Artificial Intelligence 326 (2024) 104036

Contents lists available at ScienceDirect

Artificial Intelligence

journal homepage: www.elsevier.com/locate/artint

Dual forgetting operators in the context of weakest sufficient and strongest necessary conditions

Patrick Doherty [b,a], Andrzej Szałas [a,c,*]

[a] Department of Computer and Information Science, Linköping University, SE-581 83 Linköping, Sweden
[b] Faculty of Informatics, Mahasarakham University, Mahasarakham, Thailand
[c] Institute of Informatics, University of Warsaw, Banacha 2, 02-097 Warsaw, Poland

ARTICLE INFO

Keywords:
Knowledge representation and reasoning
Forgetting
Weakest sufficient conditions
Strongest necessary conditions
Quantifier elimination

ABSTRACT

Forgetting is an important concept in knowledge representation and automated reasoning with widespread applications across a number of disciplines. A standard forgetting operator, characterized in [26] in terms of model-theoretic semantics and primarily focusing on the propositional case, opened up a new research subarea. In this paper, a new operator called *weak forgetting*, dual to standard forgetting, is introduced and both together are shown to offer a new more uniform perspective on forgetting operators in general. Both the weak and standard forgetting operators are characterized in terms of entailment and inference, rather than a model theoretic semantics. This naturally leads to a useful algorithmic perspective based on quantifier elimination and the use of Ackermann's Lemma and its fixpoint generalization. The strong formal relationship between standard forgetting and strongest necessary conditions and weak forgetting and weakest sufficient conditions is also characterized quite naturally through the entailment-based, inferential perspective used. The framework used to characterize the dual forgetting operators is also generalized to the first-order case and includes useful algorithms for computing first-order forgetting operators in special cases. Practical examples are also included to show the importance of both weak and standard forgetting in modeling and representation.

1. Introduction and motivation

From a knowledge representation and automated reasoning perspective, *remembering* is essentially what an agent system does when adding new logical statements to a knowledge or belief base. Remembering, in this context, is a powerful way of modeling and is the basis for decision making in many agent systems. On the surface, remembering appears to be straightforward, simply add a new statement to a knowledge or belief base. But what if one wants to retain consistency or some other property of the knowledge or belief base upon assertion of additional statements? The knowledge or belief base would then need to be modified in various ways. Then the problem becomes more complex and leads to different subareas in Knowledge Representation, such as belief revision [3,18,29] or research with consistency preserving operators [3,24].

This setting also leads naturally to the dual concept of *forgetting*. Given a knowledge or belief base, what does it mean to forget parts of it permanently, or temporarily for reasons of expedience? Here, on the surface also, forgetting appears to be straightforward,

[*] Corresponding author at: Department of Computer and Information Science, Linköping University, SE-581 83 Linköping, Sweden.
E-mail addresses: patrick.doherty@liu.se (P. Doherty), andrzej.szalas@liu.se (A. Szałas).

https://doi.org/10.1016/j.artint.2023.104036
Received 15 May 2023; Received in revised form 18 October 2023; Accepted 19 October 2023

Available online 23 October 2023

simply remove a statement from a knowledge or belief base. But, as in the case of remembering, there is a great deal of subtlety and choice concerning why and how one might remove a statement, or parts of statements from a knowledge or belief base.

The spectrum between explicit remembering and explicit forgetting and the operators that would be needed for specifying the different degrees in between, offer a complex set of research topics in Knowledge Representation. Lin and Reiter [26] opened up a new subarea of Knowledge Representation with the introduction of a (*standard*) forgetting operator applied to a knowledge or belief base. The forgetting operator is specified in terms of model-theoretic semantical criteria, as are its properties. The focus in their work is primarily propositional, but there is consideration of the first-order case. The basic question asked and answered is “what does it mean to forget certain concepts (propositional variables) in a knowledge base and how does this influence entailment of formulas in that knowledge base?”.

This context is the starting point for this paper. Here the interest is in exploring whether there are other well-behaved forgetting operators in the spectrum discussed above and how they may relate to the original standard forgetting operator. Such operators should also be useful representationally and pragmatically.

Consider the following motivating example where the need for an additional forgetting operator is considered. Let lt and lp stand for “low temperature” and “low pressure”, respectively. Assume we are modeling a physical system and want to maintain the property:

$$lt \vee lp. \tag{1}$$

Consider a situation when a temperature sensor associated with the system is broken and we receive no meaningful information about lt. To adapt the model for this situation, we would then want to temporarily forget lt. According to standard approaches to forgetting [26], this would result in the second-order formula:

$$\exists lt \left(lt \vee lp \right). \tag{2}$$

Formula (2) is equivalent to *true*, so this would leave us empty-handed when reasoning about maintaining (1) with the associated changes in the system. No additional consequences of the change in the system can be derived. This happens since standard forgetting has the property of preserving entailment (see Proposition 10, point 2 in [26]), where one is interested in what a given theory entails, i.e., in the necessary conditions of the theory. On the other hand, in this situation one would expect that lp itself should still be maintained, as it is a *sufficient* condition for (1). However, this weaker form of reasoning is not covered by standard forgetting.

Although this is a simple example, it allows us to target what this paper is about. We are interested in this weaker form of reasoning associated with forgetting and its relation to the stronger standard form of reasoning with forgetting and how these two forms of forgetting can be used in various applications.

The original contributions of the paper include:

- complementing the standard forgetting operator with a new one, the *weak* forgetting operator, that is dual to standard forgetting, useful in applications and, surprisingly, not explicitly considered in the literature so far[1];
- specifying forgetting operators in a general, principled framework that is directly related to entailment and inference rather than through a semantic construction of model equivalence as is typically used in defining standard forgetting (see, e.g., [26, Definition 1] and other related work);
- the formal framework introduced shows the strong dual relationship between standard forgetting and strongest necessary conditions, and weak forgetting and weakest sufficient conditions. This relationship follows naturally from the entailment-based, inferential perspective used;
- a computational framework that leverages the inferential perspective and is used for computing the result of forgetting operators is presented. It is based on the use of Ackermann’s Lemma and tautology preserving formula transformations. This framework is introduced for the propositional case of forgetting and then later extended to the first-order case;
- it is also shown that computing the propositional or first-order (or fixpoint) equivalent of the dual weak forgetting operator is typically more efficient than computing the standard forgetting operator, or than computing both the weakest and strongest necessary conditions. In the light of complexity results on standard forgetting (see, e.g., [23]), even from that one standpoint alone, it is beneficial to consider the dual weak forgetting operator as a separate operator when it is feasible to use representationally.

The rest of the paper is structured as follows. Section 2 presents preliminaries related to propositional logic and standard forgetting. Section 3 discusses forgetting operators in general and provides their second-order characterization. In Section 4, some examples are presented illustrating the approach, where the weak forgetting operator is shown to be very useful representationally. Section 5 considers the strong relationship between the dual forgetting operators and strongest necessary and weakest sufficient conditions. Section 6 shows how the formalism using dual forgetting operators can be extended to the first-order case, whereas the approach to standard forgetting and its use has been predominantly propositional in nature. In Section 7, we discuss relevant, related work. Finally, Section 8 concludes the paper with a summary and some final remarks.

