Künstliche Intelligenz manuscript No.
(will be inserted by the editor)

A Brief Survey on Forgetting from a Knowledge Representation and Reasoning Perspective

Thomas Eiter · Gabriele Kern-Isberner

Received: date / Accepted: date

Abstract Forgetting is an ambivalent concept of (human) intelligence. By definition, it is negatively related to knowledge in that knowledge is lost, be it deliberately or not, and therefore, forgetting has not received as much attention in the field of knowledge representation and reasoning (KRR) as other processes with a more positive orientation, like query answering, inference, or update. However, from a cognitive view, forgetting also has an ordering function in the human mind, suppressing information that is deemed irrelevant and improving cognitive capabilities to focus and deal only with relevant aspects of the problem under consideration. In this regard, forgetting is a crucial part of reasoning. This paper collects and surveys approaches to forgetting in the field of knowledge representation and reasoning, highlighting their roles in diverse tasks of knowledge processing, and elaborating on common techniques. We recall forgetting operations for propositional and predicate logic, as well as for answer set programming (as an important representative of nonmonotonic logics) and modal logics. We discuss forgetting in the context of (ir)relevance and (in)dependence, and explicit the role of forgetting for specific tasks of knowledge representation, showing its positive impact on solving KRR problems.

Keywords Forgetting · knowledge representation and reasoning · logic · answer set programming · nonmonononotonic reasoning · modal logics · interpolation · relevance · independence

This is a post-peer-review, pre-copyedit version of an article published in *KI – Künstliche Intelligenz*. The final authenticated version is available online at: https://doi.org/10.1007/s13218-018-0564-6.

Thomas Eiter
TU Wien, Institute of Logic and Computation
E-mail: eiter@kr.tuwien.ac.at

Gabriele Kern-Isberner
TU Dortmund, Department of Computer Science
E-mail: gabriele.kern-isberner@cs.tu-dortmund.de

1 Introduction

Forgetting is an important aspect of the human mind, which manifests itself in different forms and may serve different purposes. In a layman's view, forgetting can be seen as the process of omitting from one's memory information or knowledge in such a way that it is no longer present or reproducible. With an ever growing stream of information, bounded memory and short response time suggest that not all information can be kept and treated in the same way. From a cognitive perspective, forgetting is a gradual process in which information that is less used is moved to the background, from which it may eventually disappear; unless this happened, it may be recovered through remembering to the foreground (Ebbinghaus 1885). This basic mechanism helps us to deal with information overload and to put a focus of attention; notably, this is similarly reflected in common computer architectures.

Naturally, forgetting is thus an important aspect for systems of Artificial Intelligence (AI) that aim to model human reasoning capabilities, and in particular for the area of knowledge representation and reasoning (KRR) which aims at formalisms allowing one to process information and knowledge represented in symbolic form. Such formalisms go beyond plain data storages and cater for the possibility to connect pieces of information and make implicit information explicit through semantics evaluation. According reasoning services are of crucial importance in the emerging information society, which has become possible by the advent of the Internet and the World Wide Web. The need for forgetting in this context has already been widely acknowledged. Most notably, according to the EU Court of Justice, each person has the right to be forgotten[1] (see also the General Data Protection Regulation (GDPR) 2016/679[2] of the European Union (EU), which became recently enforceable as a binding regulation on data protection and privacy for all individuals within the EU and the European Economic Area). In particular, information about a person may be removed from certain webpages on the Internet. Such forgetting is *intentional*. While humans often forget unintentionally (but they may get aware about having forgotten something), intentional forgetting is also a crucial, but often neglected part of human reasoning. For instance, when focussing on a specific problem, humans blind out (i.e., forget) irrelevant aspects, or when trying to find a solution under restricted conditions, they have to intentionally forget ways of solving the problem in richer environments. These examples make clear that intentional forgetting in humans is a complex cognitive process that involves many aspects of knowledge and reasoning.

In this paper, we consider such operations for intentional forgetting in AI, and more specifically, from a KRR perspective. The latter focuses on logic-based methods and techniques, describing them at the surface without delving into the details of how forgetting operations work (in humans). Forgetting has been studied for many formalisms with similar motivations: reasoning should focus on essential aspects, and forgetting should help to simplify matters in the spirit of Ockham's Razor, which loosely speaking says "keep things simple whenever possible"[3] For instance, a model

[1] Judgment May 13, 2014, `http://curia.europa.eu/juris/documents.jsf?num=C-131/12`

[2] `https://eur-lex.europa.eu/legal-content/EN/ALL/?uri=CELEX:02016R0679-20160504`, enforceable beginning 25 May 2018

[3] In Latin "Entia non sunt mulitplicanda praeter necessitatem," attributed to William of Ockham, 14th c.

of a domain, an equational system describing interdependent variables, or explanations for observations should be as simple as possible to not introduce irrelevant details. From a richer information stance, omitting such details can be viewed as forgetting; thus, the latter is deeply connected to other central notions of knowledge representation like irrelevance and equivalence, and plays a crucial role for many reasoning tasks like, e.g., belief change. Notably, forgetting has to be distinguished from abstraction, which is an another important capability of humans to simplify matters: in abstraction, the language for describing information is changed, e.g., when red wines and white wines are merged to wines, while in forgetting the language remains the same or is restricted, e.g. when one forgets about white wines.

We shall consider knowledge bases *KB* that consist of formulas in one of various logics, without going into the technical details and full formal definitions; we deliberately remain superficially at the surface to convey the main ideas and emerging properties. The focus will be on classical propositional and first-order predicate logic, but we shall also touch nonmonotonic logic (viz. answer set programs) and modal logic.

As said above, forgetting aims at removing information from a knowledge base. For that, two major types of removal can be distinguished:

1. $\mathit{forget}(\mathit{KB},\Sigma')$ formalizes the operation where a part $\Sigma' \subseteq \Sigma$ of the signature (i.e., the vocabulary) Σ of the knowledge base *KB*, or in further refinement some formulas over Σ' are suppressed;
2. $\mathit{forget}(\mathit{KB},\phi)$ formalizes the operation of logically eliminating a formula ϕ from a knowledge base *KB*.

For example, removal of information about white wine from a knowledge base about food and drinks would be of type 1, while the removal that white wine goes well with fish of type 2. The forgetting operations, as described above, happen at the syntactic level; they have, however, often a semantic counterpart in terms of the models that the knowledge base should have after forgetting, This view gives rise to semantic characterizations (or alternative definitions) of forgetting. While the two types of forgetting are interrelated, it is important to be aware of the difference, and they should be not confused.

Most of the literature in KRR about forgetting is on type 1 forgetting. A very good overview on the history of forgetting in AI is given by van Ditmarsch et al. (2009), starting with Boole's variable marginalisation (Boole 1854). More recent works on forgetting in AI are by Lin and Reiter (1994), who considered forgetting of ground atoms and relations in a first-order setting; by Lang et al. (2003), who contributed an in-depth study on the computational costs of transforming theories by variable forgetting (via the link to independence); by Baral and Zhang (2005), who modeled forgetting as ignorance and considered more explicit operators for knowledge and belief; by Wang et al. (2005), who presented a forgetting operation for nonmonotonic logic programs; by Ghilardi et al. (2006a) and Wang et al. (2008), who were picking up forgetting in description logics; by Erdem and Ferraris (2007) who studied operations for forgetting of resp. abstracting from actions in planning; and by many others.

Type 2 forgetting is quite different from the works mentioned above; it amounts to the operation of *contraction* in belief change theories like the famous AGM theory Alchourrón et al. (1985). Here the signature of the knowledge base is not reduced, and roughly speaking the contraction version of $\mathit{forget}(KB, p)$ makes the atom p non-derivable, by adjusting the models of KB and adding further ones in which p is false, if needed; on the other hand, type 1 forgetting of p from the signature will not introduce any new models, but merely makes the existing ones indifferent on p.

The rest of this article is organized as follows. In the next section, we recall some basic notions of knowledge representation and reasoning as needed in the sequel. We then recall approaches to forgetting in different logic-based formalisms, starting with propositional logic (Section 3) and followed by predicate logic (Section 4), nonmononotonic logic (specifically, answer set programming, Section 5), and modal logics (Section 6). Afterwards we briefly discuss in Section 7 notions that are closely related to forgetting, and then consider in Section 8 some applications of forgetting and its relevance for problems in knowledge representation and reasoning. We conclude in Section 9 with a summary and discussion of a recent unifying framework by Delgrande (2014, 2017) for forgetting at the knowledge level, which nicely captures the essence of forgetting: this operation should not lose information about a domain per se, but rather lose the ability to express, or represent, information about the domain.

2 Elements of Knowledge Representation and Reasoning

The field of knowledge representation and reasoning deals with formal representation languages for knowledge and information, including beliefs, that have a clear formal (logical) semantics. The idea is that explicit beliefs are specified in a knowledge base from which implicit beliefs can derived (more or less) in a mechanical way that is amenable to automation. It is linked to the physical symbol system hypothesis by Newell and Simon (1976), which says that a physical symbol system [e.g., a computer] has the necessary and sufficient means for intelligent action, and to the knowledge principle of Lenat and Feigenbaum (1987), which states that if a program is to perform a complex task well, it must know a great deal about the world in which it operates.

The idea of reasoning in a mechanical way goes back at least to Leibniz, who developed the grand vision of the *"calculus ratiocinator"* as a general framework for solving a variety of problems such as philosophical disputes by reducing them to calculations; interestingly, Leibniz stressed already the prominent use of symbols in human reasoning (cf. the discussion in Peckhaus 2004). Based on the developments of formal logic that was laid out to provide solid foundation to mathematics, J. McCarthy suggested predicate logic as a tool for expressing and processing information in programs (McCarthy 1959), and discussed how to apply this in the context of planning (McCarthy 1963).

In a classical-logical environment, *knowledge bases* are sets KB of formulas ϕ constructed with a *signature* Σ. *Consequences* of a KB are determined by a relation $KB \vdash \phi$ for formulas ϕ derivable from KB, in a calculus; the set of all consequences of

KB is denoted by $Cn(KB)$. On the semantic side, *models* of KB are interpretations M of Σ in terms of suitable mathematical structures that satisfy KB (denoted $M \models KB$); the set of all models of KB is denoted by $Mod(KB)$. Desirably (and as is the case for standard calculi of predicate logic), $KB \vdash \phi$ coincides with $KB \models \phi$, i.e., derivability of ϕ from KB amounts to ϕ being true in all models M of KB.

Example 1 The simple knowledge base

$$KB = \{human(socrates), \forall x(human(x) \rightarrow mortal(x))\}$$

expresses in predicate logic that Socrates is human, and that all humans are mortal; a possible model of KB would be $M = \{human(socrates), mortal(socrates)\}$, which informally would say that Socrates is human and is mortal are both true; as a so-called *Herbrand model*, its domain of discourse would just consist of Socrates, while in general, any nonempty set can serve as this domain. As all models of KB must satisfy also $mortal(socrates)$ in the classical semantics of predicate logic, we have that $KB \models mortal(socrates)$. Accordingly we have $KB \vdash mortal(socrates)$ using inferences rules from predicate calculus: e.g., *specialization* of the implication yields $human(socrates) \rightarrow mortal(socrates)$, which together with $human(socrates)$ yields $mortal(socrates)$ by *modus ponens*.

However, reasoning with predicate logic encounters many challenges: First, $KB \models \phi$ is undecidable in general. Therefore, for effective inference algorithms, fragments of predicate logic have to be found, which for instance has led to the development of description logics (Baader et al. 2003). Second, for real-world applications, the capability of dealing with incomplete, inconsistent, and uncertain information is indispensable. Humans are amazingly capable of dealing with imperfect information, as the following example shows.

Example 2 The fact that humans are mortal is common knowledge; moreover most people are convinced that humans (usually) cannot fly. Nevertheless, San Giuseppe da Copertino was eyewitnessed to fly many times (cf. Figure 1); approved as a miracle, he was beatified by Pope Benedict XIV in 1753. People believed in this without giving up the (imperfect) belief that humans cannot fly, or that they are mortal.

The previous example shows a typical case of commonsense reasoning, as done by humans. It appeared that for a mathematical formalization, classical logic is inadequate due to its inherent *monotonicity*: if a knowledge base KB entails ϕ, then $KB' \models \phi$ for every KB' that extends KB. However, incomplete information, as in the previous example may require *non-monotonic inference*: if we know that Giuseppe da Copertino is human and that humans (normally) do not fly, then we conclude that he can not fly by default; however, when we learn that he indeed can fly, then we revert the conclusion. Technically, after augmenting KB with the fact $flies(giuseppe\_da\_copertino)$ to KB', we no longer can derive $\neg flies(giuseppe\_da\_copertino)$.

The field of *nonmonotonic reasoning (NMR)* has addressed this issue and many respective formalisms have been proposed (cf. Brewka et al. 2011b). Besides formalisms that extend classial logic by viewing incompleteness in qualitative terms, also quantitative frameworks that allow for nonmonotonic reasoning in a more fine-grained way have been provided by, e.g., probability theory (Pearl 1988), or fuzzy

Fig. 1: S. Giuseppe da Copertino in flight at the Basilica of Loreto (Ludovico Mazzanti, 18th c. source: Wikipedia)

logic (Klir and Yuan 1996). In this article, we focus on *answer set programming (ASP)* (Brewka et al. 2011a; Schaub and Woltran 2018), which has become increasingly popular due to availability of a language rich in constructs for declarative knowledge representation and the availability of efficient solvers for evaluation.

An ASP knowledge base (or program) consists of logical IF-THEN rules, in which incomplete information can be expressed via *default negation* (also known as *negation as failure*, due to historic reasons). For example, the program

$$\begin{aligned} & \mathit{student}(\mathit{sam}). \\ \mathit{male}(x) \vee \mathit{female}(x) \leftarrow {} & \mathit{student}(x). \\ \mathit{single}(x) \leftarrow {} & \mathit{student}(x), \mathit{not}\ \mathit{married}(x). \end{aligned} \tag{1}$$

represents knowledge about students and their sex and marital status. While the first two statements are self-explanatory, the third expresses a default rule: if a student is not known to be married, then s/he is concluded to be single.

The semantics of an ASP program can be defined in terms of special models ("answer sets") of the rules cast to classical formulas, where $\leftarrow$ is turned into logical (material) implication $\rightarrow$ and *not* into negation $\neg$. The program above has two answer sets, viz. $M_1 = \{\mathit{student}(\mathit{sam}), \mathit{single}(\mathit{sam}), \mathit{male}(\mathit{sam})\}$, and $M_2 = \{\mathit{student}(\mathit{sam}),$ $\mathit{single}(\mathit{sam}),$
$\mathit{female}(\mathit{sam})\}$. Accordingly, we infer from them that Sam is single, but we remain agnostic about Sam's sex, as neither $\mathit{male}(\mathit{sam})$ nor $\mathit{female}(\mathit{sam})$ is true in every answer set.

Non-monotonic inference relations $KB \models_n \phi$ lack well-known properties of classical logic in general; in particular, there is *no left strengthening*, i.e., $KB \models_n \phi$ does not imply $KB \cup \{\psi\} \models_n \phi$. In the above example, we may have $KB \models_n \mathit{single}(\mathit{sam})$ but $KB \cup \{\mathit{married}(\mathit{sam})\} \not\models_n \mathit{single}(\mathit{sam})$. Moreover, *transitivity fails*: $KB \models_n \phi$ and $\phi \models_n \psi$ (where ϕ is viewed as a knowledge base) does not imply $KB \models_n \psi$. For example, in the above example we have $KB \models_n \mathit{married}(\mathit{sam}) \leftarrow \mathit{not}\ \mathit{single}(\mathit{sam})$ and $\mathit{married}(\mathit{sam}) \leftarrow \mathit{not}\ \mathit{single}(\mathit{sam}) \models_n \mathit{married}(\mathit{sam})$, but $KB \not\models_n \mathit{married}(\mathit{sam})$.

Another extension of classic logic that is important for knowledge representation are modal logics, which add *modal operators* such as $\Box\phi$, $\Diamond\phi$ to express that ϕ is necessarily (always) true, respectively possibly (sometimes) true. The use of modal logic for expressing knowledge and belief has a long tradition, going back to the influential work of Hintikka (1962); the book of Fagin et al. (1995) is a popular source on the topic. In particular, the operator $K\phi$ is frequently used in order to express that intuitively ϕ is known to be true by the agent, where, e.g., $\phi =$ "my car is parked at Amnesia lane". The *epistemic state* of an agent might include besides ordinary formulas ϕ also formulas like $K\phi$, $\neg K\phi$, or $K\neg K\phi$, where the latter expresses introspection: the agent knows that it does not know whether ϕ holds. For reasoning in modal logic, inference systems are extended with further axioms that give rise to families of modal logic.

For example, if

$$KB = \{\ \mathit{inflagranti} \rightarrow \mathit{guilty},\ \ K\mathit{guilty} \vee K\neg \mathit{guilty} \rightarrow \mathit{justice}, \mathit{inflagranti}\}$$

is a representation of an agent's epistemic state, then *KB* entails *justice*: from the first implication and the fact *inflagranti* we conclude *guilty*, from which in turn we conclude *Kguilty* by *necessitation*;[4] then, by the second implication, we conclude *justice*.

Semantically, modal logics can be typically characterized in terms of Kripke models (Kripke 1959, 1963), which are possible world structures with classical interpretations at each world and transition relations between worlds.

While modal logics are monotonic, nonmonotonic variants have been considered in order to formalize commonsense reasoning; in particular, answer set semantics can be characterized in terms of nonmonotonic modal logic; we refer the interested reader to (Marek and Truszczyński 1993; Cabalar et al. 2018).

