J. 2016. The ultimate guide to forgetting in answer set programming. In Baral, C.; Delgrande, J. P.; and Wolter, F., eds., *Principles of Knowledge Representation and Reasoning: Proceedings of the Fifteenth International Conference, KR 2016, Cape Town, South Africa, April 25-29, 2016.*, 135–144. AAAI Press.

Kern-Isberner, G.; Bock, T.; Sauerwald, K.; and Beierle, C. 2017. Iterated contraction of propositions and conditionals under the principle of conditional preservation. In Benzmüller, C.; Lisetti, C. L.; and Theobald, M., eds., *GCAI 2017, 3rd Global Conference on Artificial Intelligence, Miami, FL, USA, 18-22 October 2017.*, volume 50 of *EPiC Series in Computing*, 78–92. EasyChair.

Kern-Isberner, G. 2001. *Conditionals in nonmonotonic reasoning and belief revision.* Springer, Lecture Notes in Artificial Intelligence LNAI 2087.

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

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

Spohn, W. 1988. Ordinal conditional functions: a dynamic theory of epistemic states. In Harper, W., and Skyrms, B., eds., *Causation in Decision, Belief Change, and Statistics, II*. Kluwer Academic Publishers. 105–134.

Wong, K.-S. 2009. *Forgetting in Logic Programs*. Ph.D. Dissertation, University of New South Wales.