[1] In the paper, we call the new operator the *weak* forgetting operator and the standard operator found in the literature, either the *standard* or the *strong* forgetting operator.

2. Preliminaries

2.1. Classical propositional logic

For the sake of simplicity, we initially present ideas starting with classical propositional logic, $\mathcal{L}_0$, with truth constants $\mathbb{T}$ (true) and $\mathbb{F}$ (false), an enumerable set of propositional variables $\mathcal{V}_0$, and standard connectives $\neg, \wedge, \vee, \rightarrow, \equiv$. We shall also use second-order quantifiers $\exists p, \forall p$, where $p \in \mathcal{V}_0$. The meaning of quantifiers in the propositional context is:

$$\exists p\big(A(p)\big) \overset{\text{def}}{\equiv} A(p=\mathbb{F}) \vee A(p=\mathbb{T}); \tag{3}$$

$$\forall p\big(A(p)\big) \overset{\text{def}}{\equiv} A(p=\mathbb{F}) \wedge A(p=\mathbb{T}), \tag{4}$$

where $A(p = expr)$ denotes a formula obtained from A by substituting all occurrences of p in A by expression $expr$.

Remark 2.1. Throughout the paper, we deal with first-order and second-order quantifiers. The former bind variables ranging over domain elements (first-order variables). The latter bind variables representing formulas (second-order variables), frequently just propositional or relational variables. □

By a *theory* we mean a finite set of formulas.[2] A theory is identified with a conjunction of formulas it contains. We often write $\bar{p}$ to denote a tuple of propositional variables, and $Th(\bar{p})$ to indicate that theory Th is formed over a vocabulary consisting of variables in $\bar{p}$. Similarly, we often write $A(\bar{p})$ to indicate that formula A is formed over a vocabulary consisting of $\bar{p}$. When needed, we separate variables to indicate those bound by quantifiers and those that are not. In such cases we write $Th(\bar{p},\bar{q})$, $A(\bar{p},\bar{q})$ to indicate that their vocabulary consists of tuples $\bar{p},\bar{q}$. If $\bar{p} = \langle p_1,\dots,p_k\rangle$ is a k-tuple of propositional variables then $\exists\bar{p}$ abbreviates $\exists p_1 \dots \exists p_k$.

We say that a formula A is *at least as strong* as a formula B (wrt →), if $A \rightarrow B$ is a tautology ($\models A \rightarrow B$). In such a case, we also say that B is *at least as weak* than A (wrt →), A is a *sufficient condition* for B, and B is a *necessary condition* for A.

A formula $A(p)$ is *positive* wrt $p \in \mathcal{V}_0$, if all occurrences of p in A are in the scope of an even number of negations.[3] $A(p)$ is *negative* wrt $p \in \mathcal{V}_0$, if all occurrences of p in A are in the scope of an odd number of negations. By a *literal*, we mean a propositional variable or its negation.

2.2. Standard forgetting

Standard forgetting has been introduced in [26] using a model-theoretic framework. The intuition behind this operator is to forget a part of the vocabulary of a theory and remember as much as possible using the remaining vocabulary, where logical consequences are concerned. Theorem 8 and Proposition 10 in [26] provide, among others, the following important properties of forgetting, where $\mathit{forget}(Th(\bar{p},\bar{q}),\bar{p})$ denotes *forgetting* about $\bar{p}$ in $Th(\bar{p},\bar{q})$.

Theorem 2.2 *(Lin, Reiter). Let $\bar{p}$ and $\bar{q}$ be disjoint tuples of propositional variables, A be a formula not containing occurrences of variables from $\bar{p}$ and $Th(\bar{p},\bar{q})$ be a theory. Then:*

$$-\ \mathit{forget}(Th(\bar{p},\bar{q}),\bar{p}) \equiv \exists\bar{p}\big(Th(\bar{p},\bar{q})\big); \tag{5}$$

$$-\ Th(\bar{p},\bar{q}) \models A \text{ iff } \mathit{forget}(Th(\bar{p},\bar{q}),\bar{p}) \models A. \quad \square \tag{6}$$

Notice that (5) can serve as an alternative definition of standard forgetting. We therefore omit discussion and use of the model-theoretic definition used in [26]. One can also observe that due to the deduction theorem for classical propositional logic, (6) can be expressed as:

$$\models Th(\bar{p},\bar{q}) \rightarrow A \text{ iff } \models \mathit{forget}(Th(\bar{p},\bar{q}),\bar{p}) \rightarrow A. \tag{7}$$

2.3. An Ackermann-like approach to second-order quantifier elimination

Theorem 2.2(5) indicates that computing forgetting using the Lin&Reiter-like forgetting operator calls for eliminating second-order quantifiers $\exists\bar{p}(\dots)$. While definitions (3)–(4) allow one to eliminate second-order quantifiers, they lead to an exponential growth of the resulting formula wrt the number of quantifiers. Therefore, as a computationally more appropriate tool, we will use the following lemma of Ackermann, already proved in [2] (see also [11,19]). This usage typically results in much shorter formulas as a result of quantifier eliminations in many cases.

[2] Such finite sets of formulas are frequently referred to as *belief bases*. We use the term *theory*, as belief bases suggest finite domains, which is generally not required when we extend our results to the first-order case.

[3] As usual, we consider $B \rightarrow C$ to stand for $\neg B \vee C$, and $B \equiv C$ to stand for $(\neg B \vee C) \wedge (B \vee \neg C)$.

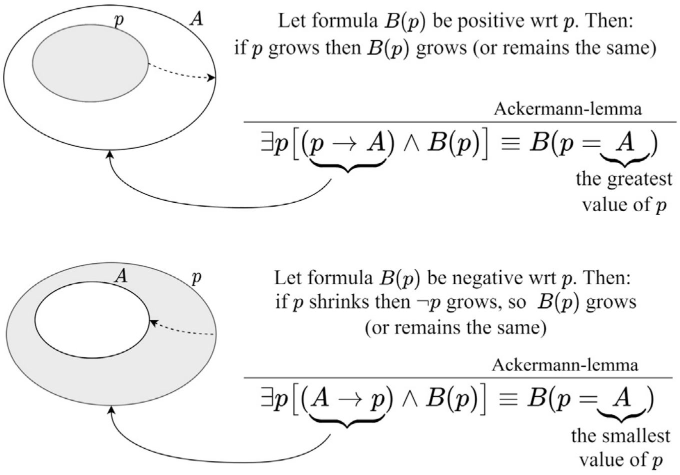
p
A
Let formula $B(p)$ be positive wrt p. Then:
if p grows then $B(p)$ grows (or remains the same)
Ackermann-lemma
$\exists p\big[\underbrace{(p \to A)} \wedge B(p)\big] \equiv B(p = \underbrace{A}_{\text{the greatest value of } p})$
A
p
Let formula $B(p)$ be negative wrt p. Then:
if p shrinks then $\neg p$ grows, so $B(p)$ grows
(or remains the same)
Ackermann-lemma
$\exists p\big[\underbrace{(A \to p)} \wedge B(p)\big] \equiv B(p = \underbrace{A}_{\text{the smallest value of } p})$

Fig. 1. An illustration of intuitions behind Ackermann-like lemmas.