3 Forgetting in Propositional Logic

We start our survey with forgetting in propositional logic, where this operation can be traced back to the early days of formal logics. In his seminal work, Boole (1854) dealt with *exclusion of middle terms* (aka *variable elimination*) as an operation that aims at omitting variables from propositional formulas that are indefinite, or deemed to be irrelevant for a particular context.

Example 3 Assume we have a plain knowledge base about food and drinks, given by

$$KB = \{\mathit{red\_wine}\ \vee\ \mathit{white\_wine}, \mathit{fish} \rightarrow \mathit{white\_wine}, \mathit{beef} \rightarrow \mathit{red\_wine}\};$$

informally, either white wine or red wine should be served, and fish calls for white wine while beef calls for read wine. We now would like to forget about $fish$ because, e.g., we don't care about fish.

[4] The rule of necessitation says that when a formula ϕ is derived, we can also derive $K\phi$. The modality K is here viewed akin to $\Box$, for which this rule had been originally introduced.

A respective forgetting operation should yield a new (syntactic) knowledge base KB' that describes the result, in which the variable *fish* should no longer occur; clearly, we have to elaborate on the change of meaning for KB' compared to KB. The key idea is to consider the possible valuations of the variable *fish* and to conditionalize the knowledge base KB on them; this can be achieved by plugging in the possible truth values one by one and joining the results. Let us assume for simplicity that KB consists of a single formula ϕ (we may simply conjoin all formulas in KB).

Definition 1 (Propositional Forgetting by Variable Elimination; cf. Boole, 1854)

For a propositional knowledge base of form $KB = \{\phi\}$ and a variable p, we let

$$forget(KB, p) = \{\phi_p^+ \vee \phi_p^-\}$$

where ϕ_p^+ (resp. ϕ_p^-) is ϕ with p replaced by $\top$ ='true' (resp., $\bot$ = 'false').

Lin (2001) observed that this notion of forgetting has been re-discovered several times, e.g. in the database context (Weber 1986); it appeared in AI then in the landmark papers (Lin and Reiter 1994; Lang et al. 2003). Let us consider our example again.

Example 4 (Ex. 3, cont'd) We rewrite our knowledge base to $KB = \{\phi\}$, where $\phi = (r \vee w) \wedge (f \rightarrow w) \wedge (b \rightarrow r)$, where we use the abbreviations r, w, f, b for *red_wine*, *white_wine*, *fish*, *beef*, respectively. Then,

$$\begin{aligned} forget(KB, f) &= \{\, \phi_f^+ \vee \phi_f^- \,\} \\ &= \{\, \big[(r \vee w) \wedge (\top \rightarrow w) \wedge (b \rightarrow r)\big] \vee \big[(r \vee w) \wedge (\bot \rightarrow w) \wedge (b \rightarrow r)\big] \} \\ &= \{\, \big[(r \vee w) \wedge w \wedge (b \rightarrow r)\big] \vee \big[(r \vee w) \wedge (b \rightarrow r)\big] \,\} \\ &= \{\, (r \vee w) \wedge (b \rightarrow r) \,\}. \end{aligned}$$

Here, "=" on the second and the following lines means logical equivalence, where the latter result by rewriting using laws of Boolean logic.

After forgetting, the semantics of a knowledge base has changed, too, but we expect the models of the posterior knowledge base to be similar to those of the prior knowledge base. More precisely, semantic changes should affect only the forgotten variable p. This gives rise to the following notion of similarity $\sim_p$ defined between models M_1 and M_2:

$$M_1 \sim_p M_2 \Leftrightarrow M_1 \text{ and } M_2 \text{ agree on all variables except possibly } p$$

In propositional logic, models respectively interpretations M assign each variable one of the Boolean values $true$ or $false$; as common, we view M as the set of variables that are assigned true in it. In our example above, e.g. $M_1 = \{r, w, f\}$ is a model of KB, and for $M_2 = \{r, w\}$ we have $M_1 \sim_f M_2$. The models of a knowledge base after forgetting are then as follows.

Proposition 1 (Semantic Characterization of Forgetting) *For a knowledge base $KB = \{\phi\}$ on signature Σ and a variable $p \in \Sigma$, we have for $\Sigma' = \Sigma \setminus \{p\}$ that*

(i) $Mod_\Sigma(forget(KB, p)) = \{M_1 \mid M_1 \sim_p M_2 \text{ for some } M_2 \in Mod_\Sigma(KB)\}$,

(ii) $Mod_{\Sigma'}(forget(KB,p)) = \{M|_{\Sigma'} \mid M \in Mod_{\Sigma}(KB)\}$.

(Here the subscript of *Mod* indicates the signature of concern.) That is, (i) purges all information on p in models of *KB* while (ii) projects p off.

Example 5 (Ex. 4, cont'd) We apply Proposition 1 to Example 4. In this case, we have $\Sigma' = \Sigma \setminus \{f\}$ and $Mod(KB) = \{\{r\},\{w\}, \{r,w\},\{b,r\},\{b,r,w\},\{f,w\}, \{f,w,r\}, \{f,b,w,r\}\}$. Proposition 1 yields

(i) $Mod_{\Sigma}(forget(KB,f)) = Mod(KB) \cup \{\{f,r\},\{f,b,r\}\}$, and
(ii) $Mod_{\Sigma'}(forget(KB,f)) = \{\{r\},\{w\},\{r,w\},\{b,r\},\{b,r,w\}\}$.

A disadvantage of this forgetting operation is that *KB* must be a single formula. We can easily extend this to a finite set of formulas $KB = \{\phi_1,\ldots,\phi_n\}$, for which $forget(KB,p)$ is (by taking $\phi = \phi_1 \wedge \cdots \wedge \phi_n$ as described) logically equivalent to $\{\phi_{i_p}^{+} \vee \phi_{j_p}^{-} \mid 1 \leq i,j \leq n\}$; moreover, we can further extend this to arbitrary (possibly infinite) knowledge bases *KB* by setting $forget(KB,p) = \{\phi_p^+ \vee \psi_p^- \mid \phi, \psi \in KB\}$; the semantic characterization in Proposition 1 remains valid.

Proposition 2 (Semantic Characterization of Forgetting (Extended)) *For any (possibly infinite) knowledge base KB on signature Σ and a variable $p \in \Sigma$, we have for $\Sigma' = \Sigma \setminus \{p\}$ that*

(i) $Mod_{\Sigma}(forget(KB,p)) = \{M_1 \mid M_1 \sim_p M_2 \text{ for some } M_2 \in Mod_{\Sigma}(KB)\}$,
(ii) $Mod_{\Sigma'}(forget(KB,p)) = \{M|_{\Sigma'} \mid M \in Mod_{\Sigma}(KB)\}$.

(We omit proofs in this survey, but the interested reader is invited to check this result which does not require advanced skills.) Simplifications may be applied, e.g. if p does not occur in $\phi \in KB$, we can just leave it untouched, i.e., we have $forget(KB,p) = forget(KB \setminus \{\phi\},p) \cup \{\phi\}$.

By resorting to a special form of knowledge bases, alternatives such as the following are viable. A knowledge base is in *clausal form* (or *conjunctive normal form*), if all its formulas are clauses, i.e., disjunctions $\gamma = \ell_1 \vee \cdots \vee \ell_m$ of literals, which are variables p or their negations $\neg p$.

Definition 2 (Propositional Forgetting, Alternative; Delgrande, 2014) For any propositional *KB* and any propositional atom p, define

$$forget'(KB,p) = \{\gamma \in CNF(KB) \mid p \text{ does not occur in } \gamma\} \cup Res(CNF(KB),p)$$

where $CNF(KB)$ is an arbitrary clausal form of *KB*, and $Res(S,p)$ is the set of all *resolvents* of clauses in *KB* w.r.t. p, i.e., $Res(S,p) = \{\alpha \vee \beta \mid \alpha \vee p, \beta \vee \neg p \in S\}$.

This operation also works for infinite knowledge bases *KB*, as every formula can be converted using Boolean laws into an equivalent set of clauses and thus $CNF(KB)$ always exists; however this conversion may cause an exponential blowup in general.

Example 6 In Example 3, *KB* is easily converted to clausal form by turning the implications $\alpha \to \beta$ into clauses $\neg\alpha \vee \beta$; thus we obtain $CNF(KB) = \{r \vee w, \neg f \vee w, \neg b \vee r\}$. As f occurs only in a single clause, $Res(CNF(KB),f) = \emptyset$ and thus

$$forget'(KB,f) = \{r \vee w, \neg b \vee r\} \cup Res(CNF(KB),f) \;=\; \{r \vee w, \neg b \vee r\}.$$

There exists another characterization of forgetting in terms of clauses that does not refer to the syntax of the formulas in the knowledge base, but to its logical consequences. A clause $\gamma \neq \top$ is a *prime implicate* of a knowledge base *KB*, if $KB \models \gamma$ and $KB \not\models \gamma'$ for every proper subclause γ' of γ that results by omitting some literals.

Proposition 3 (Prime Implicate Characterization; cf. Lin 2001; Lang et al., 2003) *For any propositional KB, and propositional atom p,*

$$forget(KB, p) = \{\gamma \mid \gamma \textit{ is a prime implicate of KB s.t. p does not occur in } \gamma\}.$$

Example 7 (Ex. 5, cont'd) Recall that the knowledge base in Example 3 can be viewed in clausal form as $CNF(KB) = \{r \vee w, \neg f \vee w, \neg b \vee r\}$. Each of the clauses in this set is in fact a prime implicate of *KB*; and it appears that no further prime implicate exists. Thus, according to Proposition 3, we obtain that $forget(KB, f) = \{r \vee w, \neg b \vee r\}$, which coincides with $forget'(KB, f)$.

In order to forget multiple variables, the operation $forget(KB, p)$ can be simply iterated:

- $forget(forget(KB, p_1), p_2)$ forgets first p_1, then p_2;
- for any sequence $\sigma = p_1, p_2, \ldots, p_k$ of variables, let

$$forget(KB, \sigma) = forget(forget(KB, \sigma'), p_k)$$

  where $\sigma' = p_1, \ldots, p_{k-1}$ and $forget(KB, \varepsilon) = KB$ (with ε the empty sequence).

Example 8 (Ex. 4, cont'd) We continue Example 4, where $KB = \{\phi\}$, and $\phi = (r \vee w) \wedge (f \rightarrow w) \wedge (b \rightarrow r)$. Let us forget all food with $\sigma = f, b$; then

$$\begin{aligned} forget(forget(KB, f), b) &= \{forget(\{(r \vee w) \wedge (b \rightarrow r)\}, b)\} \\ &= \{[(r \vee w) \wedge (\top \rightarrow r)] \vee [(r \vee w) \wedge (\bot \rightarrow r)]\} \\ &= \{r \vee w\}. \end{aligned}$$

Notably, if we change the order of food to $\sigma = b, f$ then $forget(forget(KB, b), f)$ yields the same (logically equivalent) result.

That the order of variables does not matter in the previous example is not by accident; it is one of the important properties of iterated forgetting. The seminal paper of Lin and Reiter (1994) contains, in explicit form or as easy consequences, this and the following properties of forgetting.

Proposition 4 (Properties of Variable Forgetting; Lin and Reiter, 1994) *For any KB and sequence* $\sigma = p_1, \ldots, p_k$ *of variables,*

(i) for any permutation σ' *of* σ*, it holds that* $forget(KB, \sigma)$ *and* $forget(KB, \sigma')$ *are logically equivalent, i.e.,*
$Mod(forget(KB, \sigma)) = Mod(forget(KB, \sigma'))$ *(order independence)*

(ii) $KB \models forget(KB, \sigma)$ *(over-approximation)*

(iii) if ψ *is a formula in which no* p_i *occurs,*
$KB \models \psi \Leftrightarrow forget(KB, \sigma) \models \psi$ *(consequence invariance)*

Thus, by (i) forgetting a set V of variables boils down to iterative forgetting, and from the semantic view, we can simply write $\mathit{forget}(KB,V)$. By (ii) all models of KB are preserved but new ones may be added, and by (iii) the consequences not mentioning any forgotten variable do not change. However, $\mathit{forget}(KB,\sigma)$ may grow exponentially, even if we use a variant of $\mathit{forget}(KB,p)$ that simplifies respectively rewrites the result in polynomial time: after forgetting all variables $p_1,\ldots,p_n$ from KB, the knowledge base is variable-free and thus either a tautology (if KB is satisfiable) or a contradiction (if KB is unsatisfiable); thus some intermediate $\mathit{forget}(KB,p_1,\ldots,p_i)$ must be exponential in the size of KB, unless $NP = P$.

Variable forgetting aims at eliminating any occurrence of a variable. A more fine-grained extension considered by Lang et al. (2003) is to forget literals, where an atom p and its negation $\neg p$ are distinguished. Let for any formula ϕ, literal l and $b \in \{0,1\}$ the expression $\phi_{l\leftarrow b}$ stand for $\phi_{p\leftarrow b}$ (resp. $\phi_{p\leftarrow 1-b}$) if l is a positive literal $l = p$ (resp. negative literal $l = \neg p$)."

Definition 3 (Propositional Literal Forgetting; Lang et al., 2003) For a propositional $KB = \{\phi\}$, a literal l, and a set $L = \{l_1,\ldots,l_k\}$ of literals, let

- $\mathit{forgetLit}(KB,l) = \{\phi|_{l\leftarrow 1} \vee (\neg l \wedge \phi)\}$, and
- $\mathit{forgetLit}(KB,L) = \mathit{forgetLit}(\mathit{forgetLit}(KB,L\setminus\{l\}),l)$ for any literal $l \in L$, where $\mathit{forgetLit}(KB,\emptyset) = KB$.

This definition is sound, as is does semantically not matter which literal l from L is chosen, i.e., as above we have order independence. Furthermore, one could equivalently replace in $\mathit{forgetLit}(KB,l)$ the disjunct $\neg l \wedge \phi$ by $\neg l \wedge \phi|_{l\leftarrow 0}$. Let us consider how literal forgetting works on our food and drink knowledge base.

Example 9 (Ex. 4, cont'd) We continue Example 4 with $KB = \{\phi\}$ and $\phi = (r \vee w) \wedge (f \rightarrow w) \wedge (b \rightarrow r)$, and forget the literal $l = \mathit{fish}$ rather than the variable $p = \mathit{fish}$:

$$\begin{aligned}
\mathit{forgetLit}(KB,f) &= \{\phi|_{f\leftarrow 1} \vee (\neg f \wedge \phi)\}\\
&= \{[(r\vee w)\wedge(\top \rightarrow w)\wedge(b\rightarrow r)] \vee [\neg f \wedge (r\vee w)\wedge(f\rightarrow w)\wedge(b\rightarrow r)]\}\\
&= \{[w\wedge(b\rightarrow r)] \vee [\neg f\wedge(r\vee w)\wedge(b\rightarrow r)]\}\\
&= \{[(w\vee\neg f\wedge r\vee\neg f\wedge w]\wedge(b\rightarrow r)\}\\
&= \{(w\vee\neg f\wedge r)\wedge(b\rightarrow r)\}.
\end{aligned}$$

Notably, the result is different from $\mathit{forget}(KB,f)$; it still mentions f, but in a negative occurrence. Semantically, $\mathit{forgetLit}(KB,f)$ has less models than $\mathit{forget}(KB,f)$).

This last observation is not by accident, and the models of $\mathit{forgetLit}(KB,L)$ always form a subset of the models of $\mathit{forget}(KB,L)$. Semantically, literal forgetting can be characterized in terms of model similarity as follows.

Proposition 5 (Semantic Characterization of Literal Forgetting; Lang et al., 2003) *For any knowledge base KB and set L of literals, the models of* $\mathit{forgetLit}(KB,L)$ *are expressed by*

$$\begin{aligned}
&\mathit{Mod}(\mathit{forgetLit}(KB,L)) =\\
&\quad \{M_1 \mid M_1 \sim_{\{p_1,\ldots,p_{k'}\}} M_2,\\
&\qquad L' = \{(\neg)p_1,\ldots(\neg)p_{k'}\} \subseteq L,\ M_2 \models KB \cup L'\}.
\end{aligned}$$

That is, all models M_1 are collected such that by changing the truth values of some atoms p_i (if needed) to satisfy all l_i, we obtain a model M_2 of *KB*. Note that M_1 is a model of $\mathit{forget}(\mathit{KB}, \{p_1, \ldots, p_{k'}\})$, and thus clearly of $\mathit{forget}(\mathit{KB}, V)$ where V are all variables occurring in L. Thus as said above, literal forgetting is stronger than variable forgetting. Further characterisations are possible; e.g. the one for variable forgetting in Proposition 3 extends to literal forgetting.

Among the properties of literal forgetting, the following ones are particularly noticable.

Proposition 6 (Properties of Literal Forgetting; Lang et al., 2003) *For any* $\mathit{KB} = \{\phi\}$ *and set L of literals, the following properties hold:*

(i) $\mathit{KB} \models \mathit{forgetLit}(\mathit{KB}, L)$,
(ii) *if* $L \subseteq L'$ *then* $\mathit{forgetLit}(\mathit{KB}, L) \models \mathit{forgetLit}(\mathit{KB}, L')$,
(iii) *if* $\mathit{KB} \models \mathit{KB}'$ *then* $\mathit{forgetLit}(\mathit{KB}, L) \models \mathit{forgetLit}(\mathit{KB}', L)$,
(iv) *if* $L = \{p, \neg p\}$, *then* $\mathit{forgetLit}(\mathit{KB}, L)$ *is equivalent to* $\mathit{forget}(\mathit{KB}, p)$,
(v) *if KB is of the form* $\mathit{KB} = \mathit{KB}_1 \vee \mathit{KB}_2$, *i.e.,* $\phi = \phi_1 \vee \phi_2$ *where* $\mathit{KB}_1 = \{\phi_1\}$ *and* $\mathit{KB}_2 = \{\phi_2\}$,[5] *then* $\mathit{forgetLit}(\mathit{KB}, L) = forget(\mathit{KB}_1, L) \vee \mathit{forget}(\mathit{KB}_2, L)$

Item (i) ensures that all models of *KB* are preserved, and thus $\mathit{forgetLit}(\mathit{KB}, L)$ is an over-approximation of *KB*; item (ii) ensures that the latter increases if more literals are forgotten, and item (iii) that it decreases if *KB* is logically strengthened (e.g. by adding formulas). Next, by item (iv) forgetting a variable amounts to forgetting both its positive and negative literal form. Finally, (v) states that forgetting literals commutes with disjunction, which allows us to (iteratively) decompose formulas for forgetting. Item (iii), which is not explicit in (Lang et al. 2003), is in fact an easy consequence of (v).