Lemma 2.3 *(Propositional Ackermann lemma). Let A be a propositional formula without occurrences of propositional variable p, and $B(p)$ be a propositional formula on a vocabulary containing p*[4]*:*

- *if $B(p)$ is positive wrt p then:* $\exists p\big((p \to A) \wedge B(p)\big) \;\equiv\; B(p = A)$; (8)
- *if $B(p)$ is negative wrt p then:* $\exists p\big((A \to p) \wedge B(p)\big) \;\equiv\; B(p = A)$. □ (9)

The lemma remains true when p is replaced by a second-order variable, say P, representing propositional formulas. For example, (8) can be formulated as:

- if $B(P)$ is positive wrt P then: $\exists P\big((P \to A) \wedge B(P)\big) \;\equiv\; B(P = A)$.

Lemma 2.3 serves as a blueprint for specifying an algorithm to eliminate second-order quantifiers in many cases. It shows that if one can syntactically transform a formula $F(\bar{p})$, where $p \in \bar{p}$, into an equivalent formula with the syntactic structure on the lefthand side of equivalences (8) or (9), respectively, then one can eliminate $\exists p$ by substitution of A for p in B, resulting in the equivalent $B(p = A)$. This syntactic technique can be iterated for all second-order quantifiers in $F(\bar{p})$, resulting in a logically equivalent propositional formula.

To transform a formula into a form required in (8) or (9), one can use the DLS algorithm of [11]. For propositional formulas, one of the forms in the lefthand sides of equivalences (8) or (9) can always be obtained, thus guaranteeing removal of all second-order quantifiers in an arbitrary propositional theory.

Fig. 1 illustrates the idea behind Ackermann-like lemmas, where the terms "grows" and "shrinks", illustrated by dashed arrows within ovals, refer to the standard ordering on truth values, $\mathbb{F} < \mathbb{T}$, compatible with the semantics of implication, which can be given by[5]:

$$\text{(the truth value of } p \to q) \overset{\text{def}}{=} \text{('the truth value of } p\text{'} \leq \text{'the truth value of } q\text{')}.$$

Given the constraint $p \to A$ in (8) (or, respectively, $A \to p$ in (9)), the greatest value of $B(p)$ is obtained when p takes its greatest (respectively, smallest) value, i.e., the value given by A. Of course, in such cases, the existential quantifier $\exists p(\ldots)$ obtains the greatest value of $B(p)$, achieved for $p = A$. In this case, each of the constraints $p \to A$ (in (8)) as well as $A \to p$ (in (9)) evaluate to $\mathbb{T}$, and so disappear from the result.

3. The entailment-based inferential perspective on forgetting

3.1. Some intuitions

Let $Th(\bar{p}, \bar{q})$ be a propositional theory over a vocabulary consisting of $\bar{p}, \bar{q}$.[6] When forgetting $\bar{p}$ in the theory $Th(\bar{p}, \bar{q})$, one can delineate two alternative views, one existing and one new, with both resulting in a theory expressed in the vocabulary containing $\bar{q}$ only:

[4] For the sake of clarity we assume that B contains p, but the lemma is trivially true also when this is not the case.

[5] More formally, we deal here with the construction of a partial order among formulas for the Lindenbaum and Tarski algebra [21]. The Lindenbaum–Tarski algebra of a theory T consists of the equivalence classes of formulas of the theory, where two formulas are equivalent when the theory T proves that each implies the other. The partial order in question is then defined as: $||A|| \leq_T ||B||$ iff $T \vdash A \to B$, where $||C||$ denotes an equivalence class of a formula C.

[6] When we refer to tuples of variables as arguments, like in $Th(\bar{p}, \bar{q})$, we always assume that $\bar{p}$ and $\bar{q}$ are disjoint.

- *strong (standard) forgetting* $F^{NC}\big(Th(\bar{p},\bar{q});\bar{p}\big)$: a theory that preserves the entailment of necessary conditions over $\bar{q}$;
- *weak forgetting* $F^{SC}\big(Th(\bar{p},\bar{q});\bar{p}\big)$: a theory that preserves the entailment by sufficient conditions over $\bar{q}$.

The rationale behind the operator $F^{NC}\big(Th(\bar{p},\bar{q});\bar{p}\big)$ is that one wants to remember a theory on vocabulary $\bar{q}$, whose consequences are also consequences of the original theory. That is, for any formula A on a vocabulary disjoint with $\bar{p}$,

$$\models Th(\bar{p},\bar{q}) \rightarrow A \text{ iff } \models F^{NC}\big(Th(\bar{p},\bar{q});\bar{p}\big) \rightarrow A. \tag{10}$$

In this case, a formula A is a consequence of the result of forgetting, $F^{NC}()$, iff it is a consequence of the original theory $Th(\bar{p},\bar{q})$. Notice the similarity of (7) and (10) indicating that *forget*() and $F^{NC}()$ act in the same manner. However, while in the approach of [26], the property (7) is a derived theorem, in our approach it is a fundamental starting point.

The rationale behind the weak forgetting operator $F^{SC}\big(Th(\bar{p},\bar{q});\bar{p}\big)$ is that one wants to remember a theory on vocabulary $\bar{q}$ such that a formula implies $F^{SC}\big(Th(\bar{p},\bar{q});\bar{p}\big)$ iff it implies the original theory. That is, for any formula A on a vocabulary disjoint with $\bar{p}$,

$$\models A \rightarrow Th(\bar{p},\bar{q}) \text{ iff } \models A \rightarrow F^{SC}\big(Th(\bar{p},\bar{q});\bar{p}\big), \tag{11}$$

i.e., a formula A implies the result of weak forgetting, $F^{SC}()$, iff it implies the original theory $Th(\bar{p},\bar{q})$.

3.2. The operator $F^{NC}()$

The requirement (10) that $F^{NC}\big(Th(\bar{p},\bar{q});\bar{p}\big)$ preserves the entailment of necessary conditions over a vocabulary disjoint with $\bar{p}$ can be expressed as:

$$\forall P\Big(\forall\bar{p}\big(Th(\bar{p},\bar{q}) \rightarrow P\big) \equiv \big(F^{NC}\big(Th(\bar{p},\bar{q});\bar{p}\big) \rightarrow P\big)\Big), \tag{12}$$

where P is a second-order variable representing an arbitrary formula over a vocabulary disjoint with $\bar{p}$.

Using (12) as a basis, let us derive an important Theorem 3.1, characterizing the operator, $F^{NC}()$.