Example 10 (Ex. 9, cont'd) We reconsider the knowledge base *KB* from Example 4 and forget both f and $\neg f$, i.e., $L = \{f, \neg f\}$; then

$$\begin{aligned}
\mathit{forgetLit}(\mathit{KB}, \{\neg f, f\}) &= \mathit{forgetLit}(\mathit{forgetLit}(\mathit{KB}, f), \neg f) \\
&= \mathit{forgetLit}(\phi, \neg f), \;\text{ where }\; \phi = (w \vee \neg f \wedge r) \wedge (b \to r) \\
&= \{\phi|_{\neg f \leftarrow 1} \vee \neg\neg f \wedge \phi\} \\
&= \{[(w \vee \top \wedge r) \wedge (b \to r)] \vee [f \wedge (w \vee \neg f \wedge r) \wedge (b \to r)] \\
&= \{[(w \vee r) \wedge (b \to r)] \vee [f \wedge w \wedge (b \to r)] \\
&= \{(w \vee r \vee f \wedge w) \wedge (b \to r) \\
&= \{(w \vee r) \wedge (b \to r)\}.
\end{aligned}$$

Thus, the result of $\mathit{forgetLit}(\mathit{KB}, \{\neg f, f\})$ is the same as of $\mathit{forget}(\mathit{KB}, f)$ in Example 4, in line with Proposition 6.

For more information on variable and literal forgetting, we refer to (Lang et al. 2003).

The forgetting operations above fall into type 1 of the classification in the Introduction, which aims at suppressing certain formulas over part of the signature. The

[5] Here $\mathit{KB}_1 \vee \mathit{KB}_2 = \{\phi \vee \psi \mid \phi \in \mathit{KB}_1, \psi \in \mathit{KB}_2\}$.

type 2 forgetting operations are concerned with eliminating formulas from a knowledge base, such that they are no longer among the consequences that can be derived from it. This is the very subject of the area of Belief Change, in which Alchourrón et al. (1985) presented a theoretical framework for belief change in a landmark paper. AGM theory considers change operations on deductively closed theories like revision (adopting beliefs) and contraction (giving up beliefs) and their interrelationships, describing by sets of postulates what they deem to be a rational belief change. For example, AGM contractions should be successful, i.e., the belief to be contracted cannot be derived after contraction, and the result of contraction should not depend upon the specific syntactic form of the belief to be given up. The basic form of AGM contraction deals with forgetting a singleton formula, more general forms aiming to forget a set of formulas have also been developed (for a good introduction to so-called multiple contraction, cf., e.g., (Fuhrmann and Hansson 1994)). Likewise, belief contraction has also been considered for belief bases (which are not necessarily deductively closed) (Hansson 1999), and thus are closer to the knowledge bases considered here, or for epistemic states (Konieczny and Pino Pérez 2017; Kern-Isberner et al. 2017).

In order to illustrate the difference between contraction and forgetting, consider the knowledge base $KB = \{w \vee r,\ f \rightarrow w,\ b \rightarrow r\}$. AGM contraction of f from KB would not change KB at all because f cannot be entailed from KB, whereas forgetting the variable f would result in $KB' = \{w \vee r,\ b \rightarrow r\}$. This can also be achieved by contraction via contracting KB with, e.g., $f \rightarrow w$, by simply dropping the formula. However, while in this case, forgetting the variable f yields the same result, this is no longer so if KB would contain in addition $\neg f \rightarrow w$: then KB would entail w and while forgetting f would leave KB semantically unchanged, contracting $f \rightarrow w$ from KB would have to make w non-entailed.

Therefore, it seems that the relationship between forgetting and contraction is complicated and needs further investigations. As in the context of KRR, forgetting typically refers to type 1 operations, we focus on this type in the following.

4 Forgetting in Predicate Logic

In this section, we turn to forgetting in the setting of first-order predicate logic (FOL), where we shall first consider how to forget facts, relations and individuals in a general setting, and then look into description logics as a premier family of fragments of FOL for decidable reasoning.

4.1 Forgetting facts

Lin and Reiter (1994) considered forgetting in the context of FOL; more precisely, they considered forgetting *facts* $p = P(\mathbf{t})$, where $\mathbf{t} = t_1, \ldots, t_k$ is a tuple of terms, and *relations (predicates) P* from a (finite) knowledge base KB.

Example 11 (Lin and Reiter, 1994) Suppose we have the plain knowledge base $KB = \{student(John) \vee student(Joe) \vee teacher(John)\}$ and we want to forget about the fact

$p = \mathit{student}(\mathit{John})$ respectively the predicate $P = \mathit{student}$. In the following, we may shorten *student* to *stud* and *teacher* to *teach*.

The key idea is to view propositional forgetting as a special case: a propositional variable p amounts to a 0-ary predicate P (i.e., $p = P(\mathbf{t})$ where $\mathbf{t}$ is void), and one lifts variable elimination to first-order logic. However, a naive syntactic replacement of $P(\mathbf{t})$ by its valuation to true resp. false is not guaranteed to achieve semantic indifference, as expressed by similarity relations between models; this is because $\mathbf{t}$ may be accessed indirectly in the evaluation of atoms over P.

For example, forgetting $P(a)$ from $\phi = \forall x.P(x)$ should yield models in which $P(x)$ is true for all elements x different from a, for which the value of P should remain open; however, as $P(a)$ does not occur in ϕ, a replacement as in Definition 1 would leave ϕ semantically unchanged.

For this reason, we replace in ϕ every atom of the form $P(\mathbf{t}')$ by the logically equivalent formula $[\mathbf{t}' = \mathbf{t} \wedge P(\mathbf{t})] \vee [\mathbf{t}' \neq \mathbf{t} \wedge P(\mathbf{t}')]$, where $\mathbf{t}' = \mathbf{t}$ stands for $t'_1 = t_1 \wedge \cdots \wedge t'_k = t_k$ and $\mathbf{t}' \neq \mathbf{t}$ is $\neg(\mathbf{t}' = \mathbf{t})$; denote then by $\phi[p]$ the formula obtained in this way. Armed with this notation, we can then define forgetting of a fact as follows.

Definition 4 (Fact Forgetting; Lin and Reiter, 1994) For every FOL knowledge base $\mathit{KB} = \{\phi\}$ and variable-free fact $p = P(\mathbf{t})$, we let

$$\mathit{forget}(\mathit{KB}, p) = \{\phi_p^+ \vee \phi_p^-\}$$

where ϕ_p^+ (resp. ϕ_p^-) is $\phi[p]$ with p replaced by $\top$ ='true' (resp., $\bot$ = 'false').

Let us consider some examples from Lin and Reiter (1994).

Example 12 Recall $\mathit{KB} = \{\phi\}$ with $\phi = \mathit{stud}(\mathit{John}) \vee \mathit{stud}(\mathit{Joe}) \vee \mathit{teach}(\mathit{John})$ from Example 11, and consider $p = \mathit{stud}(\mathit{John})$. From Definition 4, we obtain

$$\begin{aligned} \phi[\mathit{stud}(\mathit{John})] = {} & [(\mathit{John} = \mathit{John} \wedge \mathit{stud}(\mathit{John})) \vee\ (\mathit{John} \neq \mathit{John} \wedge \mathit{stud}(\mathit{John}))] \\ & \vee [(\mathit{Joe} = \mathit{John} \wedge \mathit{stud}(\mathit{John})) \vee\ (\mathit{Joe} \neq \mathit{John} \wedge \mathit{stud}(\mathit{Joe}))] \\ & \vee \mathit{teach}(\mathit{John}); \end{aligned}$$

As $\mathit{John} = \mathit{John} = \top$, $\mathit{John} \neq \mathit{John} = \bot$, $\mathit{John} = \mathit{Joe} = \bot$, $\mathit{John} \neq \mathit{Joe} = \top$, we obtain

$$\begin{aligned} \mathit{forget}(\mathit{KB}, \mathit{stud}(\mathit{John})) &= \phi^+_{\mathit{stud}(\mathit{John})} \vee \phi^-_{\mathit{stud}(\mathit{John})} \\ &= ([(\top \wedge \top) \vee (\bot \wedge \top)] \vee [(\bot \wedge \top) \vee (\top \wedge \top] \vee \mathit{teach}(\mathit{John})) \\ &\quad \vee ([(\top \wedge \bot) \vee (\bot \wedge \bot)] \vee [(\bot \wedge \bot) \vee (\top \wedge \bot] \vee \mathit{teach}(\mathit{John})) \\ &= \top. \end{aligned}$$

Similarly, for $\mathit{KB} = \{\phi\}$ and $\phi = \exists x.\mathit{student}(x)$, forgetting $\mathit{student}(\mathit{John})$ yields

$$\begin{aligned} \phi[\mathit{stud}(\mathit{John})] &= \exists x.[x = \mathit{John} \wedge \mathit{stud}(\mathit{John})] \vee [x \neq \mathit{John} \wedge \mathit{stud}(x)] \\ \mathit{forget}(\mathit{KB}, \mathit{stud}(\mathit{John})) &= \exists x.([x = \mathit{John} \wedge \top] \vee [x \neq \mathit{John} \wedge \mathit{stud}(x)]) \vee \\ &\quad\ \exists x.([x = \mathit{John} \wedge \bot] \vee [x \neq \mathit{John} \wedge \mathit{stud}(x)]) \\ &= \top. \end{aligned}$$

Finally, forgetting $student(John)$ from $KB = \{\phi\}$ where $\phi = \forall x.student(x)$ is exactly the scenario discussed before Definition 4; wo obtain

$$\begin{aligned}
\phi[stud(John)] &= \forall x.[x = John \land stud(John)] \lor [x \neq John \land stud(x)], \\
forget(KB, stud(John)) &= \forall x.([x = John \land \top] \lor [x \neq John \land stud(x)]) \lor \\
&\quad\ \forall x.([x = John \land \bot] \lor [x \neq John \land stud(x)]) \\
&= \forall x.(x = John \lor stud(x)) \lor \forall x.(x \neq John \land stud(x)), \\
&= \forall x.[x \neq John \to stud(x)],
\end{aligned}$$

which expresses the intuitive result mentioned in the discussion.

Like in the propositional case, we can forget a sequence $\sigma = p_1, \ldots, p_k$ of facts by iteration, where again the order does not matter. For a semantic characterization of forgetting multiple facts, we have to lift the similarity notion $M_1 \sim_p M_2$ between propositional models to first-order models M_1, M_2 (this is straightforward) and by extending it from a single fact p to a sequence σ as follows:

$M_1 \sim_\sigma M_2 \Leftrightarrow$ models M_1, M_2 fully agree except possibly
for the valuation of the p_i in σ.

We then note the following properties of forgetting facts, where for any sentence ψ and sequence $\sigma = p_1, \ldots,\ p_k$ of variable-free facts, $\psi \downarrow \sigma$ denotes the formula obtained by replacing iteratively in ψ for $p_i = P_i(\mathbf{t_i})$, $i = 1, \ldots, k$, every occurrence of an atom $P_i(\mathbf{t'_i})$ with $P_i(\mathbf{t'_i}) \land \mathbf{t_i} \neq \mathbf{t'_i}$.

Proposition 7 (Semantic Properties) *Lin and Reiter, 1994] For any FOL knowledge base $KB = \{\phi\}$ and sequence $\sigma = p_1, \ldots,\ p_k$ of facts, the following properties hold:*

(i) $KB \models forget(KB, \sigma)$;
(ii) for any permutation σ' of σ, $forget(KB, \sigma')$; is equivalent to $forget(KB, \sigma)$;
(iii) $Mod(forget(KB, \sigma)) = \{M_1 \mid M_1 \sim_\sigma M_2$ for some $M_2 \in Mod(KB)\}$;
(iv) for every sentence ψ, $KB \models \psi \downarrow \sigma \Leftrightarrow forget(KB, \sigma) \models \psi \downarrow \sigma$.

Thus, forgetting is by (i) an over-approximation (as all models are preserved but new ones possibly added), by (ii) order-independent such that for any set F of variable-free facts, we can simple write $forget(KB, F)$, and by (iii) reflects semantic similarity. Furthermore, item (iv) expresses a form of irrelevance: after forgetting facts, an adjusted sentence is entailed iff it is entailed by the original knowledge base; thus, the forgotten facts are not relevant for the entailment. For more on the use of forgetting for query answering, we refer to (Lin and Reiter 1994).

4.2 Forgetting Relations/Predicates

Forgetting a relation P, like *student*, from an FOL formula is more intricate. Here, we start with the semantic aspects of forgetting and first lift the semantic notion of model agreement, by defining that

$M_1 \sim_P M_2 \Leftrightarrow$ models M_1, M_2 fully agree except possibly
for the valuation of P.

With this, we are able to define the semantic effects of relation forgetting:

Definition 5 (Semantic Relation Forgetting; Lin and Reiter, 1994) Given an FOL *KB* and a relation symbol *P*,

$$Mod(forget(KB,P)) = \{M_1 \mid M_1 \sim_P M_2 \text{ for some } M_2 \in Mod(KB)\}.$$

The following proposition shows how this forgetting operation can be described on the syntactic level, where we move to second-order logic (SOL) in order to quantify over relation symbols:

Proposition 8 (Syntactic Characterization; Lin and Reiter, 1994) *Given an FOL $KB = \{\phi\}$ and a relation symbol R, we let $forget(KB,P) = \{\exists R.\phi(P/R)\}$, i.e., replace P in ϕ by R where R is a fresh relation symbol of the same arity as P, and put an existential quantifier.*

We illustrate relation forgetting by some examples.

Example 13 (Ex. 12, cont'd) We want to forget $P = student$ from the knowledge base

$$KB = \{student(John) \vee student(Joe) \vee teacher(John)\},$$

and obtain by Proposition 8

$$\begin{aligned} forget(KB, student) &= \{\exists R.R(John) \vee R(Joe) \vee teacher(John)\} \\ &= \{\top\}, \end{aligned}$$

as some model exists in which $R(John)$ is true; thus we are left semantically with a tabula rasa.

For the slightly modified knowledge base

$$KB' = \{(student(John) \vee (student(Joe)) \wedge teacher(John))\},$$

forgetting *student* results in

$$\begin{aligned} forget(KB', student) &= \{\exists R.(R(John) \vee (R(Joe)) \wedge teacher(John))\} \\ &= \{teacher(John)\} \end{aligned}$$

Relation forgetting extends to sequences $\sigma = P_1, P_2, \ldots, P_k$ of relation symbols in a straightforward way, and has similar properties as forgetting sequences of facts.

That the formula of $forget(KB,P)$ is no longer an FOL sentence, but from SOL has its challenges for the usage (in particular, for reasoning, which in SOL is highly undecidable in general). Hence a natural question is whether an equivalent FOL formula ψ exists; this is known as the *'existence' problems of forgetting*. In case of an affirmative answer, the question is how to compute such an FOL formula ψ for $forget(KB,P)$. While the existence problem is unsurprisingly not decidable in general, even in decidable cases and when an FOL formula ψ can be constructed, possibly under restricted syntax, the computation may be resource intense; e.g., scenarios for well-known description logics that require double exponential time for deciding existence and even more time and space for computation are known (cf. Section 4.4).

For a more extensive discussion of forgetting relations from an FOL knowledge base, we refer to (Delgrande 2017). We note here, however, that (Zhang and Zhou

2010) defined a notion of *weak forgetting* based on postulates from their earlier work (Zhang and Zhou 2009). It amounts to using in Definition 5, which they called *strong forgetting*, the more liberal similarity relation $M_1 \equiv_P M_2$ capturing that every formula not involving P has on M_1 and M_2 the same truth value. It allows to represent $forget(KB, P)$ by the set $Cn(KB) \cap L_{\Sigma \setminus \{P\}}$ of all consequences of KB in which P does not occur, respectively by any (possibly finite) subset thereof; for strong forgetting, such a representation does not always exist.

4.3 Forgetting Individuals

Deliberately forgetting a person is a problem that may arise particularly in cases where this person did some harm to someone, and remembering him or her is painful. For example, in ancient Egypt, the name of the heretic pharaoh Echnaton was erased from records years after his death; in German philosophy, a famous case is Martin Lampe, who had been a servant of Immanuel Kant for many years and was dismissed by Kant for personal misconducts. Kant expressed his struggle with effectively erasing Lampe from his memory by writing “Der Name Lampe muß nun völlig vergessen werden!” (“The name Lampe must be forgotten completely now!”) in one of his little notebooks (Wikipedia 2018).

In logical knowledge bases, specific individuals in a domain of discourse are usually represented by constant symbols, which serve as “names” for individuals. Delgrande (2014) proposed a technique to forget a constant symbol from the signature as follow.

Definition 6 (Forgetting Individuals; Delgrande, 2014) For a knowledge base $KB = \{\phi\}$ and constant symbol c, let

$$forget(KB, c) = (KB \setminus KB_c) \cup \{\exists x. \bigwedge_{\phi \in KB_c} \phi[c/x]\},$$

where KB_c is the set of all formulas in KB in which c occurs, x is a fresh variable and $\phi[c/x]$ denotes replacement of c by x in ϕ.