Theorem 3.1. *For arbitrary tuples of propositional variables $\bar{p},\bar{q}$ and $Th(\bar{p},\bar{q})$,*

1. $F^{NC}\big(Th(\bar{p},\bar{q});\bar{p}\big) \equiv \exists\bar{p}\big(Th(\bar{p},\bar{q})\big)$.
2. $F^{NC}\big(Th(\bar{p},\bar{q});\bar{p}\big) \equiv$ *forget*$(Th(\bar{p},\bar{q});\bar{p})$.
3. $F^{NC}\big(Th(\bar{p},\bar{q}),\bar{p}\big)$ *is the strongest (wrt →) formula over vocabulary $\bar{q}$, satisfying* (12).

Proof. By a standard propositional tautology, we represent equivalence ≡ in (12) as the conjunction of left-to-right (→) and right-to-left (←) implications.

The analysis of the left-to-right direction (→)

Let us start with the analysis of the left-to-right direction (→) of (12). Since variables $\bar{p}$ occur only in $Th(\bar{p},\bar{q})$, the implication is equivalent to:

$$\forall P\Big(\big(\exists\bar{p}\big(Th(\bar{p},\bar{q})\big) \rightarrow P\big) \rightarrow \big(F^{NC}\big(Th(\bar{p},\bar{q});\bar{p}\big) \rightarrow P\big)\Big). \tag{13}$$

In order to apply Lemma 2.3, we have to transform (13) into an equivalent form:

$$\neg\exists P\Big(\big(\exists\bar{p}\big(Th(\bar{p},\bar{q})\big) \rightarrow P\big) \wedge F^{NC}\big(Th(\bar{p},\bar{q});\bar{p}\big) \wedge \neg P\Big). \tag{14}$$

We eliminate the second-order quantifier $\exists P$ from (14) using Lemma 2.3(9). As a result we obtain the following formula equivalent to (14):

$$\neg\Big(F^{NC}\big(Th(\bar{p},\bar{q});\bar{p}\big) \wedge \neg\exists\bar{p}\big(Th(\bar{p},\bar{q})\big)\Big), \tag{15}$$

which in turn is equivalent to:

$$F^{NC}\big(Th(\bar{p},\bar{q});\bar{p}\big) \rightarrow \exists\bar{p}\big(Th(\bar{p},\bar{q})\big). \tag{16}$$

The analysis of the right-to-left direction (←)

For the right-to-left direction (←) of (12) we proceed as follows. Since variables $\bar{p}$ occur only in $Th(\bar{p},\bar{q})$, after moving $\forall\bar{p}$ to the antecedent of $Th(\bar{p},\bar{q}) \rightarrow P$ thus replacing it by $\exists\bar{p}$, the implication (←) is equivalent to:

$$\forall P\Big(\big(F^{NC}\big(Th(\bar{p},\bar{q});\bar{p}\big) \rightarrow P\big) \rightarrow \big(\exists\bar{p}\big(Th(\bar{p},\bar{q})\big) \rightarrow P\big)\Big), \tag{17}$$

which is equivalent to:

$$\neg\exists P\Big(\big(F^{NC}\big(Th(\bar{p},\bar{q});\bar{p}\big) \rightarrow P\big) \wedge \exists\bar{p}\big(Th(\bar{p},\bar{q})\big) \wedge \neg P\Big). \tag{18}$$

As before, we eliminate the second-order quantifier $\exists P$ from (18) using Lemma 2.3(9). As a result, we obtain the following formula equivalent to (18):

$$\neg\Big(\exists\bar{p}\big(Th(\bar{p},\bar{q})\big)\wedge\neg F^{NC}\big(Th(\bar{p},\bar{q});\bar{p}\big)\Big), \tag{19}$$

which in turn is equivalent to:

$$\exists\bar{p}\big(Th(\bar{p},\bar{q})\big)\rightarrow F^{NC}\big(Th(\bar{p},\bar{q});\bar{p}\big). \tag{20}$$

Combining (16) and (20), Point 1 of Theorem 3.1, is derived. Now, Point 2 of Theorem 3.1, follows directly from the characterization formulated in Point 1, together with Theorem 2.2(5). To see that Point 3 of Theorem 3.1 holds, it suffices to notice that, by (20), $F^{NC}\big(Th(\bar{p},\bar{q});\bar{p}\big)$ is no stronger than $\exists\bar{p}\big(Th(\bar{p},\bar{q})\big)$. Since it is proved to be equivalent to $\exists\bar{p}\big(Th(\bar{p},\bar{q})\big)$, it is the strongest (wrt →) formula over vocabulary $\bar{q}$, satisfying (12). □

3.3. *The operator $F^{SC}()$*

By analogy to $F^{NC}()$, the requirement (11) that $F^{SC}\big(Th(\bar{p},\bar{q});\bar{p}\big)$ preserves entailment by sufficient conditions can be expressed as:

$$\forall P\Big(\big(P\rightarrow F^{SC}\big(Th(\bar{p},\bar{q});\bar{p}\big)\big)\equiv\forall\bar{p}\big(P\rightarrow Th(\bar{p},\bar{q})\big)\Big), \tag{21}$$

where P is again a second-order variable representing an arbitrary formula over a vocabulary disjoint with $\bar{p}$.

Using (21) as a basis, let us derive an important Theorem 3.2, characterizing the operator, $F^{SC}()$.

Theorem 3.2. *For arbitrary tuples of propositional variables $\bar{p},\bar{q}$ and $Th(\bar{p},\bar{q})$,*

1. $F^{SC}\big(Th(\bar{p},\bar{q});\bar{p}\big)\equiv\forall\bar{p}\big(Th(\bar{p},\bar{q})\big)$.
2. $F^{SC}\big(Th(\bar{p},\bar{q});\bar{p}\big)$ *is the weakest (wrt →) formula over vocabulary $\bar{q}$, satisfying* (21).

Proof. As in Section 3.2, we represent equivalence ≡ in (21) as the conjunction of two implications (→) and (←).

The analysis of the left-to-right direction (→)

Let us first consider implication (→). Since P is $\bar{p}$ free, the implication is equivalent to:

$$\forall P\Big(\big(P\rightarrow F^{SC}\big(Th(\bar{p},\bar{q});\bar{p}\big)\big)\rightarrow\big(P\rightarrow\forall\bar{p}\big(Th(\bar{p},\bar{q})\big)\big)\Big). \tag{22}$$

To apply Lemma 2.3, we transform (22) to an equivalent form:

$$\neg\exists P\Big(\big(P\rightarrow F^{SC}\big(Th(\bar{p},\bar{q});\bar{p}\big)\big)\wedge P\wedge\neg\forall\bar{p}\big(Th(\bar{p},\bar{q})\big)\Big). \tag{23}$$

We eliminate the second-order quantifier $\exists P$ from (23), using Lemma 2.3(8). As a result, we obtain the following formula equivalent to (23):

$$\neg\Big(F^{SC}\big(Th(\bar{p},\bar{q});\bar{p}\big)\big)\wedge\neg\forall\bar{p}\big(Th(\bar{p},\bar{q})\big)\Big), \tag{24}$$

which in turn is equivalent to:

$$F^{SC}\big(Th(\bar{p},\bar{q});\bar{p}\big)\big)\rightarrow\forall\bar{p}\big(Th(\bar{p},\bar{q})\big). \tag{25}$$

The analysis of the right-to-left direction (←)

The implication (←) of (21) is equivalent to:

$$\forall P\Big(\big(P\rightarrow\forall\bar{p}\big(Th(\bar{p},\bar{q})\big)\big)\rightarrow\big(P\rightarrow F^{SC}\big(Th(\bar{p},\bar{q});\bar{p}\big)\big)\Big). \tag{26}$$

To apply Lemma 2.3, we transform (26) to an equivalent form:

$$\neg\exists P\Big(\big(P\rightarrow\forall\bar{p}\big(Th(\bar{p},\bar{q})\big)\big)\wedge P\wedge\neg F^{SC}\big(Th(\bar{p},\bar{q});\bar{p}\big)\Big). \tag{27}$$

We eliminate the second-order quantifier $\exists P$ from (27), using Lemma 2.3(8). As a result, we obtain the following formula equivalent to (27):

$$\neg\Big(\forall\bar{p}\big(Th(\bar{p},\bar{q})\big)\wedge\neg F^{SC}\big(Th(\bar{p},\bar{q});\bar{p}\big)\Big), \tag{28}$$

which in turn is equivalent to:

$$\forall\bar{p}\big(Th(\bar{p},\bar{q})\big)\rightarrow F^{SC}\big(Th(\bar{p},\bar{q});\bar{p}\big)\big). \tag{29}$$

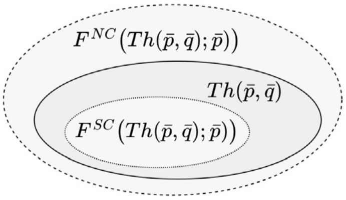
$F^{NC}(Th(\bar{p},\bar{q});\bar{p}))$
$Th(\bar{p},\bar{q})$
$F^{SC}(Th(\bar{p},\bar{q});\bar{p}))$

Fig. 2. The relationships between forgetting operators and the original theory in terms of entailment: the ovals inclusion indicates that the formula in an inner oval entails the formula in an outer one.

Combining (25) and (29), the characterization of $F^{SC}()$ in Point 1, of Theorem 3.2 is derived. To see that Point 2, of Theorem 3.2 holds, it suffices to notice that, by (25), $F^{SC}(Th(\bar{p},\bar{q});\bar{p})$ is no weaker than $\forall\bar{p}(Th(\bar{p},\bar{q}))$. Since it is proved to be equivalent to $\forall\bar{p}(Th(\bar{p},\bar{q}))$, it is the weakest (wrt →) formula over vocabulary $\bar{q}$, satisfying (21). □

3.4. Combining $F^{NC}()$ and $F^{SC}()$ in a dual or complementary perspective

The net result of the analysis provided is that standard or strong forgetting, and weak forgetting are complementary in an intuitive and formally concise manner.

Weak forgetting, $F^{SC}()$, allows one to *remember* more than using solely standard (strong) forgetting, $F^{NC}()$, relative to a specific theory. This can be shown to be very useful in applications of forgetting operators as exhibited in Section 4.

In the original example (1), the weak forgetting operator $F^{SC}()$ can be applied naturally,

$$F^{SC}((lt \vee lp); lt) \equiv \forall lt\,(lt \vee lp) \equiv lp \tag{30}$$

As suggested in Section 1, lt is forgotten, while retaining additional information lp about the physical system.

It is interesting to observe that $F^{NC}()$ and $F^{SC}()$ partition the theory $Th(\bar{p},\bar{q})$ into three nicely related classes of formulas (see Fig. 2):

- the innermost oval, $F^{SC}(Th(\bar{p},\bar{q});\bar{p})$: formulas in vocabulary $\bar{q}$ that imply the original theory (sufficient conditions of the theory);
- the central oval: the original theory $Th(\bar{p},\bar{q})$;
- the outermost oval, $F^{NC}(Th(\bar{p},\bar{q});\bar{p})$: formulas in vocabulary $\bar{q}$ implied by the original theory (necessary conditions of the theory).

Notice that this also implies that $\models F^{SC}(Th(\bar{p},\bar{q});\bar{p}) \to F^{NC}(Th(\bar{p},\bar{q});\bar{p})$ and, by the propositional deduction theorem, $F^{SC}(Th(\bar{p},\bar{q});\bar{p}) \models F^{NC}(Th(\bar{p},\bar{q});\bar{p})$.

From another perspective,

- $F^{NC}(Th(\bar{p},\bar{q});\bar{p})$ is “the best” upper approximation of $Th(\bar{p},\bar{q})$ using the restricted vocabulary (as shown in Point 3 of Theorem 3.1);
- $F^{SC}(Th(\bar{p},\bar{q});\bar{p})$ is “the best” lower approximation of $Th(\bar{p},\bar{q})$ using the restricted vocabulary (as shown in Point 2 of Theorem 3.2).

3.5. The computational perspective for weak forgetting

The computational overhead involved in computing weak forgetting $F^{SC}()$ is, in the worst case, the same as for computing standard forgetting $F^{NC}()$, which is exponential in the size of the input formula, yet there are some pragmatic distinctions. Transformations of formulas made for $F^{NC}()$ can many times be reused in computing $F^{SC}()$. Moreover, frequently there may be a more substantial complexity gain. Notice that in applications, theories are typically presented as sets of formulas, $Th(\bar{p},\bar{q}) = \{A_1(\bar{p},\bar{q}),\dots,A_k(\bar{p},\bar{q})\}$, interpreted as the conjunction $Th(\bar{p},\bar{q}) \equiv A_1(\bar{p},\bar{q}) \wedge \dots \wedge A_k(\bar{p},\bar{q})$. By Theorem (7),

$$\begin{aligned} F^{SC}(Th(\bar{p},\bar{q});\bar{p}) \equiv \forall\bar{p}(Th(\bar{p},\bar{q})) \equiv\ & \forall\bar{p}(A_1(\bar{p},\bar{q}) \wedge \dots \wedge A_k(\bar{p},\bar{q})) \equiv \\ & \forall\bar{p}(A_1(\bar{p},\bar{q})) \wedge \dots \wedge \forall\bar{p}(A_k(\bar{p},\bar{q})). \end{aligned}$$

This partitioning allows one to eliminate second-order quantifiers $\forall\bar{p}$ in each conjunct separately. Though this has to be repeated k times, the formulas involved are smaller, which in typical cases results in a significant complexity reduction.