That is, forgetting is achieved by anynomization: it becomes opaque which individual has certain properties.

Example 14 Let $KB = \{student(John) \vee student(Joe) \vee teacher(John)\}$; then

$$forget(KB, John) = \{\exists x.student(x) \vee student(Joe) \vee teacher(x)\}$$

Delgrande showed that the above definition is effective in the sense that all consequences of the original knowledge base that do not mention c are retained, and that no new consequences emerge; in technical terms,

$$Cn_{\Sigma'}(forget(KB, c)) = \{\phi \in Cn_{\Sigma}(KB) \mid c \text{ does not occur in } \phi\},$$

where Cn’s subscript shows the signature used and $\Sigma' = \Sigma \setminus \{c\}$.

As an alternative to Definition 6, one might think of the perhaps more natural replacement of c with a fresh constant symbol, which is easier to implement; e.g. that in Example 14 *John* is replaced by $'xyz'$. This may look sufficient to blur intuitive meaning that is typically associated with symbols in applications. Technically, the new knowledge base would be isomorphic to the old one, and the same erasure of sentences from the latter would happen. Experts easily recognize that this variant is just a *skolemized form* of Definition 6. However, fresh constants leave traces in the knowledge base, such that it may be more transparent what forgetting operations about individuals happened in the past.

4.4 Forgetting in Description Logics

So far, we have considered forgetting in the context of FOL in general. For specific applications and reasoning tasks, a number of fragments of FOL have been studied and dedicated formalisms were developed based on them. Perhaps the most widely known such formalism are *description logics (DLs)* (Baader et al. 2003), which is a family of prominent languages for writing ontologies with a formal semantics; they are increasingly used because some of the standard recommendations of the World Wide Web consortium, which governs the development of the Web, for ontology languages are based on DLs. Technically, DLs amount to fragments of classical FOL with a special syntax, tailored for convenient representation of structured knowledge. The signature of a DL contains *concepts* (classes), *roles* (binary relations), and *individuals* (constants), and DL knowledge bases $KB = \mathscr{T} \cup \mathscr{A}$ consist of *taxonomic axioms* $\mathscr{T}$ and factual *assertions* $\mathscr{A}$. The axioms in $\mathscr{T}$ are of the form $\alpha \sqsubseteq \beta$, where α and β are concept expressions (which intuitively describe classes of individuals), while the assertions are facts about concept membership and role relationships between individuals. The most-widely known and "mother" of other DLs is $\mathscr{ALC}$, in which concept expressions can be formed using boolean connectives and role restrictions of the form $\forall R.C$ and $\exists R.C$, which allow one to describe the classes of individuals x such that every (some) individual y fulfilling $R(x,y)$ belongs to class C.

Example 15 The following DL knowledge base expresses knowledge on persons in general and on students in particular:

	DL KB	FOL
$\mathscr{T}$	$student \sqsubseteq person$	$\forall x.student(x) \rightarrow person(x)$
	$person \sqsubseteq \forall parent.adult$	$\forall x,y(person(x) \land parent(x,y) \rightarrow adult(y))$
$\mathscr{A}$	$student(John)$	$student(John)$

where the rightmost column shows translations of DL formulas to FOL formulas. Intuitively, the first axiom says that students are persons, and the second that for a person, each parent must be an adult. The assertion is self-explanatory.

The restricted DL syntax ensures (usually) *decidability* of reasoning, e.g. satisfiability and *query answering* ($KB \models person(John)$) (Baader et al. 2003).

Motivated by the need for efficient reasoning (which in expressive DLs may be difficult to achieve) and the desire to structure and decompose large knowledge bases into parts for modular use, forgetting from DL knowledge bases $KB = \mathscr{T} \cup \mathscr{A}$ has been widely studied (see e.g. Ghilardi et al. 2006a; Konev et al. 2008, 2009, 2013;

Koopmann 2015; Koopmann and Schmidt 2013; Ludwig and Konev 2014; Lutz and Wolter 2011; Lutz et al. 2012; Nikitina and Rudolph 2012; Wang et al. 2008; Zhao and Schmidt 2017, 2018b). The bulk of work has focused on different aspects: forgetting concepts, or roles from $\mathscr{T}$, and forgetting both concepts/roles from $\mathscr{T} \cup \mathscr{A}$.

Sometimes, forgetting is possible by a classic result due to Ackermann (1935), which customized for DL syntax is as follows.

Proposition 9 (Substitution Lemma; cf. Zhao and Schmidt, 2017) *Let KB be a DL knowledge base (in a suitable description logic, e.g. the expressive $\mathscr{ALCHOQ}$) such that*

(i) $KB \supseteq \{\alpha_1 \sqsubseteq P, \ldots \alpha_n \sqsubseteq P\}$ where P occurs in no α_i, and
(ii) P occurs in $KB \setminus \{\alpha_1 \sqsubseteq P, \ldots \alpha_n \sqsubseteq P\}$ only negative,

then $forget(KB,P) = KB[P/\alpha_1 \sqcup \cdots \sqcup \alpha_n]$, i.e., KB with substituting $\alpha_1 \sqcup \cdots \sqcup \alpha_n$ for P, where $\sqcup$ represents disjunction in DL.

Example 16 The knowledge base $KB = \{\, student \sqsubseteq person,\, person \sqsubseteq \forall parent.adult,\, student(John)\,\}$ from Example 15 satisfies the prerequisites of Proposition 9 for $P = person$; therefore

$$forget(KB,person) = \{\, student \sqsubseteq \forall parent.adult,\, student(John)\,\}.$$

However, it is not obvious how to apply this for nested concepts/roles (e.g., *parent*, *adult*) in *KB* from Example 15. Rewriting the knowledge base may help; in the knowledge base of the example, we may rewrite the axiom $person \sqsubseteq \forall parent.adult$ to the equivalent axiom $\exists parent^{-}.person \sqsubseteq adult$, which intuitively says that anyone who is a parent of a person must be an adult. However, such an equivalence preserving rewriting may not be feasible in general, or lead outside a particular DL (in this case, outside $\mathscr{ALC}$); and even if the technique would work for the FOL translation of a *KB*, the question would be how to translate the result back. Generalizations of such techniques can be found in (Zhao and Schmidt 2017).

In general, forgetting in DLs is difficult because a syntactic result for $forget(KB,P)$ that captures the semantic forgetting (based on model similarity) might not always exist in a specific DL (in fact, this applies to most of the classical DLs); this problem may be even undecidable, as e.g. for the DLs $\mathscr{ALC}$ (Konev et al. 2013) and $\mathscr{EL}$ (Botoeva et al. 2016). If $forget(KB,P)$ exists and is expressible in the DL at hand, it yields a *uniform interpolant*[6] for all consequences from *KB* on the remainder signature. Deciding the existence of such an interpolant is, however, expensive in general; e.g., for taxonomic knowledge bases in $\mathscr{ALC}$ and $\mathscr{EL}$, respectively, it requires single and double exponential time, respectively, and a smallest uniform interpolant occupies triple exponential space in both cases (Lutz and Wolter 2011; Nikitina and Rudolph 2012); for several expressive DLs, decidability is unexplored (cf. Koopmann 2015). There are different approaches to overcome these problems, among them to extend the DL language (e.g., from $\mathscr{ALC}$ to $\mathscr{ALC}_V$, Koopmann and Schmidt

[6] The notion of uniform interpolant will be described in Section 7, to which we also defer further discussion of the subject.

2013) or to introduce auxiliary vocabulary. For example, forgetting from the knowledge base $KB = \{Carnivor \sqsubseteq \exists eats.Meat,\ Vegetarian \sqsubseteq \forall eats.\neg Meat\}$ the role *eats* results in the knowledge base $\{Carnivor \sqsubseteq \exists \mathsf{u}.Meat,\ Carnivor \sqcap Vegetarian \sqsubseteq \bot\}$, where u is the universal role that holds between any individuals; the latter is stronger than the uniform interpolant $\{Carnivor \sqcap Vegetarian \sqsubseteq \bot\}$ even if we use instead of $\mathscr{ALC}$ the more expressive description logic $\mathscr{ALCOIH}$, say, for our knowledge base *KB*. Extending the language with the universal role u removes this problem. Similarly, forgetting *Meat* from the knowledge base $KB' = \{Carnivor \sqsubseteq \exists eats.Meat$, $Vegetarian \sqsubseteq \exists eats.\neg Meat\}$ results in $\{Carnivor \sqsubseteq \exists eats.\top$, $Vegetarian \sqsubseteq \exists eats.\top$, $Carnivor \sqcap Vegetarian \sqsubseteq\ \geq 2\, eats.\top\}$, and thus requires number restrictions (cf. Zhao and Schmidt 2018b). Furthermore, for practical applications, only approximations of $forget(KB, P)$ are computed (cf. Lutz et al. 2012; Koopmann 2015); we refer to (Zhao and Schmidt 2017, 2018b), (Koopmann 2015; Botoeva et al. 2016) for further discussion and background. Finally, several tools for forgetting from DL knowledge bases have been developed, among them one based on SCAN (Ohlbach et al. 2018), using skolemization and second-order quantifier elimination (see Alassaf and Schmidt 2017), Lethe (Koopmann 2015) and the one by (Ludwig and Konev 2014), which are based on resolution, and FAME (Zhao and Schmidt 2018a), which builds on the Ackermann-Lemma approach.

As of today, forgetting in DLs and the study of related problems (see Section 8) is a rich and active field of research; a detailed coverage of the subject, however, requires a dedicated survey and is beyond the scope of this article.

5 Forgetting in Answer Set Programming

We now leave KRR formalisms based on classical logic and turn to nonmonotonic formalisms. For such, work on forgetting (as of type 1) is rather scarce, and most studies have been devoted to rule-based formalisms, and in particular to nonmonotonic logic programs under different semantics; some seminal works are (Zhang et al. 2005; Wang et al. 2005; Alferes et al. 2013; Knorr and Alferes 2014; Antoniou et al. 2012). We concentrate in this article on programs under the answer set semantics. First, we give an overview on challenges that forgetting operations in this area have to deal with, and then point out an interesting connection between forgetting and reasoning from minimal models.

5.1 Challenges of Forgetting in ASP

For ASP programs, a number of different approaches to forgetting exist, and we refer to the recent paper by Leite (2017) for an excellent and comprehensive survey. Among the many proposals we find both syntactic operators like weak forgetting or strong forgetting (Zhang et al. 2005) and semantic approaches like semantic weak/strong forgetting Eiter and Wang (2008) or forgetting based on the monotonic core of answer set semantics (Wang et al. 2012).

The nonmonotonic nature and the fact that syntax matters in formulating rules poses a big challenge to define forgetting for ASP. Answer sets have the particular property of *supportedness*, which means that for each atom p that is true in an answer set M, there must be some rule which is applicable and allows to derive p. In this way, M may be constructed starting from facts by repeated rule application. A simple syntactic operation that eliminates p from the rules may disrupt this derivability.

Example 17 To strongly forget $student(sam)$ from the program listed in (1) on p. 6, a straightforward approach would be to omit all rules mentioning it, and restrict all other rules so as to not talking about $student(sam)$:

$$\begin{aligned} male(x) \vee female(x) &\leftarrow student(x), x \neq sam. \\ single(x) &\leftarrow student(x), not\ married(x), x \neq sam. \end{aligned}$$

Note that $single(sam)$ is also lost, which means that no semantic similarity of answer sets of the form $M \sim_p M'$ is warranted. Weak forgetting would omit all rules mentioning $single(sam)$ resp. prevent mentioning it outside the scope of '*not*', and would remove the occurrences under '*not*' resp. regard them as true; in this example, weak and strong forgetting coincide.

An idea to overcome this is by defining the result of forgetting p from KB straight in terms of semantic similarity $M \sim_p M'$. However, due to the minimality of answer sets, it may be impossible to obtain any program for $forget(KB, p)$ which captures semantic similarity: e.g., simply forget $student(sam)$ from $KB = \{student(sam)\}$; then by minimality, the only answer set of $forget(KB, student(sam))$ is $M = \emptyset$, which means that $student(sam)$ is switched to false (note that we are dealing here with 2-valued "classical" models). A way out is to either switch to a richer syntax that allows for comparable models (e.g., nested logic programs), or to restrain the similar models such that a program exists, as in (Eiter and Wang 2008).

Even then, one is left with another problem: the particular form of rules to express $forget(KB, p)$ matters. In general, ASP programs KB and KB' may have the same answer sets, but after adding further rules, this may no longer hold; i.e., the *substitution property*, that we can replace KB by KB' in any context of other rules, does not hold. For example, $KB = \{pub \leftarrow thirsty, not\ sunday.\}$ and $KB' = \{pub \leftarrow thirsty, weekend.\}$ have the same answer set $M = \emptyset$; but in context of an additional fact $thirsty$, the answer sets are different, i.e., $\{pub\}$ for KB, and $\emptyset$ for KB'. Note that there are also examples where KB and KB' have the same classical models (which is here not the case).

Programs KB and KB' enjoying the substitution property are called *strongly equivalent*, which amounts to equivalence in the nonclassical logic HT (Lifschitz et al. 2001). Its model-theory and consequence relation can thus serve as a basis for defining a forgetting operation that is robust with respect to strong equivalence; this has been attempted e.g. by Wong (2009); Wang et al. (2012, 2013); Delgrande and Wang (2015). Specifically, Wong (2009) presented a complete calculus for HT-entailment of rules from disjunctive programs, which he used for defining forgetting operators by program transformation; Wang et al. (2012, 2013) presented more elaborate notions of semantic forgetting under HT entailment that overcome some issues with operators proposed by Wong. Finally, Delgrande and Wang (2015) presented a syntactic notion of forgetting under HT-entailment, which has compared to (Wang et al. 2012,

2013) the benign property that the result of forgetting is easy to compute, and in fact by a program tranformation that adapts resolution-based forgetting in Definition 2 to answer set programs. Still, however, there is no canonical way of forgetting atoms from a logic program; this is somewhat similar to belief change, where also a plethora of operators have been proposed in the literature. In the vein of the AGM theory of belief change, properties respectively postulates have been proposed in order to guide the development and the assessment of approaches for forgetting. Wong (2009) was the first to characterize an ASP forgetting operator in terms of postulates, while later Wang et al. (2012) characterized their semantic forgetting in terms of postulates that Zhang and Zhou (2009) had proposed in the context of modal logic (see Section 6). Numerous further such postulates exist for ASP forgetting (cf. Leite 2017; Gonçalves et al. 2016a), like the following ($Mod(\cdots)$ stands here for the set of answer sets):

- *Strengthened Consequence:*
  $Mod(forget(KB,p)) \subseteq Mod_{\Sigma\setminus\{p\}}(KB)$
- *Weak Equivalence:*
  $Mod(KB) = Mod(KB')$ implies $Mod(forget(KB,p)) = Mod(forget(KB',p))$
- *Strong Persistence:*
  For any KB, atom p and KB' not involving p,

$$Mod_{\Sigma\setminus\{p\}}(KB \cup KB') = Mod(forget(KB,p) \cup KB').$$

Notably, strong persistence is related to strong equivalence of programs, in that the same context KB' is considered before and after the forgetting operation. It has been introduced in (Knorr and Alferes 2014) as a generalization of *Consequence Persistence*, which results as the special case $KB' = \emptyset$. While desirable, it turns out that strong persistence is not always possible to achieve, as Gonçalves et al. (2016b) showed on the following example. Given the program

$$KB = \left\{ \begin{array}{ll} p \leftarrow not\ q. & a \leftarrow p. \\ q \leftarrow not\ p. & b \leftarrow q. \end{array} \right\},$$

one cannot forget p,q from KB and expect all semantic relations between a and b to prevail. If we attempt e.g. to let $forget(KB,\{p,q\}) = \{a \leftarrow not\ b., b \leftarrow not\ a.\}$, then we have for $KB' = \{a \leftarrow b., b \leftarrow a.\}$ that $KB \cup KB'$ has some answer set that contains a and b, while $forget(KB,\{p,q\}) \cup KB'$ has no answer set. By considering different additions KB' and the presence resp. absence of certain answer sets for $KB \cup KB'$, one can argue that no program $forget(KB,\{p,q\})$ is possible that would satisfy the condition of *Strong Persistence*.

Current research questions are when is strong persistence achievable, and which weaker notions of persistence would be helpful. Moreover, algorithms and implementations for forgetting in ASP programs with the desired properties are needed.

5.2 Minimal Model Reasoning and Forgetting

There is broad agreement that for programs without default negation, the semantics should be given in terms of its minimal models, that is, the models M of KB (viewed as a classical knowledge base) such that no model $M' \subset M$ of *KB* exists; hence, inference of a formula ϕ is tantamount to inference of ϕ from the circumscription of *KB* (McCarthy 1986); In fact, every answer set of an arbitrary program *KB* (with negation allowed) is a minimal model of *KB*. Thus, minimal models are at the heart of ASP, and negation serves as a mechanism to select more plausible ones.

An interesting connection between minimal models and forgetting was found by Lang et al. (2003), which links the latter to reasoning from minimal models. Let us denote by $MMod(KB) = \{M \in Mod(KB) \mid \nexists M' \in Mod(KB) : M' \subset M\}$ the set of all minimal models of a knowledge base *KB*, and by $KB \models_{min} \phi$ that a formula ϕ is true in all minimal model of *KB*, i.e., $M \models \phi$ for every $M \in MMod(KB)$.

Proposition 10 (Minimal Model Reasoning by Forgetting; Lang et al., 2003) *For every propositional knowledge base KB and formula ϕ,*

$$KB \models_{min} \phi \text{ if and only if } KB \models forgetLit(KB \cup \{\phi\}, L^-)$$

where L^- is the set of all negative literals over the signature.

This result is intuitively explained by the fact that $KB \models_{min} \phi$ holds iff for each positive clause $\gamma = p_1 \vee \cdots \vee p_k$, the knowledge base $KB \cup \{\neg p_1, \ldots, \neg p_k\}$ has a model M only if $KB \cup \{\phi\} \cup \{\neg p_1, \ldots, \neg p_k\}$ has some model M'. Now if γ is maximal (i.e., is not extendable), M must be a minimal model of *KB* and coincide with M'. As a model of $KB \cup \{\phi\}$, M' is also a model of $forgetLit(KB \cup \{\phi\}, L^-)$, as is every model $M'' \supseteq M'$, and in particular, model M'' of *KB*; this results in the stated condition.

Example 18 Consider the simple knowledge base $KB = \{male(sam) \vee female(sam).\}$ and the formula $\phi = \neg male(sam) \vee \neg female(sam)$. Then we have $MMod(KB) = \{\{male(sam)\}, \{female(sam)\}\}$ and $KB \models_{min} \phi$. Furthermore, where $m = male(sam)$, $f = female(sam)$,

$$\begin{aligned}
forgetLit(KB \cup \{\phi\}, \{\neg m, \neg f\}) &= forgetLit(forgetLit(KB \cup \{\phi\}, \neg m), \neg f) \\
&= forget(\{(\bot \vee f) \wedge (\top \vee \neg f) \vee m \wedge (m \vee f) \wedge (\neg m \vee \neg f)\}, \neg f) \\
&= forget(\{f \vee m\}, \neg f) \\
&= \{\bot \vee m \vee f \wedge (f \vee m)\} \\
&= f \vee m,
\end{aligned}$$

where simplifications are used. We have $KB \models f \vee m$, as stated by Proposition 10.

As mentioned above, the entailment $KB \models_{min} \phi$ amounts to inference of ϕ from the circumscription of *KB* (McCarthy 1986); more precisely, it is the inference $CIRC(KB; P, Q, Z) \models \phi$ in the circumscription framework where P are all variables and Q, Z are empty ("global circumscription"). Lang et al. (2003) showed also how to express any inference $CIRC(KB; P; Q; Z) \models \phi$ in terms of an entailment $KB \models forgetLit(KB', L')$, where KB' results from *KB* by forgetting some literals. Thus, forgetting can serve as a computational tool that allows one to reduce circumscription easily to classical inference without the need for expensive operations such as satisfiability or minimality tests.

On the other hand, deciding $KB \models_{min} \phi$ is harder than propositional inference, and is complete for the class $\Pi_2^p = \text{co-}NP^{NP}$ of the polynomial hierarchy (Eiter and Gottlob 1993). By complexity-theoretic considerations, we may thus not expect to construct a formula for $\mathit{forgetLit}(KB \cup \{\phi\}, L^-)$ in polynomial time; moreover, a small (polynomial) size formula for $\mathit{forgetLit}(KB \cup \{\phi\}, L^-)$ is unlikely to exist; this can be derived from results about the succinctness of circumscription (Cadoli et al. 2000).

6 Forgetting in Modal Logic

We end our tour through forgetting in different KRR formalisms with a look at modal logics. Among its many incarnations, in particular the modal logic S5 is prominent as a basic logic of knowledge (cf. Fagin et al. 1995). Semantically, the meaning of formulas ϕ in S5 can be described by special Kripke models of the form $M = (W, R, \tau, w)$, where W is a set of possible worlds, R is a binary transition relation $R(w, w')$ between possible worlds w and w', τ assigns each possible world $w \in W$ a model, i.e., a value assignment $\tau(w)$ over an underlying set of propositional variables, and w is the actual world. In S5, R is universal, i.e., from every world w any world w' can be reached (including w itself), and each possible world w can be identified with $\tau(w)$; thus one writes simply $M = (W, w)$. The semantics of S5 is captured by modal logic axiom schemes, which are used by calculi for this logic.

Baral and Zhang (2005) defined forgetting from knowledge bases in modal logic S5 as a special form of update. In their approach, the idea is to exploit the power of knowledge and to interpret forgetting a formula ϕ from a knowledge base KB as making the agent *ignorant* about ϕ. To this end, $\mathit{forget}(KB, \phi)$ is realized by updating KB with the formula $\neg K\phi \wedge \neg K\neg\phi$: after this update, the agent would neither know ϕ nor $\neg\phi$. Furthermore, by basic logical properties s/he will be aware of that, as $K(\neg K\phi \wedge \neg K\neg\phi)$ holds.

The update may not always be realized by a simple syntactic operation, such as by variable elimination in the classical propositional setting.

Example 19 Suppose the agent holds the knowledge base

$$KB = \{\mathit{inflagranti} \rightarrow \mathit{guilty}, K\mathit{guilty} \vee K\neg\mathit{guilty} \rightarrow \mathit{justice}\}$$

and she wants to forget *guilty* from it. By variable elimination, e.g. constructing $\mathit{forget}(KB, \mathit{guilty}) = \{\phi^+_{\mathit{guilty}} \vee \phi^-_{\mathit{guilty}}\}$ where ϕ^+_{guilty} (resp. ϕ^-_{guilty}) is the conjunction of the formulas in KB where *guilty* is replaced by $\top$ (resp., $\bot$), we obtain a knowledge base that is equivalent to the formula $\mathit{inflagranti} \rightarrow \mathit{justice}$. However, the latter is not a consequence of KB; this is in contrast to the intuition that forgetting is a semantic weakening and thus should not lead to new consequences. Even worse, if we forget *guilty* from the extended knowledge base $KB' = KB \cup \{\neg\mathit{justice}\}$, which is satisfiable, then $\mathit{forget}(KB', \mathit{guilty})$ is inconsistent!

Baral and Zhang thus defined a more sophisticated approach which performs the update with $\neg K\phi \wedge \neg K\neg\phi$ in a way such that possible worlds are added to a model, if needed, in order to create ignorance. If we concentrate on atomic formulas ϕ, i.e., variables p, then a Kripke model $M' = (W', w')$ is a model of the update of KB with $\neg Kp \wedge \neg K\neg p$, exactly if some model $M = (W, w)$ of KB exists such that

(i) $w = w'$,
(ii) $W = W'$ if $M \not\models Kp$ and $M \not\models K\neg p$, and
(iii) $W' = W \cup \{w^*\}$ where (a) $p \notin w^*$ if $M \models Kp$ and (b) $p \in w^*$ if $M \models K\neg p$

as follows from (Baral and Zhang 2005, Proposition. 7). For the belief revision aficionados, this update resembles an AGM contraction where countermodels are added in order to construct remainder sets for partial meet contraction (Alchourrón et al. 1985).

Example 20 (cont'd) In Example 19, forgetting *guilty* from KB' by update with the formula $\neg K guilty \wedge \neg K\neg guilty$ leaves KB unchanged, as in every model M of KB', in some possible world *guilty* is true and in some *guilty* is false; on the other hand, the same update would for KB yield a model $M' = (\{w_1, w_2\}, w_1)$ from $M = (\{w_1\}, w_1)$ where $w_1 = \{$*justice*, *inflagranti*, *guilty*$\}$, as we can add $w_2 = \{$*justice*, *inflagranti*$\}$ for item (iii.a) which is not a model of KB.

In the paper (van Ditmarsch et al. 2009), the authors considered forgetting in a broader epistemic setting and distinguished different forms of forgetting a propositional variable p:

- *Becoming ignorant:* defined as above, $K(\neg Kp \wedge \neg K\neg p)$ holds (introspective forgetting)
- *Becoming unaware:* remove p from the signature; to revert forgetting, i.e., remembrance, is no longer possible.
- *Forgetting values:* considering multi-valued properties, e.g. a 4-digit pincode, which are representable by multiple boolean variables, e.g., represent pincode by $c^i_j =$ "the i-th digit of the code is j"; for this, one cane use a similar procedure as for single boolean variables.
- *Multi-agent versions of forgetting:* this is a group version for becoming ignorant and should involve prior common knowledge followed by some sort of *collective memory loss*.

They showed that besides variable elimination (cf. Example 19) two further construction methods for forgetting in classic propositional logic discussed by Lang et al. (2003) fail in modal logics, namely, to compute the strongest consequence not mentioning p, and the strongest formula on similar models, i.e., on $\{M \mid M \sim_p M', M' \in Mod(KB)\}$, respectively. Instead, the authors modeled forgetting as an *event* or *action* $Fg(p)$ in a dynamic epistemic logic. Intuitively, the formula $[Fg(p)]\phi$ expresses that ϕ holds after forgetting p. The agent moves to parallel worlds where s/he can not distinguish p from $\neg p$. They considered *introspective forgetting* (i.e., $K(\neg K\phi \wedge \neg K\neg\phi)$), and explored the relation to unawareness forgetting. Notably, also a sound and complete axiomatization of the logic has been provided by van Ditmarsch et al. (2009).

Zhang and Zhou (2009) observed that the approaches by Baral and Zhang (2005) and van Ditmarsch et al. (2009) have weaknesses and do not yield intuitive results in certain cases. In their work, they made two major contributions in order to adress these weaknesses: (1) to consider a more fine-grained notion of similarity between models than van Ditmarsch et al. (2009) did, and (2) to present a set of postulates that any reasonable forgetting operator should satisfy in the spirit of the AGM theory for belief change (Alchourrón et al. 1985).

As for 1), the similarity relation $M_1 \sim_V M_2$ for variables V is replaced by V-*bisimilarity* $\leftrightarrow_V$: if $M_1 = (W_1, w_1)$ and $M_2 = (W_2, w_2)$, then w_1 and w_2 coincide modulo V, and for every $w_1^* \in W_1$ there is some $w_2^* \in W_2$ that coincides with it modulo V and vice versa. The ordinary notion of bisimilarity, which results for $V = \emptyset$, is a well-known condition for models that are inseparable by formulas in modal logic. This leads then to an elegant semantic definition of knowledge forgetting.

Definition 7 (Modal Logic Forgetting; Zhang and Zhou, 2009) For any propositional modal-logic knowledge base *KB* and set *V* of variables,

$$Mod(forget(KB), V) = \{M \mid \exists M' \in Mod(KB), M \leftrightarrow_V M'\}.$$

It appeared that this definition comes with nice properties: first, for finite *KB* the result of forgetting always exists, i.e., there is some modal formula ϕ whose S5-models are $Mod(forget(KB), V)$; moreover, such ϕ (of possibly double exponential size) can be constructed using an algorithm in (Ghilardi et al. 2006b). Second, if *KB* consists only of objective (modal-free) formulas, then ϕ is objective as well. That is, this knowledge forgetting properly generalizes forgetting in classical propositional logic. Third, that forgetting is syntax independent, i.e., whenever *KB* and and KB' are logically equivalent (in S5), then also $forget(KB)$ and $forget(KB')$ are logically equivalent.

As for 2), Zhang and Zhou presented the following four postulates for forgetting V from *KB*, where $IR(KB, V)$ means that some knowledge base KB' logically equivalent to *KB* exists in which no variable from *V* occurs:

(W) $KB \models forget(KB, V)$ (weakening)
(PP) if $KB \models \phi$ and $IR(\phi, V)$, then $forget(KB, V) \models \phi$ (positive persistence)
(PP) if $KB \not\models \phi$ and $IR(\phi, V)$, then $forget(KB, V) \not\models \phi$ (negative persistence)
(IR) $IR(forget(KB, V), V)$ (irrelevance).

Informally, (W) expresses that forgetting should weaken the knowledge base, and (PP) and (NP) that consequences that do not mention any of the forgotten variables remain unaffected; finally, (IR) states that forgetting has a syntactical representation on the reduced signature.

The main result of Zhang and Zhou is then that these postulates capture knowledge forgetting.

Proposition 11 (Zhang and Zhou, 2009) *For any modal logic knowledge bases KB, KB' and set V of variables, KB' expresses $Mod(forget(KB, V))$ iff $forget(KB, V) := KB'$ fulfills (W), (PP), (NP), and (IR) iff KB' is logically equivalent to $\{\phi \mid KB \models \phi, IR(\phi, V)\}$.*

Furthermore, it has been shown that knowledge forgetting as in Definition 7 enjoys several further properties of variable forgetting from classical logic bases. Like the one in (Baral and Zhang 2005), it can be used to define updates of modal-logic knowledge bases, and it has been applied to model games with bounded memory.

Furthermore, knowledge forgetting as in Definition 7, which amounts to existential bisimulation quantification, has been recently extended to the multi-modal setting (Fang et al. 2016). The authors have also shown how for a variable p, $\mathit{forget}(\mathit{KB}, p)$ can be constructed by simple literal elimination from a canonical form of *KB*. In this way, they managed to show that the result of forgetting exists for a range of multi-modal logics, viz. K_n, D_n, T_n, $K45_n$, $KD45_n$, and $S5_n$, where n is the number of agents.

7 Related Notions

In this section, we consider some concepts and notions that are closely related to forgetting; some of them have already appeared in more or less explicit form in the previous sections.

Relevance. As we have discussed in the Introduction, forgetting is a means to reduce information, based on different motivations. In most cases, the operation aims to drop information that is not relevant for a specific application. It is thus suggestive to consider the notion of irrelevance a bit closer in the context of forgetting.

From a logical point of view, a set V of variables is *relevant* for a propositional knowledge base *KB*, if in every knowledge base KB' that is logically equivalent to *KB*, some variable from V must occur in some formula of KB'; this is equivalent to the definition that *KB* has some prime implicate that mentions some variable from V (Lakemeyer 1997), and coincides with other notions like influenceability (Boutilier 1994, cf. Lang et al., 2003). Now recall that the result of $\mathit{forget}(\mathit{KB}, V)$ can be described by a knowledge base in which no variable from V occurs. Semantically, forgetting V from *KB* strives for indifference w.r.t. V, which is achieved by adding to the models of *KB* all $\sim_V$-similar models. No such addition is necessary if every such model is a model of *KB* – and in this case, *KB* and $\mathit{forget}(\mathit{KB}, V)$ coincide. In fact, the following property holds.

Proposition 12 (Irrelevance as Forgetting; Lang et al., 2003) *Given a propositional knowledge base KB and a set V of variables, V is not relevant for KB if and only if KB is logically equivalent to* $\mathit{forget}(\mathit{KB}, V)$.

This notion of irrelevance may be adjusted to particular usages of knowledge bases. If e.g. a knowledge base serves to find answers to a query, then the irrelevance of variables can be contextualized: a set of variables V is *irrelevant* for a knowledge base *KB with respect to a query q*, if $\mathit{KB} \models q$ holds if and only if $\mathit{forget}(\mathit{KB}, p) \models q$ holds. This naturally extends then to a set or class Q of queries.

Example 21 Recall that for the knowledge base

$$\mathit{KB} = \{\mathit{red\_wine} \ \vee \ \mathit{white\_wine},\ \mathit{fish} \to \mathit{white\_wine},\ \mathit{beef} \to \mathit{red\_wine}\}$$

from Example 3, forgetting the variable *fish* yields the knowledge base

$$\mathit{forget}(\mathit{KB}, \mathit{fish}) = \{\mathit{red\_wine} \vee \mathit{white\_wine}, \mathit{beef} \to \mathit{red\_wine}\}.$$

Thus, *fish* is irrelevant e.g. for the query whether to drink *white_wine*, but not whether fish should go with white wine, i.e. $\mathit{fish} \to \mathit{white\_wine}$, which was dropped from *KB*.

We remark that the relation in Proposition 12 does not carry over to forgetting facts from FOL knowledge bases; this is because a fact $P(\mathbf{t})$ may not occur in *KB* and thus is literally not relevant for *KB*, as $\mathit{student}(\mathit{John})$ for $\mathit{KB} = \{\forall x.\mathit{student}(x)\}$ in Example 12. Thus, relevance requires a more careful consideration in this setting.

Independence. The above notion of relevance is in fact the dual of independence: A propositional knowledge base *KB* is *independent* of a set V of variables, if there exists some knowledge base KB' that is logically equivalent to *KB* in which no variable from V occurs. In other words, *KB* is independent of V iff V is irrelevant for *KB*.

Semantically, this means that independence holds iff forgetting V from *KB* does not introduce any new models; that is, $\mathit{Mod}(\mathit{KB})$ is closed under similar models: whenever M is a model of *KB* and $M' \sim_V M$ holds, then M' is also a model of *KB*.

An analogous characterization of independence (resp. relevance) in terms of forgetting holds in the more general setting of forgetting literals (cf. Lang et al. 2003). Conversely, forgetting literals L can be also characterized in terms of independence, similarly as in (Lin 2001) for variables:

Proposition 13 (Lang et al., 2003: Forgetting = Strongest Necessary Consequence) *For a propositional knowledge base* $\mathit{KB} = \{\phi\}$ *and literals* L*,* $\mathit{forgetLit}(\mathit{KB},L)$ *is the logically strongest* KB' *that is Lit-independent from* L *such that* $\mathit{KB} \models \mathit{KB}'$*.*

Here, Lit-independent from L means that there is some logically equivalent knowledge base in which no literal from L occurs, and logically strongest means that for every KB'' with this property, $\mathit{KB}' \models \mathit{KB}''$ holds.

Example 22 Let us consider the knowledge base

$\mathit{KB} = \{\mathit{red\_wine} \vee \mathit{white\_wine},\ \mathit{white\_wine} \vee \mathit{caviar} \to \mathit{happy},\ \mathit{red\_wine} \to \mathit{happy}\}$.

By Proposition 13, we obtain $\mathit{forgetLit}(\mathit{KB},\{\mathit{caviar}\}) = \{\mathit{red\_wine} \vee \mathit{white\_wine}, \mathit{happy}\}$. Indeed, *KB* is $\{\mathit{caviar}\}$-independent, as it does not occur positively in *KB*. Furthermore, *KB* logically entails *happy* and also $\mathit{red\_wine} \vee \mathit{white\_wine}$, but neither *red_wine* nor *white_wine*; thus $\mathit{forget}(\mathit{KB},\{\mathit{caviar}\})$ is as described.