Notice also that frequently in applications, the formulas $A_1(\bar{p},\bar{q}),\dots, A_k(\bar{p},\bar{q})$ are syntactically structured in the form of rules/implications (or sequents) of the form:

$$(\ell_1 \wedge \dots \wedge \ell_r) \to (\ell_{r+1} \vee \dots \vee \ell_s), \tag{31}$$

where ℓ_i $(i = 1,\dots,s)$ are literals. Of course, (31) is equivalent to $\neg\ell_1 \vee \dots \vee \neg\ell_r \vee \ell_{r+1} \vee \dots \vee \ell_s$. Eliminating second-order quantifiers $\bar{p}$ from:

Algorithm 1: Eliminating second-order quantifiers from formulas of the form (32).

```
Data: Formula $A \equiv \forall\bar{p}\left(\neg\ell_1 \vee \ldots \vee \neg\ell_r \vee \ell_{r+1} \vee \ldots \vee \ell_s\right)$    (i.e., $A \equiv$ (32))
Result: Formula $B$ equivalent to $A$, without second-order quantifiers
begin
    set $B$ to become the formula obtained from $A$ after removing double negations;
    if there is a literal and its negation in $B$ then
        set $B$ to become $\mathbb{T}$
    else
        set $B$ to become the formula obtained from $B$ after removing literals involving propositional variables from $\bar{p}$, where the empty disjunction that
          may be obtained is $\mathbb{F}$
    end
    Remove from $B$ quantifiers binding propositional variables in $\bar{p}$
end
```

$$\forall\bar{p}\left(\neg\ell_1 \vee \ldots \vee \neg\ell_r \vee \ell_{r+1} \vee \ldots \vee \ell_s\right) \tag{32}$$

is then straightforward, as shown in Algorithm 1.

The following example illustrates this method.

Example 3.3. Consider the following formula:

$$\forall q\forall r\big(\neg\neg q \vee \neg r \vee \neg s \vee t\big). \tag{33}$$

After removing double negation in front of q, then removing the literals involving quantified variables (i.e., q and $\neg r$), and finally removing the quantifiers, the algorithm outputs $(\neg s \vee t)$ as a formula equivalent to (33). □

It is worth emphasizing that strong forgetting is more suitable for conjunctive theories, while weak forgetting is more suitable for disjunctive ones, in particular, clauses/rules. For example, when forgetting p from the conjunction $p \wedge A$ with A not containing p, we have $F^{NC}\big(p \wedge A; p\big) \equiv A$, and $F^{SC}\big(p \wedge A; p\big) \equiv \mathbb{F}$. Here $F^{SC}\big(p \wedge A; p\big)$ trivializes, while $F^{NC}\big(p \wedge A; p\big)$ remains potentially meaningful. Dually, when one forgets p from the disjunction $p \vee A$ the respective results are $F^{NC}\big(p \vee A; p\big) \equiv \mathbb{T}$, and $F^{SC}\big(p \vee A; p\big) \equiv A$. In this case $F^{NC}\big(p \wedge A; p\big)$ trivializes, while its dual $F^{SC}\big(p \vee A; p\big)$ does not.

4. Some examples

The first example shows that standard forgetting can be too strong in scenarios, where formulas (rules) depend on some uncontrolled parameters. That is, propositions whose truth values depend on the parameter's values, occur only in the rule's premises. In such cases, when a parameter's value becomes unknown, e.g., due to a failure of a measuring device, one would like to forget the corresponding proposition as this becomes meaningless. In standard forgetting, all rules involving such propositions disappear, reducing to $\mathbb{T}$.

Example 4.1. Consider a toy expert system for maintaining a proper balance between temperature and pressure in a production process. While the pressure can be controlled, we assume that outside temperature is not controllable. Let:

- mt, ht stand for "medium temperature" and "high temperature";
- lp, mp stand for "maintain low pressure" and "maintain medium pressure".

The considered theory, $Th(mt, ht, lp, mp)$, consists of the following formulas:

$$mt \rightarrow lp \vee mp; \tag{34}$$

$$ht \rightarrow lp. \tag{35}$$

Assume the temperature sensor is broken and one wants to forget related propositions mt, ht. Using Theorem 3.1(1) and Theorem 3.2(1), simple transformations of formulas, and Lemma 2.3, we obtain that:

- $F^{NC}\big(Th(mt, ht, lp, mp); mt, ht\big)$ is equivalent to $\mathbb{T}$.
- $F^{SC}\big(Th(mt, ht, lp, mp); mt, ht\big)$ is equivalent to $(lp \vee mp) \wedge lp$, which is equivalent to lp.

Notice that $F^{SC}\big(\big)$ is much more informative: when the temperature is unknown, for safety reasons it is better to maintain low pressure (lp) as a sufficient condition for satisfying rules (34), (35). □

The following example illustrates the use of forgetting when there are restrictions on accessing sensitive, personal data directly from a specific person or belief base. One gets around this by leveraging implicit information about that person through an inferential process that involves the use of forgetting.

Example 4.2. Assume Eve works in human resources in a company where Joe is employed. Eve faces some cultural barriers, or legislative restrictions, in asking Joe about his potential addiction to alcohol or drugs. In order to find the answer, she may use some extra knowledge/beliefs in addition to some neutral questions or observations in probing for an answer. To illustrate the approach consider a belief, expressed by the following simple formula, where fdd stands for “frequently denies driving”, ld stands for “likes driving”, and pa stands for “potentially addicted”:

$$fdd \to (\neg ld \vee pa). \tag{36}$$

When using (36) to detect Joe’s potential addiction to alcohol or drugs, Eve may wonder when this formula implies pa, without referring to pa itself. That is, she might be interested in the theory:

$$Th(pa, ld, fdd) \overset{\text{def}}{\equiv} \big(\underbrace{(fdd \to (\neg ld \vee pa))}_{(36)} \to pa\big),$$

but forgetting about pa. In this case, weak forgetting can be used:

$$F^{SC}\big(Th(pa, ld, fdd); pa\big). \tag{37}$$

According to Theorem 3.2(1), (37) is equivalent to:

$$\forall pa\big((fdd \to (\neg ld \vee pa)) \to pa\big), \tag{38}$$

which is equivalent to:

$$\neg\exists pa\big((fdd \to (\neg ld \vee pa)) \wedge \neg pa\big). \tag{39}$$

Formula (39) can easily be transformed into a form required for application of Lemma 2.3:

$$\neg\exists pa\big(\underbrace{(pa \to \mathbb{F})}_{\neg pa} \wedge (fdd \to (\neg ld \vee pa))\big).$$

Now Lemma 2.3(8) results in the following equivalent formula:

$$\neg\big(fdd \to (\neg ld \vee \mathbb{F})\big), \tag{40}$$

which is equivalent to $\neg\big(fdd \to \neg ld\big)$, and thus equivalent to $fdd \wedge ld$. To derive the conclusion that Joe is potentially addicted (pa), Eve might then try to find out independently, whether Joe frequently denies driving while at the same time likes driving. If both these assertions are true, then pa is true too.

Observe that $F^{NC}\big(Th(pa, ld, fdd); pa\big)$ is equivalent to: $\exists pa\big((fdd \to (\neg ld \vee pa)) \to pa\big)$, and thus equivalent to $\mathbb{T}$ (with witness $pa \overset{\text{def}}{\equiv} \mathbb{T}$). So, it could not be used in this particular context. □

The last example in this section is inspired by applying forgetting to the problem of restoring consistency of belief bases merged from different sources [24].