Based on this characterization, we obtain the following properties

Proposition 14 (Independence and Forgetting; Lang et al., 2003) *For a propositional knowledge base* $\mathit{KB} = \{\phi\}$*, an atom* p*, and literals* L*, the following statements hold:*

(i) Remove literals: *KB is Lit-independent from L iff KB is logically equivalent to* $\mathit{forgetLit}(\mathit{KB},L)$*.*

(ii) Query preservation: *If* ϕ *is Lit-independent from L, then* $\mathit{forgetLit}(\mathit{KB},L) \models \phi$ *iff* $\mathit{KB} \models \phi$*, i.e., formula* ϕ *follows after forgetting L from KB iff it follows so before.*

(iii) Forgetting decomposition: *If* $\mathit{KB} = \mathit{KB}_1 \cup \mathit{KB}_2$*, where* KB_1 *and* KB_2 *share no variables, then* $\mathit{forget}(\mathit{KB},p) = \mathit{forget}(\mathit{KB}_1,p) \cup \mathit{forget}(\mathit{KB}_2,p)$*.*

$$KB = \left\{\begin{array}{l} rain \rightarrow road\_wet \\ road\_wet \rightarrow danger \\ rain \end{array}\right\} \xrightarrow[\text{query: } \neg rain \vee danger]{forget(KB, road\_wet)} KB' = \left\{\begin{array}{l} rain \rightarrow danger \\ rain \\ \end{array}\right\} \begin{array}{l} \phi = KB \\ \psi = \neg rain \vee danger \\ \rho = KB' \text{ (interpolant)} \end{array}$$

Fig. 2: Reduced knowledge base for query entailment via interpolation

(iv) Independent subparts: *If $KB = KB_1 \cup KB_2$ and KB_1 is independent of p, then $forget(KB,p) = KB_1 \cup forget(KB_2,p)$.*

We note that in item (iii), the disjointness condition is important. E.g., for $KB_1 = \{p\}$ and $KB_2 = \{\neg p\}$, $forget(KB_1 \cup KB_2, p) = \{\bot\}$ is inconsistent while $forget(KB_1,p) = forget(KB_2,p) = \{\top\}$ and thus the union $forget(KB_1,p) \cup forget(KB_2,p)$ is consistent. Recall that for the disjunctive form $KB = KB_1 \vee KB_2$, the decomposition $forget(KB,p) = forget(KB_1,p) \vee forget(KB_2,p)$ holds by Proposition 6.(v) even if KB_1 and KB_2 share variables. Likewise in item (iv), no disjointness condition is needed, as to obtain $forget(KB_1 \cup KB_2)$, we can replace KB_1 by some KB_1' in which p does not occur, and variable elimination leaves KB_1' untouched.

Uniform Interpolation. A further approach to forgetting is possible via *interpolation*, which deals with sentences "in between" a sentence and consequence of the latter. The following theorem is a classic result on this issue.

Theorem 1 (Interpolation Lemma by Craig, 1957) , *For every sentences $\phi \not\equiv \top$ and $\psi \not\equiv \bot$ in FOL such that $\phi \models \psi$, there exists some sentence ρ on the joint signature of ϕ and ψ, such that $\phi \models \rho$ and $\rho \models \psi$.*

Interpolation can thus be exploited to eliminate part of the signature from a knowledge base *KB* such that a reduced knowledge base entails a query faithfully; Figure 2 illustrates this on a simple example: for the entailment of the query $q = \neg rain \vee danger$, we can obtain the reduced knowledge base KB' by forgetting *road_wet*.

For a set respectively *class Q* of queries, a *uniform interpolant* ρ may be chosen, which is an interpolant for every query ρ in Q. In particular, if Q consists of all sentences that do not mention a predicate symbol P that should be forgotten, a uniform interpolant ρ for Q is a candidate representation of $forget(KB,P)$; in case the latter exists (i.e., an FOL sentence exists that captures the class $\{M \mid \exists M' \in Mod(KB), M \sim_P M'\}$ of models), then uniform interpolants ρ exist as well and any strongest such ρ represents $forget(KB,P)$ faithfully. Unfortunately, the existence of uniform interpolants does not ensure that semantic $forget(KB,P)$ exists in general. Therefore uniform interpolants provide an approximation of $forget(KB,P)$ in general; for answering queries over the reduced signature, this approximation is in fact exact.

Finding uniform interpolants is thus an attractive approach for forgetting in the context of classical FOL which has been extensively exploited (e.g. in description logics, Konev et al. 2009, started a systematic investigation of languages to express uniform interpolants for different query classes Q). The notion is intimately related to *inseparability* and *conservative extensions* of logical theories; we refer to (Lutz and

Wolter 2011; Botoeva et al. 2016) for further information and a broad discussion of the subject.

For nonmonotonic KR formalisms, there is much less work on uniform interpolation. Gabbay et al. (2011) considered *Equilibrium Logic*, which is a logical view of ASP as a nonmonotonic version of the nonclassical logic HT (Pearce 2006), and they characterized projective ASP equivalence of (Eiter et al. 2005) as inseparability via interpolation. Moreover, they exploited results on forgetting facts in ASP (Eiter and Wang 2008) to show a uniform interpolation property for facts and conjunctions of literals. Wang et al. (2013) later generalized this to all formulas in ASP.

More recently, Delgrande and Wang (2015) defined a consequence-based notion of forgetting propositional variables V from an ASP program. In their approach, the result of $\mathit{forget}(KB, V)$ is a uniform interpolant for all consequences ϕ of KB that do not involve any variable from V; furthermore, they gave an algorithm for constructing such an interpolant. It remains to be seen, however, to what extent these results can be extended to non-propositional programs.

Finally, as for modal logics, the knowledge forgetting of Zhang and Zhou (2009) in Definition 7 amounts to existential bisimulation quantification. Any logic that is closed under the elimination of this quantification has the interpolation property, and for several modal logics, this was known (Fang et al. 2016); the authors have extended this to the range of multimodal logics K_n , D_n, T_n, $K45_n$, $KD45_n$, and $S5_n$.

8 Applications in KRR

In this section, we briefly report on some ways of using forgetting in order to solve various problems in the field of knowledge representation and reasoning. Our report should give an impression on the manifold of applications and thus provide evidence for the relevance of the subject. It is partly based on (Lang et al. 2003; Koopmann 2015; Leite 2017), but by no means covers the different usages of forgetting and the literature exhaustively.

Query Answering Forgetting and/or independence properties have been widely used to optimize query answering from knowledge bases, for different purposes. To remove redundancy, query answering can be limited to consider formulas independent from a set V of variables. These variables are forgotten from the formulas respectively, while consequences that are independent of V are preserved. The same method is used in the FOL context or dedicated fragments like description logics, where predicates (respectively, concepts and/or roles) are forgotten in order to reduce the knowledge base to a relevant part for query answering; this may be done offline based on the characteristics of the online queries, computing e.g. interpolants (see Section 7).

It is worth pointing out that besides for deduction, forgetting is also useful for abduction and explanation finding, as needed e.g. in diagnostic reasoning. To this end, explanations for observations on a system are formed using abnormality assumptions on the non-working of components in the system. Then all but abnormality is forgotten, and the broken (i.e., abnormal) components are distilled out as an explanation. For example, a formula $\phi = \mathit{switch\_on} \wedge \neg ab_{bulb} \rightarrow \mathit{light}$ may express that there is light if the switch is on and the bulb of a lamp is ok. If we observe the switch is on but the light is off, a broken bulb is a possible explanation; it results by $forget(\{\phi, switch\_on, \neg light\}, \{switch\_on, light\} = \{ab_{bulb}\}$.

Knowledge Base Engineering In fact, the decomposition technique may also be utilized at an abstract level of knowledge engineering in order to structure large or complex knowledge bases. The latter are decomposed into thematic sub-KBs (e.g., geographic aspects, events in a city, public transport), and parts that are irrelevant for a specific task are forgotten. In the analysis of ontologies, restricted views are computed that use only a limited set of symbols of current interest. Hidden relations between concepts and individuals are then made explicit (Konev et al. 2009, cf.), which aids in clarifying relations among predicates respectively propositions. Furthermore, applying changes to an ontology can effect undesired new entailments regarding already defined concepts. The new entailments in the common signature of two ontologies are referred to as their *logical difference* (Konev et al. 2008). This logical difference can easily be computed by checking for entailment of the axioms of the respective uniform interpolants that implement forgetting (cf. Ludwig and Konev 2014, and the discussion of interpolation in Section 7). For an application, only a part $\Sigma' \subseteq \Sigma$ of the KB signature may be relevant. Uniform interpolants can be used to extract small subsets of ontologies, as *modules*, for reusing them in specialized contexts; we refer to (Botoeva et al. 2016) and references therein for more details. When multiple users access an ontology, protecting confidential information is critical (Grau and Motik 2010). A possible solution to this problem is to let users share a uniform interpolant of the original ontology, in which confidential concepts and relations have been eliminated. Summarising these different applications, forgetting is an important method for knowledge engineering in order to provide views on knowledge bases, by respecting relevance of and interest in information.

Belief Revision and Knowledge Base Update A general form of knowledge base update, resp. belief change, can be performed by using forgetting operations. In order to update a (propositional) knowledge base KB with a formula ϕ, one can proceed in three steps as follows:

(1) first, relevant variables V for the update of KB by ϕ are determined;
(2) then all variables in V are forgotten from KB, resulting in $KB' = forget(KB, V)$;
(3) finally, ϕ is added to KB' to obtain the result $upd(KB, \phi) = KB' \cup \{\phi\}$.

The only problem which remains here is how to determine V. Weber (1986) proposed to collect in V all variables that occur in some *minimal disagreement set* between KB and ϕ, where a disagreement set is any set D of variables on which some model M of KB and some model M' of ϕ disagree, i.e., $D = (M \setminus M') \cup (M' \setminus M)$ is the symmetric set difference between M and M'; D is minimal, if no disagreement set $D' \subset D$ exists. We illustrate this in the following example.

Example 23 Consider $KB = \{rain, rain \to road\_wet\}$. We want to update KB with $\phi = \neg road\_wet$.The single minimal disagreement set according to (Weber 1986) here is $D = \{road\_wet\}$, which results for $M = \{rain,\ road\_wet\}$ and $M' = \{rain\}$. Forgetting $V = D$ leads to

$$\begin{aligned} KB' &= \{\phi^{+}_{road\_wet} \vee \phi^{-}_{road\_wet}\} \\ &= \{[rain \wedge (rain \rightarrow \bot)] \vee [rain \wedge (rain \rightarrow \top)]\} \\ &= \{rain\}, \end{aligned}$$

so that we obtain $upd(KB, \phi) = \{rain, \neg road\_wet\}$ in the end. Informally, what has happened here is that the formula $rain \rightarrow road\_wet$ has been given up in order to arrive at a satisfiable knowledge base.

One may ask how this three-step approach compares to the AGM framework (Alchourrón et al. 1985), and whether in particular the AGM axioms for belief revision would be satisfied. In fact, the technique to first remove conflicts, which is achieved by the steps (1) and (2), and then adding the update formula in step (3) is very much in the spirit of the Levi identity from AGM, which defines revision in terms of contraction and expansion (Gärdenfors and Rott 1994). In general, forgetting all relevant variables would be quite a drastic form of belief contraction according to the contraction principles of AGM theory, and thus one may expect that Weber's operator does not satisfy the AGM revision postulates; this is indeed the case because it is not complying with the minimal change paradigm in AGM theory; for more details, cf. (Katsuno and Mendelzon 1991b).

Notably, further belief update operators from the literature can be the characterized by the three-step method from above, such as the modified PMA (MPMA) approach (Doherty et al. 1998) and equivalent approaches (see Herzig and Rifi 1999). While the latter do not satisfy the AGM postulates either, they satisfy the postulates of Katsuno and Mendelzon (1991a), which aim to capture the essence of a world change as opposed to a belief change. The fact that forgetting variables V from a knowledge base KB boils down to forgetting V from each model of KB individually matches a similar KM-postulate for update suggesting that the three-step approach to belief update from above is in fact better suited for the KM framework. Indeed, a respective update operator by Zhang and Zhou (2009) for modal-logic knowledge bases, which is based on their knowledge forgetting for Step 2 and generalizes the MPMA approach, satisfies the KM postulates. The latter holds also for the belief update approach by Baral and Zhang (2005), which generalizes the PMA approach. In this context, it might be even more rewarding to compare forgetting operators to erasure operations which have also been considered in (Katsuno and Mendelzon 1991a) as counterparts to update operations.

Inconsistency Tolerance. Inconsistencies occur if too much information is available, also containing conflicts and contradictions. A suggestive way to remedy this problem is to reduce information by forgetting. Restoring consistency is deeply related to revision and update, e.g., Hansson (1999) calls this operation *consolidation*; in the databases and KRR communities, it is called a *repair* (cf. Bertossi 2011).

Several authors have developed methods to restore consistency based on forgetting. Lang and Marquis (2002) used forgetting to merge knowledge bases $KB_1, \ldots KB_n$

when their naive union may result in inconsistency. Xiao et al. (2017) represent *repair semantics* of description logics by forgetting in the following way. Given a DL knowledge base $KB = \mathscr{T} \cup \mathscr{A}$, they defined an *(AR-)repair* for KB to be any maximal $\mathscr{A}' \subseteq \mathscr{A}$ such that the sub-base $KB' = \mathscr{T} \cup \mathscr{A}'$ of KB is satisfiable. Then KB *repair-entails* a query q, denoted $KB \models_r q$, if each repair KB' for KB logically entails q. For certain DLs, e.g. DL-Lite, repair-entailment is expressible via forgetting: for (boolean conjunctive) queries q over signature Σ', we have

$$KB \models_r q \text{ if and only } KB' \models q \text{ for some repair } KB' = \mathscr{T}' \cup \mathscr{A}' \text{ for } KB,$$

where KB' is expressed in DL-Lite over the signature $\Sigma \setminus \Sigma'$, and $\mathscr{T}' = \mathit{forget}(\mathscr{T}, \Sigma \setminus \Sigma')$. Moreover, KB' is effectively computable (for variants of repairs, even in polynomial time under data complexity) (Xiao et al. 2017).

Reasoning about Knowledge. Forgetting can also be exploited for reasoning about knowledge in a multi-agent setting. Su et al. (2009) equipped agents with variable forgetting to reason about their own knowledge or that of other agents, given a background knowledge KB. The key notions in this approach are *observable variables* (for each of the agents) and *weakest sufficient condition (WSC)* for a formula, which is computed by forgetting. The central issue is: does agent a_i know in a state s, from her observable variables O_i, whether a formula ϕ holds (denoted $(KB, s) \models K_i\phi$)?. Here, a *state* s is a valuation of the variables V that satisfies KB. Answering this question boils down to apply forgetting, using results of Lin (2001):

Theorem 2 (Knowledge as WSC; Su et al., 2009) *For any state s and objective formula ϕ,*

$$(KB, s) \models K_i\phi \leftrightarrow WSC_i^{\phi} \;\; \textit{and} \;\; (KB, s) \models \neg K_i \neg\phi \leftrightarrow SNC_i^{\phi},$$

where

- $WSC_i^{\phi} = \neg \mathit{forget}(KB \cup \{\neg\phi\}, V \setminus O_i)$ *is the* weakest sufficient condition *for ϕ w.r.t. a_i, and*
- $SNC_i^{\phi} = \mathit{forget}(KB \cup \{\phi\}, V \setminus O_i)$ *is the* strongest necessary condition *for ϕ w.r.t. a_i.*

We remark that intuitively, the weakest sufficient condition is an explanation for ϕ, based on KB; rewritten to disjunctive normal form $\delta_1 \vee \cdots \vee \delta_m$, each disjunct δ_i is a (stronger) explanation for ϕ as well; in particular, ϕ is the weakest explanation that is formed by disjunction of the strongest such explanations in terms of Ockham's Razor.

We illustrate this approach by an example from (Su et al. 2009).

Example 24 Consider the following knowledge base, which represents information about message sending and acknowledgement betwen two agents *Alice* and *Bob*:

$$KB = \left\{ \begin{array}{l} Bob\_recv\_msg \rightarrow Alice\_send\_msg \\ Bob\_send\_ack \rightarrow Bob\_recv\_msg \\ Alice\_recv\_ack \rightarrow Bob\_send\_ack \end{array} \right\}.$$

Suppose the observables of the agents are

$$\begin{aligned} O_{Bob} &= \{Bob\_recv\_msg, Bob\_send\_ack\}, \\ O_{Alice} &= \{Alice\_send\_msg, Alice\_recv\_ack\}. \end{aligned}$$

From the knowledge base and their observables, the agents can conclude information that they have not observed. For instance, Alice knows *Bob_recv_msg* if *Alice_recv_ack* holds by chaining rules. Then, a sufficient condition for Alice to know *Bob_recv_msg* would be *Alice_recv_ack*, and this is indeed the weakest sufficient condition for this knowledge, given *KB*.

Thus from her observations, Alice can get to know *Bob_recv_msg* only by receiving Bob's acknowledgement. Similarly, Alice can know *Bob_send_ack* only by receiving the acknowledgement.

As to Bob's knowledge, he knows *Alice_send_msg* if *Bob_recv_msg* holds, and by the knowledge base, the latter must necessarily hold to conclude *Alice_send_msg*. Indeed, the only way Bob gets to know *Alice_send_msg* is that he receives Alice's message, so this is the strongest necessary condition. Finally, Bob does not know *Alice_recv_ack*, and Alice obtains her view on Bob's knowledge by computing all she knows about Bob's observations, using the knowledge base.