Example 4.3. Assume that Jack (the first source) considers building an outdoor recreational complex consisting of tennis courts (tc) and a swimming pool (sp). He allocates a budget for the investments. Let ‘bdg’ express that the overall investment fits in the budget, ‘$loan$’ stand for getting a loan and ‘inv’ denote involvement of an additional investor. Jack’s requirements are then expressed as:

$$tc \wedge sp \wedge (bdg \vee loan \vee inv). \tag{41}$$

He hires an external consultant (the second source) who believes that in addition to outdoor sport facilities, an indoor squash court (isq) and a gym center (gc) are necessary to attract enough customers. This can be expressed by $\underbrace{(tc \vee sp)}_{\text{outdoor facilities}} \to (isq \wedge gc)$. The consultant is well aware of specific costs for the different centers and courts. Consequently, the consultant believes that building an additional indoor squash court will not require a loan ($isq \to \neg loan$), while for a gym center a loan will be needed ($gc \to loan$). Summing up, the external consultant’s beliefs can be expressed as:

$$\big((tc \vee sp) \to (isq \wedge gc)\big) \wedge \big(isq \to \neg loan\big) \wedge \big(gc \to loan\big). \tag{42}$$

Jack’s beliefs, expressed by (41), merged with the consultant’s beliefs, which are expressed by (42), are jointly inconsistent in regard to ‘$loan$’. The investor decides to forget about ‘$loan$’. According to consistency restoring strategies considered in [24], in order to derive meaningful conclusions, Jack can forget about ‘$loan$’ in selected parts of the merged belief bases. Assume that he considers forgetting ‘$loan$’ in (41) or (42):

- $F^{NC}\big(\{(41)\}; loan\big) \equiv tc \wedge sp,$ (43)

$$F^{SC}\big(\{(41)\}; loan\big) \equiv tc \wedge sp \wedge (bdg \vee inv); \tag{44}$$

– $$F^{NC}\big(\{(42)\}; loan\big) \equiv \big((tc \vee sp) \to (isq \wedge gc)\big) \wedge \big(isq \to \neg gc\big), \tag{45}$$

$$F^{SC}\big(\{(42)\}; loan\big) \equiv \big((tc \vee sp) \to (isq \wedge gc)\big) \wedge \neg isq \wedge \neg gc. \tag{46}$$

Notice that weak forgetting $F^{SC}()$ adds substantial additional informative content in comparison to standard forgetting $F^{NC}()$, by providing sufficient conditions for the respective formulas:

- (44) extends (43) by information that to satisfy the formula (41) after forgetting about '$loan$', it suffices that additionally $(bdg \vee inv)$ holds;
- (46) extends (45) by information that to satisfy the formula (42) after forgetting about '$loan$', it suffices that additionally $\neg isq$ and $\neg gc$ hold. □

Although the examples considered may seem somewhat contrived due to their brevity, they do target deeper generic applications in which forgetting operators can play a substantial inferential role. Additionally, the pragmatic use and importance of the weak forgetting operator is clearly shown as a useful inferential complement to the standard forgetting operator.

5. Relationship with strongest necessary and weakest sufficient conditions

Strongest necessary and weakest sufficient conditions have been introduced in [25]. Let us recall the definitions with minor adjustments.

By *a necessary condition of a formula* $A(\bar{p}, \bar{q})$ *on propositional variables* $\bar{q}$ *under theory* $Th(\bar{p}, \bar{q})$ we shall understand any formula $B(\bar{q})$ containing only symbols in $\bar{q}$ such that $Th(\bar{p}, \bar{q}) \models A(\bar{p}, \bar{q}) \to B(\bar{q})$. Such a formula $B(\bar{q})$ is the *strongest necessary condition*, denoted by $snc\big(Th(\bar{p}, \bar{q}); A(\bar{p}, \bar{q}); \bar{q}\big)$ if, additionally, for any necessary condition $C(\bar{q})$ of $A(\bar{p}, \bar{q})$ on $\bar{q}$ under $Th(\bar{p}, \bar{q})$, we have that $T(\bar{p}, \bar{q}) \models B(\bar{q}) \to C(\bar{q})$.

By *a sufficient condition of a formula* $A(\bar{p}, \bar{q})$ *on propositional variables* $\bar{q}$ *under theory* $Th(\bar{p}, \bar{q})$ we shall understand any formula $B(\bar{q})$ containing only symbols in $\bar{q}$ such that $Th(\bar{p}, \bar{q}) \models B(\bar{q}) \to A(\bar{p}, \bar{q})$. Such a formula is the *weakest sufficient condition*, denoted by $wsc\big(Th(\bar{p}, \bar{q}); A(\bar{p}, \bar{q}); \bar{q}\big)$ if, additionally, for any sufficient condition $C(\bar{q})$ of $A(\bar{p}, \bar{q})$ on $\bar{q}$ under $Th(\bar{p}, \bar{q})$, we have that $Th(\bar{p}, \bar{q}) \models C(\bar{q}) \to B(\bar{q})$.

The following second-order characterization of $snc()$ and $wsc()$ has been provided in [13].

Lemma 5.1. *For any* $A(\bar{p}, \bar{q})$, $\bar{q}$ *and* $T(\bar{p}, \bar{q})$:

$$snc\big(Th(\bar{p}, \bar{q}); A(\bar{p}, \bar{q}); \bar{q}\big) \equiv \exists \bar{p}\big(Th(\bar{p}, \bar{q}) \wedge A(\bar{p}, \bar{q})\big); \tag{47}$$

$$wsc\big(Th(\bar{p}, \bar{q}); A(\bar{p}, \bar{q}); \bar{q}\big) \equiv \forall \bar{p}\big(Th(\bar{p}, \bar{q}) \to A(\bar{p}, \bar{q})\big). \quad \square \tag{48}$$

By use of Theorem 3.1(1), Theorem 3.2(1) and Lemma 5.1, the following corollary shows that the dual forgetting operators and strongest necessary and weakest sufficient conditions are mutually definable.