Now suppose we want to find out whether Alice knows *Bob_recv_msg* in state $s = \{Alice\_recv\_ack\}$, i.e., does $(KB, s) \models K_A\, Bob\_recv\_msg$ hold? We have

$$\begin{aligned} O_{Alice} &= \{Alice\_send\_msg, Alice\_recv\_ack\} \\ \phi &= Bob\_recv\_msg \end{aligned}$$

By Theorem 2, $(KB, s) \models K_i\phi \leftrightarrow \neg forget(KB \cup \{\neg\phi\}, V \setminus O_i)$. We consider the *forget*-formula, where we use $B\_r\_m = Bob\_recv\_msg$, $B\_s\_a = Bob\_send\_ack$, $A\_r\_a = Alice\_recv\_ack$:

$$\begin{aligned} \neg forget(KB \cup \{\neg\phi\}, V \setminus O_i) &= \neg forget(KB \cup \{\neg B\_r\_m\}, \{B\_r\_m, B\_s\_a\}) \\ &= \neg forget(\{\neg A\_r\_a, \neg B\_s\_a, \neg B\_r\_m\}, \{B\_r\_m, B\_s\_a\}) \\ &= \{\neg\neg A\_r\_a\} \\ &= \{A\_r\_a\} \end{aligned}$$

Since *Alice_recv_ack* is true in s, we obtain that $K_A\, Bob\_recv\_msg$ holds.

Reasoning about Actions and Planning. Cognitive robotics is another application domain of forgetting, e.g., for determining cause-effect relationships. Fluents "independent" of taking an action can be forgotten, when the effects of this action are to be determined. By a reverse analysis, one can single out actions that are relevant for achieving a certain change, or effect. Progressing a database means one forgets about the initial state, resp. its state description (Lin and Reiter 1997). Liu and Lakemeyer (2009) applied the notion of forgetting in the situation calculus, obtaining some interesting results about the FOL-definability and computability of progression for local-effect actions; more on forgetting in the situation calculus can be found, e.g., in (Rajaratnam et al. 2014). A kind of abstraction is considered by Erdem and Ferraris (2007) when forgetting actions in planning.

9 Discussion and Conclusion

In the previous sections, we have surveyed forgetting in different formalisms for knowledge representation and reasoning, ranging from classical logic to dedicated languages and to extensions. We have also briefly touched some notions closely related to forgetting, and we have considered different usages of forgetting as a tool for problem solving. This and the rich, continuously growing literature on forgetting provides evidence that forgetting is an important topic in KRR.

On the other hand, the survey also shows that there is no generally accepted precise definition of forgetting; many interpretations and implementations are possible. The most common ones are viewing forgetting as "becoming unaware", or as "becoming ignorant"; but also abstraction may be regarded as a type of forgetting. As a general purpose definition of forgetting seems elusive, a natural question is whether for certain types of forgetting, definitions are possible at an abstract level, such that properties and results established for the notions are inherited by concrete KRR formalisms as instances; in such a view, also a better understanding of the notion of forgetting across different formalisms may be gained.

An important contribution in this direction is the recent work by Delgrande (2014, 2017), who considered forgetting as losing the ability to express, or represent, information about the domain, rather than losing information about a domain per se. He proposed an abstract framework based on logics that have signatures Σ and languages L_Σ of formulas over them, and are equipped with a Tarskian consequence relation $Cn(KB)$ and models $Mod(KB)$ for a knowledge base (set of formulas) KB in L_Σ. For example, classical propositional logic, FOL, modal logic S4, DLs, but also ASP fit this framework.

Forgetting a part $\Sigma' \subseteq \Sigma$ of the signature from a knowledge base KB over Σ amounts to keep the consequences that are on $\Sigma \setminus \Sigma'$, i.e., do not mention any symbol in Σ'. This is called *knowledge level forgetting*, formally defined as

$$forget(KB,\Sigma') = Cn_\Sigma(KB) \cap L_{\Sigma\setminus\Sigma'},$$

and has the following syntactical properties (i.e., with respect to the consequence relation):

Proposition 15 (Knowledge level forgetting: syntax; Delgrande, 2014) *Let KB be a knowledge base in L_Σ for a signature Σ, and let $\Sigma' \subseteq \Sigma$ be a subsignature. Then*

(i) $forget(KB,\Sigma') \subseteq Cn_\Sigma(KB)$ *(consequence loss)*
(ii) If $KB \leftrightarrow KB'$, *then* $forget(KB,\Sigma') \leftrightarrow forget(KB',\Sigma')$ *(irrelevance of syntax)*
(iii) $forget(KB,\Sigma') = Cn_{\Sigma\setminus\Sigma'}(forget(KB,\Sigma'))$ *(closure)*
(iv) $forget(KB,\Sigma_1 \cup \Sigma_2) = forget(forget(KB,\Sigma_1),\Sigma_2)$ *(iterativity)*
(v) $Cn_{\Sigma\setminus\Sigma'}(KB) = Cn_{\Sigma\setminus\Sigma'}(forget(KB,\Sigma'))$, *(conservative extension)*

In particular, (i) states that no new consequences can emerge, and (ii) that the syntactic form of the knowlege base does not matter. Item (iii) states that the result of forgetting is deductively closed, while (iv) says that one can forget stepwise in arbitrary order. Finally (v) ensures that all consequences on the reduced signature are faithfully preserved.

The previous proposition is complemented by the following semantic properties:

Proposition 16 (Knowledge level forgetting: semantics; Delgrande, 2014) *Let KB be a knowledge base in L_Σ for a signature Σ, and let $\Sigma' \subseteq \Sigma$ be a subsignature. Then*

(i) $Mod_{\Sigma\setminus\Sigma'}(forget(KB,\Sigma')) = Mod_\Sigma(KB)|_{\Sigma\setminus\Sigma'}$ *(model projection)*

(ii) $Mod_{\Sigma}(forget(KB,\Sigma')) = (Mod_{\Sigma}(KB)|_{\Sigma\setminus\Sigma'})_{\uparrow\Sigma}$ *(model extension)*

where $|_{\Sigma\setminus\Sigma'}$ *denotes the restriction of each model* M *to* $\Sigma \setminus \Sigma'$*, and* $_{\uparrow\Sigma}$ *are all extensions of the interpretations from* Σ' *to* Σ*.*

Thus by items (i) and (ii), the models of forgetting are obtainable by elementary projection and extension operations on the models of the knowledge base *KB*.

As shown by Delgrande (2014; 2017), a number of proposals for forgetting in various KR formalisms fit this framework; we highly recommend (Delgrande 2017) as an excellent and deeper study of forgetting approaches for logic-based knowledge bases.

Some of the results on forgetting for these formalisms (e.g., essential aspects of relationships) may be understood better in the abstract view. Furthermore, the link to uniform interpolation becomes more apparent: any finite representation $KB' = forget(KB,\Sigma')$ for forgetting, possibly in an enriched language, would be a uniform interpolant for $forget(KB,\Sigma')$. It remains to be seen how far the knowledge level forgetting framework can be applied, and which novel, interesting properties may be obtained at the abstract level.

Since there is no unifying definition of forgetting covering at least its most important aspects, we are far from having an established theory of forgetting. Many further issues need to be addressed. First and maybe most crucial, how to decide what to forget? The cognitive aspects of forgetting are important for agents, dealing both with unaware and deliberate notions of forgetting. Understanding the duality between forgetting and remembrance is particularly important here. On the technical side, we need algorithms and implementations. For this, finding representations of forgetting would be helpful, answering the question whether a finite characterization of $forget(KB,\Sigma')$ exists. Moreover, more work is needed to identify inexpensive fragments of KR formalisms such as decomposable negation normal form, DNNF, for propositional formulas (Darwiche 2001); note that in this context prime-implicate based computation of forgetting appeared to be expensive (Lin 2001). Moreover, software tools that support users in forgetting have to be provided.

Furthermore, connections from forgetting to other operations like filtering, i.e., applying domain-specific relevance, must be elaborated in more detail. Understanding forgetting vs. change is particularly interesting. In this survey, we have considered contraction of belief (e.g., in the AGM framework) as a form of deliberate forgetting. Delgrande (2014) has pointed out that at the knowledge level, forgetting a fact in predicate logic is not meaningful in general, as this is not expressible in terms of a subsignature $\Sigma' \subseteq \Sigma$ that should be forgotten. As an example, after forgetting $student(John)$ still $\exists x.(x = John \wedge student(x))$ would hold, which logically implies $student(John)$. Delgrande thus recommended to view $student(John)$ as *assertion*, and do *belief contraction*. Since contraction is closely related to revision in the AGM framework, relating forgetting to belief change embeds forgetting operations into a more general change scenario where forgetting is part of more comprehensive operations. This may shed some light on the questions why something should be forgotten, and what purposes forgetting is to serve. We have addressed these questions in some places throughout this survey by recalling suitable approaches. However, further work towards a general theory of (all aspects of) forgetting is needed.

Acknowledgements The authors are grateful to the reviewers for their constructive comments to improve this article. This work was partly supported by DFG Priority Programme 1921 *Intentional Forgetting*. The first author is grateful to DFG for supporting his attendance of the internal workshop of the Programme in Limburg, Germany, May 3-4, 2018.

References

Ackermann W (1935) Untersuchungen über das Eliminationsproblem der mathematischen Logik. Mathematische Annalen 110:390–413, URL http://eudml.org/doc/159730

Alassaf R, Schmidt RA (2017) A preliminary comparison of the forgetting solutions computed using scan, LETHE and FAME. In: Koopmann P, Rudolph S, Schmidt RA, Wernhard C (eds) Proceedings of the Workshop on Second-Order Quantifier Elimination and Related Topics (SOQE 2017), Dresden, Germany, December 6-8, 2017., CEUR-WS.org, CEUR Workshop Proceedings, vol 2013, pp 21–26, URL http://ceur-ws.org/Vol-2013/paper12.pdf

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

Alferes JJ, Knorr M, Wang K (2013) Forgetting under the well-founded semantics. In: Cabalar P, Son TC (eds) Logic Programming and Nonmonotonic Reasoning, 12th International Conference, LPNMR 2013, Corunna, Spain, September 15-19, 2013. Proceedings, Springer, Lecture Notes in Computer Science, vol 8148, pp 36–41, DOI 10.1007/978-3-642-40564-8_4, URL https://doi.org/10.1007/978-3-642-40564-8_4

Antoniou G, Eiter T, Wang K (2012) Forgetting for defeasible logic. In: Bjørner N, Voronkov A (eds) Logic for Programming, Artificial Intelligence, and Reasoning - 18th International Conference, LPAR-18, Mérida, Venezuela, March 11-15, 2012. Proceedings, Springer, Lecture Notes in Computer Science, vol 7180, pp 77–91, DOI 10.1007/978-3-642-28717-6_9, URL https://doi.org/10.1007/978-3-642-28717-6_9

Baader F, Calvanese D, McGuinness D, Nardi D, Patel-Schneider PF (eds) (2003) The Description Logic Handbook: Theory, Implementation and Applications. Cambridge University Press

Baral C, Zhang Y (2005) Knowledge updates: Semantics and complexity issues. Artif Intell 164(1-2):209–243, DOI 10.1016/j.artint.2005.01.005, URL https://doi.org/10.1016/j.artint.2005.01.005

Baral C, Giacomo GD, Eiter T (eds) (2014) Principles of Knowledge Representation and Reasoning: Proceedings of the Fourteenth International Conference, KR 2014, Vienna, Austria, July 20-24, 2014, AAAI Press, URL http://www.aaai.org/Library/KR/kr14contents.php

Bertossi LE (2011) Database Repairing and Consistent Query Answering. Synthesis Lectures on Data Management, Morgan & Claypool Publishers, DOI 10.2200/S00379ED1V01Y201108DTM020, URL https://doi.org/10.2200/S00379ED1V01Y201108DTM020

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

Botoeva E, Konev B, Lutz C, Ryzhikov V, Wolter F, Zakharyaschev M (2016) Inseparability and conservative extensions of description logic ontologies: A survey. In: Pan JZ, Calvanese D, Eiter T, Horrocks I, Kifer M, Lin F, Zhao Y (eds) Reasoning Web: Logical Foundation of Knowledge Graph Construction and Query Answering - 12th International Summer School 2016, Aberdeen, UK, September 5-9, 2016, Tutorial Lectures, Springer, Lecture Notes in Computer Science, vol 9885, pp 27–89, DOI 10.1007/978-3-319-49493-7_2, URL https://doi.org/10.1007/978-3-319-49493-7_2

Boutilier C (1994) Toward a logic for qualitative decision theory. In: Doyle J, Sandewall E, Torasso P (eds) Proceedings of the 4th International Conference on Principles of Knowledge Representation and Reasoning (KR'94). Bonn, Germany, May 24-27, 1994., Morgan Kaufmann, pp 75–86

Boutilier C (ed) (2009) IJCAI 2009, Proceedings of the 21st International Joint Conference on Artificial Intelligence, Pasadena, California, USA, July 11-17, 2009, URL http://ijcai.org/proceedings/2009

Brewka G, Eiter T, Truszczynski M (2011a) Answer set programming at a glance. Commun ACM 54(12):92–103, DOI 10.1145/2043174.2043195, URL http://doi.acm.org/10.1145/2043174.2043195

Brewka G, Marek V, Truszczynski M (eds) (2011b) Nonmonotonic Reasoning. Essays Celebrating Its 30th Anniversary. College Publications, London, UK

Brewka G, Eiter T, McIlraith SA (eds) (2012) Principles of Knowledge Representation and Reasoning: Proceedings of the Thirteenth International Conference, KR 2012, Rome, Italy, June 10-14, 2012, AAAI Press

Cabalar P, Pearce D, Valverde A (2018) Answer set programming from a logical point of view. KI - Künstliche Intelligenz DOI 10.1007/s13218-018-0547-7, URL https://doi.org/10.1007/s13218-018-0547-7

Cadoli M, Donini F, Schaerf M, Silvestri R (2000) On compact representations of propositional circumscription. Theoretical Computer Science 182(1-2):183–202

Craig W (1957) Three uses of the Herbrand-Gentzen theorem in relating model theory and proof theory. Journal of Symbolic Logic 22(3):269–285, DOI 10.2307/2963594

Darwiche A (2001) Decomposable negation normal form. J ACM 48(4):608–647, DOI 10.1145/502090.502091, URL http://doi.acm.org/10.1145/502090.502091

Delgrande JP (2014) Towards a knowledge level analysis of forgetting. In: Baral et al. (2014), URL http://www.aaai.org/ocs/index.php/KR/KR14/paper/view/7979

Delgrande JP (2017) A knowledge level account of forgetting. J Artif Intell Res 60:1165–1213, DOI 10.1613/jair.5530, URL https://doi.org/10.1613/jair.5530

Delgrande JP, Wang K (2015) A syntax-independent approach to forgetting in disjunctive logic programs. In: Bonet B, Koenig S (eds) Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence, January 25-30, 2015, Austin, Texas, USA., AAAI Press, pp 1482–1488, URL http://www.aaai.org/ocs/index.php/AAAI/AAAI15/paper/view/9555

van Ditmarsch H, Herzig A, Lang J, Marquis P (2009) Introspective forgetting. Synthese 169(2):405–423, DOI 10.1007/s11229-009-9554-4, URL https://doi.org/10.1007/s11229-009-9554-4

Doherty P, Lukaszewicz W, Madalinska-Bugaj E (1998) The PMA and relativizing minimal change for action update. In: Cohn AG, Schubert LK, Shapiro SC (eds) Proceedings of the Sixth International Conference on Principles of Knowledge Representation and Reasoning (KR'98), Trento, Italy, June 2-5, 1998., Morgan Kaufmann, pp 258–269

Ebbinghaus H (1885) Über das Gedächtnis. Untersuchungen zur experimentellen Psychologie. Duncker & Humblot, Leipzig

Eiter T, Gottlob G (1993) Propositional circumscription and extended closed-world reasoning are Π_2^p-complete. Theor Comput Sci 114(2):231–245, DOI 10.1016/0304-3975(93)90073-3, URL https://doi.org/10.1016/0304-3975(93)90073-3

Eiter T, Wang K (2008) Semantic forgetting in answer set programming. Artif Intell 172(14):1644–1672, DOI 10.1016/j.artint.2008.05.002, URL https://doi.org/10.1016/j.artint.2008.05.002

Eiter T, Tompits H, Woltran S (2005) On solution correspondences in answer-set programming. In: Kaelbling and Saffiotti (2005), pp 97–102, URL http://ijcai.org/Proceedings/05/Papers/1177.pdf

Erdem E, Ferraris P (2007) Forgetting actions in domain descriptions. In: Proceedings of the Twenty-Second AAAI Conference on Artificial Intelligence, July 22-26, 2007, Vancouver, British Columbia, Canada, AAAI Press, pp 409–414, URL http://www.aaai.org/Library/AAAI/2007/aaai07-064.php

Fagin R, Halpern J, Moses Y, Vardi M (1995) Reasoning about Knowledge, 2nd edn. MIT Press, Cambridge, Massachusetts

Fang L, Liu Y, van Ditmarsch H (2016) Forgetting in multi-agent modal logics. In: Kambhampati S (ed) Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence, IJCAI 2016, New York, NY, USA, 9-15 July 2016, IJCAI/AAAI Press, pp 1066–1073, URL http://www.ijcai.org/Abstract/16/155

Fuhrmann A, Hansson SO (1994) A survey of multiple contractions. Journal of Logic, Language and Information 3:39–75

Gabbay DM, Pearce D, Valverde A (2011) Interpolable formulas in equilibrium logic and answer set programming. J Artif Intell Res 42:917–943, URL http://jair.org/papers/paper3329.html

Gärdenfors P, Rott H (1994) Belief revision. In: Gabbay D, Hogger C, Robinson J (eds) Handbook of Logic in Artificial Intelligence and Logic Programming, Oxford University Press, pp 35–132

Ghilardi S, Lutz C, Wolter F (2006a) Did I damage my ontology? A case for conservative extensions in description logics. In: Doherty P, Mylopoulos J, Welty CA (eds) Proceedings, Tenth International Conference on Principles of Knowledge Representation and Reasoning, Lake District of the United Kingdom, June 2-5, 2006, AAAI Press, pp 187–197, URL http://www.aaai.org/Library/KR/2006/kr06-021.php

Ghilardi S, Lutz C, Wolter F, Zakharyaschev M (2006b) Conservative extensions in modal logic. In: Governatori G, Hodkinson IM, Venema Y (eds) Advances in Modal Logic 6, papers from the sixth conference on "Advances in Modal Logic," held in Noosa, Queensland, Australia, on 25-28 September 2006, College Publications, pp 187–207, URL http://www.aiml.net/volumes/volume6/Ghilardi-Lutz-Wolter-Zakharyaschev.ps

Gonçalves R, Knorr M, Leite J (2016a) The ultimate guide to forgetting in answer set programming. In: Baral C, Delgrande JP, 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., AAAI Press, pp 135–144, URL http://www.aaai.org/ocs/index.php/KR/KR16/paper/view/12849

Gonçalves R, Knorr M, Leite J (2016b) You can't always forget what you want: On the limits of forgetting in answer set programming. In: Kaminka GA, Fox M, Bouquet P, Hüllermeier E, Dignum V, Dignum F, van Harmelen F (eds) ECAI 2016 - 22nd European Conference on Artificial Intelligence, 29 August-2 September 2016, The Hague, The Netherlands - Including Prestigious Applications of Artificial Intelligence (PAIS 2016), IOS Press, Frontiers in Artificial Intelligence and Applications, vol 285, pp 957–965, DOI 10.3233/978-1-61499-672-9-957, URL https://doi.org/10.3233/978-1-61499-672-9-957

Grau BC, Motik B (2010) Pushing the limits of reasoning over ontologies with hidden content. In: Lin F, Sattler U, Truszczynski M (eds) Principles of Knowledge Representation and Reasoning: Proceedings of the Twelfth International Conference, KR 2010, Toronto, Ontario, Canada, May 9-13, 2010, AAAI Press, URL http://aaai.org/ocs/index.php/KR/KR2010/paper/view/1244

Hansson S (1999) A textbook of belief dynamics. Kluwer Academic Publishers, Dordrecht, Netherlands

Herzig A, Rifi O (1999) Propositional belief base update and minimal change. Artif Intell 115(1):107–138, DOI 10.1016/S0004-3702(99)00072-7, URL https://doi.org/10.1016/S0004-3702(99)00072-7

Hintikka J (1962) Knowledge and Belief. Cornell University Press, Ithaca, NY

Kaelbling LP, Saffiotti A (eds) (2005) IJCAI-05, Proceedings of the Nineteenth International Joint Conference on Artificial Intelligence, Edinburgh, Scotland, UK, July 30 - August 5, 2005, Professional Book Center, URL http://ijcai.org/proceedings/2005

Katsuno H, Mendelzon AO (1991a) On the Difference between Updating a Knowledge Base and Revising it. In: Proceedings Second International Conference on Principles of Knowledge Representation and Reasoning (KR-91), pp 387 – 395

Katsuno H, Mendelzon AO (1991b) Propositional Knowledge Base Revision and Minimal Change. Artificial Intelligence 52:253–294

Kern-Isberner G, Bock T, Sauerwald K, Beierle C (2017) Iterated contraction of propositions and conditionals under the principle of conditional preservation. In: Benzmüller C, Lisetti C, Theobald M (eds) Proceedings 3rd Global Conference on Artificial Intelligence, GCAI 2017, EPiC Series in Computing, vol 50, pp 78–92

Klir G, Yuan B (eds) (1996) Fuzzy Sets, Fuzzy Logic, Fuzzy Systems. Selected Papers by Lotfi A Zadeh. World Scientific Press

Knorr M, Alferes JJ (2014) Preserving strong equivalence while forgetting. In: Fermé E, Leite J (eds) Logics in Artificial Intelligence - 14th European Conference, JELIA 2014, Funchal, Madeira, Portugal, September 24-26, 2014. Proceedings, Springer, Lecture Notes in Computer Science, vol 8761, pp 412–425, DOI 10.1007/978-3-319-11558-0_29, URL https://doi.org/10.1007/978-3-319-11558-0_29

Konev B, Walther D, Wolter F (2008) The logical difference problem for description logic terminologies. In: Armando A, Baumgartner P, Dowek G (eds) Automated Reasoning, 4th International Joint Conference, IJCAR 2008, Sydney, Australia, August 12-15, 2008, Proceedings, Springer, Lecture Notes in Computer Science, vol 5195, pp 259–274, DOI 10.1007/978-3-540-71070-7_21, URL https://doi.org/10.1007/978-3-540-71070-7_21

Konev B, Walther D, Wolter F (2009) Forgetting and uniform interpolation in large-scale description logic terminologies. In: Boutilier (2009), pp 830–835, URL http://ijcai.org/Proceedings/09/Papers/142.pdf

Konev B, Lutz C, Walther D, Wolter F (2013) Model-theoretic inseparability and modularity of description logic ontologies. Artif Intell 203:66–103, DOI 10.1016/j.artint.2013.07.004, URL https://doi.org/10.1016/j.artint.2013.07.004

Konieczny S, Pino Pérez R (2017) On iterated contraction: syntactic characterization, representation theorem and limitations of the levi identity. In: Moral S, Pivert O, Sánchez D, Marín N (eds) Proceedings of the 11th International Conference of Scalable Uncertainty Management, SUM 2017, Springer, Lecture Notes in Computer Science, vol 10564, pp 348–362

Koopmann P (2015) Practical uniform interpolation for expressive description logics. PhD thesis, University of Manchester, UK, URL http://ethos.bl.uk/OrderDetails.do?uin=uk.bl.ethos.674705

Koopmann P, Schmidt RA (2013) Uniform interpolation of -ontologies using fixpoints. In: Fontaine P, Ringeissen C, Schmidt RA (eds) Frontiers of Combining Systems - 9th International Symposium, FroCoS 2013, Nancy, France, September 18-20, 2013. Proceedings, Springer, Lecture Notes in Computer Science, vol 8152, pp 87–102, DOI 10.1007/978-3-642-40885-4_7, URL https://doi.org/10.1007/978-3-642-40885-4_7

Kripke SA (1959) A completeness theorem in modal logic. Journal of Symbolic Logic 24(1):1–14, DOI 10.2307/2964568

Kripke SA (1963) Semantical analysis of modal logic I: Normal modal propositional calculi. Zeitschrift für Mathematische Logik und Grundlagen der Mathematik 9(5-6):67–96, DOI 10.1002/malq.19630090502, URL https://onlinelibrary.wiley.com/doi/abs/10.1002/malq.19630090502, https://onlinelibrary.wiley.com/doi/pdf/10.1002/malq.19630090502

Lakemeyer G (1997) Relevance from an epistemic perspective. Artif Intell 97(1-2):137–167, DOI 10.1016/S0004-3702(97)00038-6, URL https://doi.org/10.1016/S0004-3702(97)00038-6

Lang J, Marquis P (2002) Resolving inconsistencies by variable forgetting. In: Fensel D, Giunchiglia F, McGuinness DL, Williams M (eds) Proceedings of the Eights International Conference on Principles and Knowledge Representation and Reasoning (KR-02), Toulouse, France, April 22-25, 2002, Morgan Kaufmann, pp 239–250

Lang J, Liberatore P, Marquis P (2003) Propositional independence: Formula-variable independence and forgetting. J Artif Intell Res 18:391–443, DOI 10.1613/jair.1113, URL https://doi.org/10.1613/jair.1113

Leite J (2017) A bird's-eye view of forgetting in answer-set programming. In: Balduccini M, Janhunen T (eds) Logic Programming and Nonmonotonic Reasoning - 14th International Conference, LPNMR 2017, Espoo, Finland, July 3-6, 2017, Proceedings, Springer, Lecture Notes in Computer Science, vol 10377, pp 10–22, DOI 10.1007/978-3-319-61660-5_2, URL https://doi.org/10.1007/978-3-319-61660-5_2

Lenat DB, Feigenbaum EA (1987) On the thresholds of knowledge. In: McDermott JP (ed) Proceedings of the 10th International Joint Conference on Artificial Intelligence. Milan, Italy, August 23-28, 1987, Morgan Kaufmann, pp 1173–1182, URL http://ijcai.org/Proceedings/87-2/Papers/122.pdf

Lifschitz V, Pearce D, Valverde A (2001) Strongly equivalent logic programs. ACM Trans Comput Log 2(4):526–541, DOI 10.1145/502166.502170, URL http://doi.acm.org/10.1145/502166.502170

Lin F (2001) On strongest necessary and weakest sufficient conditions. Artif Intell 128(1-2):143–159, DOI 10.1016/S0004-3702(01)00070-4, URL https://doi.org/10.1016/S0004-3702(01)00070-4

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

Lin F, Reiter R (1997) How to progress a database. Artif Intell 92(1-2):131–167, DOI 10.1016/S0004-3702(96)00044-6, URL https://doi.org/10.1016/S0004-3702(96)00044-6

Liu Y, Lakemeyer G (2009) On first-order definability and computability of progression for local-effect actions and beyond. In: Boutilier (2009), pp 860–866, URL http://ijcai.org/Proceedings/09/Papers/147.pdf

Ludwig M, Konev B (2014) Practical uniform interpolation and forgetting for ALC tboxes with applications to logical difference. In: Baral et al. (2014), URL http://www.aaai.org/ocs/index.php/KR/KR14/paper/view/7985

Lutz C, Wolter F (2011) Foundations for uniform interpolation and forgetting in expressive description logics. In: Walsh T (ed) IJCAI 2011, Proceedings of the 22nd International Joint Conference on Artificial Intelligence, Barcelona, Catalonia, Spain, July 16-22, 2011, IJCAI/AAAI, pp 989–995, DOI 10.5591/978-1-57735-516-8/IJCAI11-170, URL https://doi.org/10.5591/978-1-57735-516-8/IJCAI11-170

Lutz C, Seylan I, Wolter F (2012) An automata-theoretic approach to uniform interpolation and approximation in the description logic EL. In: Brewka et al. (2012), URL http://www.aaai.org/ocs/index.php/KR/KR12/paper/view/4511

Marek W, Truszczyński M (1993) Nonmonotonic Logics – Context-Dependent Reasoning. Springer

McCarthy J (1959) Programs with common sense. In: Proc. of the Teddington Conference on the Mechanization of Thought Processes, 1958, Her Majesty's Stationery Office, London, pp 75–91

McCarthy J (1963) Situations, actions and causal laws. Tech. rep., Stanford Artificial Intelligence Project: Memo 2

McCarthy J (1986) Applications of circumscription to formalizing common-sense knowledge. Artificial Intelligence 28:89–116

Newell A, Simon HA (1976) Computer science as empirical inquiry: Symbols and search. Commun ACM 19(3):113–126, DOI 10.1145/360018.360022, URL http://doi.acm.org/10.1145/360018.360022

Nikitina N, Rudolph S (2012) Expexpexplosion: Uniform interpolation in general EL terminologies. In: Raedt LD, Bessière C, Dubois D, Doherty P, Frasconi P, Heintz F, Lucas PJF (eds) ECAI 2012 - 20th European Conference on Artificial Intelligence. Including Prestigious Applications of Artificial Intelligence (PAIS-2012) System Demonstrations Track, Montpellier, France, August 27-31 , 2012, IOS Press, Frontiers in Artificial Intelligence and Applications, vol 242, pp 618–623, DOI 10.3233/978-1-61499-098-7-618, URL https://doi.org/10.3233/978-1-61499-098-7-618

Ohlbach HJ, Engel T, Schmidt RA, Gabbay DM (2018) Scan: Home page. URL http://www.mettel-prover.org/scan/index.html, online, accessed August 28, 2018

Pearce D (2006) Equilibrium logic. Ann Math Artif Intell 47(1-2):3–41, DOI 10.1007/s10472-006-9028-z, URL https://doi.org/10.1007/s10472-006-9028-z

Pearl J (1988) Probabilistic Reasoning in Intelligent Systems. Morgan Kaufmann, San Mateo, Ca.

Peckhaus V (2004) Calculus ratiocinator versus characteristica universalis? the two traditions in logic, revisited. History and Philosophy of Logic 25(1):3–14, DOI 10.1080/01445340310001609315, URL https://doi.org/10.1080/01445340310001609315, https://doi.org/10.1080/01445340310001609315

Rajaratnam D, Levesque HJ, Pagnucco M, Thielscher M (2014) Forgetting in action. In: Baral et al. (2014), URL http://www.aaai.org/ocs/index.php/KR/KR14/paper/view/7983

Schaub T, Woltran S (2018) Answer set programming unleashed! KI - Künstliche Intelligenz DOI 10.1007/s13218-018-0550-z, URL https://doi.org/10.1007/s13218-018-0550-z, special issue of KI on Answer Set Programming (editorial)

Su K, Sattar A, Lv G, Zhang Y (2009) Variable forgetting in reasoning about knowledge. J Artif Intell Res 35:677–716, DOI 10.1613/jair.2750, URL https://doi.org/10.1613/jair.2750

Wang K, Sattar A, Su K (2005) A theory of forgetting in logic programming. In: Veloso MM, Kambhampati S (eds) Proceedings, The Twentieth National Conference on Artificial Intelligence and the Seventeenth Innovative Applications of Artificial Intelligence Conference, July 9-13, 2005, Pittsburgh, Pennsylvania, USA, AAAI Press / The MIT Press, pp 682–688, URL http://www.aaai.org/Library/AAAI/2005/aaai05-107.php

Wang Y, Zhang Y, Zhou Y, Zhang M (2012) Forgetting in logic programs under strong equivalence. In: Brewka et al. (2012), URL http://www.aaai.org/ocs/index.php/KR/KR12/paper/view/4389

Wang Y, Wang K, Zhang M (2013) Forgetting for answer set programs revisited. In: Rossi F (ed) IJCAI 2013, Proceedings of the 23rd International Joint Conference on Artificial Intelligence, Beijing, China, August 3-9, 2013, IJCAI/AAAI, pp 1162–1168, URL http://www.aaai.org/ocs/index.php/IJCAI/IJCAI13/paper/view/6807

Wang Z, Wang K, Topor RW, Pan JZ (2008) Forgetting concepts in dl-lite. In: Bechhofer S, Hauswirth M, Hoffmann J, Koubarakis M (eds) The Semantic Web: Research and Applications, 5th European Semantic Web Conference, ESWC 2008, Tenerife, Canary Islands, Spain, June 1-5, 2008, Proceedings, Springer, Lecture Notes in Computer Science, vol 5021, pp 245–257, DOI 10.1007/978-3-540-68234-9_20, URL https://doi.org/10.1007/978-3-540-68234-9_20

Weber A (1986) Updating propositional formulas. In: Proceedings of the First Conference on Expert Database Systems, pp 487–500

Wikipedia (2018) URL https://de.wikipedia.org/wiki/Martin_Lampe

Wong KS (2009) Forgetting in logic programs. PhD thesis, School of Computer Science and Engineering, University of New South Wales

Xiao P, Wang K, Wang Z (2017) Incosistency-tolerant forgetting in dl-lite. In: Nikitina N, Song D, Fokoue A, Haase P (eds) Proceedings of the ISWC 2017 Posters & Demonstrations and Industry Tracks co-located with 16th International Semantic Web Conference (ISWC 2017), Vienna, Austria, October 23rd - to - 25th, 2017., CEUR-WS.org, CEUR Workshop Proceedings, vol 1963, URL http://ceur-ws.org/Vol-1963/paper649.pdf

Zhang Y, Zhou Y (2009) Knowledge forgetting: Properties and applications. Artif Intell 173(16-17):1525–1537, DOI 10.1016/j.artint.2009.07.005, URL https://doi.org/10.1016/j.artint.2009.07.005

Zhang Y, Zhou Y (2010) Forgetting revisited. In: Lin F, Sattler U, Truszczynski M (eds) (2010) Principles of Knowledge Representation and Reasoning: Proceedings of the Twelfth International Conference, KR 2010, Toronto, Ontario, Canada, May 9-13, 2010, AAAI Press URL http://aaai.org/ocs/index.php/KR/KR2010/paper/view/1292

Zhang Y, Foo NY, Wang K (2005) Solving logic program conflict through strong and weak forgettings. In: Kaelbling and Saffiotti (2005), pp 627–634, URL http://ijcai.org/Proceedings/05/Papers/0488.pdf

Zhao Y, Schmidt RA (2017) Role forgetting for alcoqh(universal role)-ontologies using an ackermann-based approach. In: Sierra C (ed) Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI 2017, Melbourne, Australia, August 19-25, 2017, ijcai.org, pp 1354–1361, DOI 10.24963/ijcai.2017/188, URL https://doi.org/10.24963/ijcai.2017/188

Zhao Y, Schmidt RA (2018a) FAME: an automated tool for semantic forgetting in expressive description logics. In: Galmiche D, Schulz S, Sebastiani R (eds) Automated Reasoning - 9th International Joint Conference, IJCAR 2018, Held as Part of the Federated Logic Conference, FloC 2018, Oxford, UK, July 14-17, 2018, Proceedings, Springer, Lecture Notes in Computer Science, vol 10900, pp 19–27, DOI 10.1007/978-3-319-94205-6_2, URL https://doi.org/10.1007/978-3-319-94205-6_2

Zhao Y, Schmidt RA (2018b) On concept forgetting in description logics with qualified number restrictions. In: Lang J (ed) Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI 2018, July 13-19, 2018, Stockholm, Sweden., ijcai.org, pp 1984–1990, DOI 10.24963/ijcai.2018/274, URL https://doi.org/10.24963/ijcai.2018/274