Corollary 5.2.

$$snc\big(Th(\bar{p}, \bar{q}); A(\bar{p}, \bar{q}); \bar{q}\big) \equiv F^{NC}\big(Th(\bar{p}, \bar{q}) \wedge A(\bar{p}, \bar{q}); \bar{p}\big); \tag{49}$$

$$wsc\big(Th(\bar{p}, \bar{q}); A(\bar{p}, \bar{q}); \bar{q}\big) \equiv F^{SC}\big(Th(\bar{p}, \bar{q}) \to A(\bar{p}, \bar{q}); \bar{p}\big); \tag{50}$$

$$F^{NC}\big(Th(\bar{p}, \bar{q}); \bar{p}\big) \equiv snc\big(Th(\bar{p}, \bar{q}); \mathbb{T}; \bar{q}\big); \tag{51}$$

$$F^{SC}\big(Th(\bar{p}, \bar{q}); \bar{p}\big) \equiv wsc\big(\neg Th(\bar{p}, \bar{q}); \mathbb{F}; \bar{q}\big). \quad \square \tag{52}$$

Even though Corollary 5.2 holds, notice that forgetting operators typically serve purposes different than those of strongest necessary and weakest sufficient conditions. That is, the operators $snc()$ and $wsc()$ focus on queries $A(\bar{p}, \bar{q})$ to the belief base expressed by theory $Th(\bar{p}, \bar{q})$, when the underlying language is reduced. In applying $F^{NC}()$ and $F^{SC}()$, one looks for the "best" approximations of the theory in a reduced sublanguage. For example,

- the weakest sufficient condition $wsc\big(Th(\bar{p}, \bar{q}); A(\bar{p}, \bar{q}); \bar{q}\big)$ can be used to *explain the query (observation),* $A(\bar{p}, \bar{q})$, given the theory $Th(\bar{p}, \bar{q})$, in a sublanguage consisting of $\bar{q}$;
- on the other hand, the associated weak forgetting operator $F^{SC}\big(Th(\bar{p}, \bar{q}); \bar{p}\big)$, can be used to *explain the theory* $Th(\bar{p}, \bar{q})$ in the same sublanguage, consisting of $\bar{q}$.

As an added distinction, computing $F^{SC}\big(Th(\bar{p}, \bar{q}); \bar{p}\big)$ is usually more efficient than computing $wsc\big(Th(\bar{p}, \bar{q}); A(\bar{p}, \bar{q}); \bar{q}\big)$ (see also Sections 3.5 and 6.4).

6. A first-order extension

Generalization of forgetting operators to the first-order case is an important topic of research and also essential for many applications [9,14,26]. Although limited work has been done with standard forgetting, there are no general, generic approaches in this respect. In approaching this topic, classical first- and second-order logics [7,19] will be used, assuming the following languages:

- classical first-order logic, $\mathcal{L}_1$, extending propositional logic $\mathcal{L}_0$ with the set $\mathcal{V}_1$ of *individual variables* representing domain objects, $\mathcal{R}$ of relation symbols, and first-order quantifiers $\forall, \exists$ binding individual variables;
- second-order logic, $\mathcal{L}_2$, extending $\mathcal{L}_1$ with *relational variables*, $\mathcal{V}_2$, and second-order quantifiers, also denoted by $\forall, \exists$, but binding relational variables.

Some additional terminology is also required. For a formula containing $Qx\big(A(x)\big)$, where $Q \in \{\exists, \forall\}$, by the *scope* of the quantifier Qx we mean the formula $A(x)$. The quantifier Qx *binds* the variable x in its scope. An occurrence of a variable x in a formula is *bound* if x occurs in the scope of a quantifier Qx. Otherwise the occurrence of this variable is *free*. A formula not containing free variables is called *closed*.

6.1. First-order standard and weak forgetting

When shifting from entailment (6) to implication (7), the deduction theorem for propositional logic was applied. In first-order logic, the deduction theorem requires that the formulas moved from the left-hand side to the right-hand side of $\models$, are closed. That is, in the first-order case there will be the requirement that the theories considered contain only closed formulas. In practice, this is not really a restriction. Typically, belief bases are defined using closed formulas. Even if the formulas in question contain free variables, such as in rule-based theories, they are assumed to be implicitly universally quantified.

Given the above requirement pertaining to closed theories, considerations about standard and weak forgetting provided in Section 3, including Theorems 3.1 and 3.2, are preserved. This is formulated in the following theorems, where for tuples of relation symbols $\bar{r}, \bar{s}$, $Th(\bar{r}, \bar{s})$ is a closed first-order theory over vocabulary $\bar{r}, \bar{s}$ and, rather than in the propositional version (12), (21), there is the requirement that for every second-order variable P representing first-order formulas over a vocabulary disjoint with $\bar{r}$:

- $F^{NC}\big(Th(\bar{r}, \bar{s}); \bar{r}\big)$ preserves the entailment of necessary conditions:
$$\forall P\Big(\forall \bar{r}\big(Th(\bar{r}, \bar{s}) \rightarrow P\big) \equiv \big(F^{NC}\big(Th(\bar{r}, \bar{s}); \bar{r}\big) \rightarrow P\big)\Big); \tag{53}$$
- $F^{SC}\big(Th(\bar{p}, \bar{q}); \bar{p}\big)$ preserves the entailment by sufficient conditions:
$$\forall P\Big(\big(P \rightarrow F^{SC}\big(Th(\bar{r}, \bar{s}); \bar{r}\big)\big) \equiv \forall \bar{r}\big(P \rightarrow Th(\bar{r}, \bar{s})\big)\Big). \tag{54}$$

Theorem 6.1. *For arbitrary tuples of relation symbols $\bar{r}, \bar{s}$ and a closed first-order theory $Th(\bar{r}, \bar{s})$,*

1. $F^{NC}\big(Th(\bar{r}, \bar{s}); \bar{r}\big) \equiv \exists \bar{r}\,\big(Th(\bar{r}, \bar{s})\big)$.
2. $F^{NC}\big(Th(\bar{r}, \bar{s}); \bar{r}\big) \equiv$*forget*$(Th(\bar{r}, \bar{s}); \bar{r})$.
3. $F^{NC}\big(Th(\bar{r}, \bar{s}), \bar{r}\big)$ *is the strongest (wrt →) formula over vocabulary $\bar{s}$, satisfying* (53). □

Notice that similarly to the case of Theorem 3.1.2, Theorem 6.1.2 follows from the first statement in the above theorem together with Theorem 8 of [26].

Theorem 6.2. *For arbitrary tuples of relation symbols $\bar{r}, \bar{s}$ and a closed first-order theory $Th(\bar{r}, \bar{s})$,*

1. $F^{SC}\big(Th(\bar{r}, \bar{s}); \bar{r}\big) \equiv \forall \bar{r}\big(Th(\bar{r}, \bar{s})\big)$.
2. $F^{SC}\big(Th(\bar{r}, \bar{s}); \bar{r}\big)$ *is the weakest (wrt →) formula over vocabulary $\bar{s}$, satisfying* (54). □

6.2. First-order Ackermann lemma

As in the propositional case, the first-order Ackermann Lemma provides a powerful technique for eliminating second-order quantifiers from standard and weak forgetting formulas. The DLS algorithm of [11] uses Lemma 6.3, described below.

In order to formulate the first-order version of the Ackermann lemma, let us extend the notation $A(p = expr)$ form Section 2. If A is a formula, r is a k-argument relation symbol occurring in A, $expr(x_1, \ldots, x_k)$ is an expression including variables among $x_1, \ldots, x_k$, then:

$$A\big(r(x_1, \ldots, x_k) = expr(x_1, \ldots, x_k)\big)$$

denotes a formula obtained from A by substituting all occurrences of $r(\ldots)$ by $expr(x_1, \ldots, x_k)$ where variables $x_1, \ldots, x_k$ are replaced by arguments of occurrences of $r(\ldots)$ being substituted. That is, $r(x_1, \ldots, x_k) = expr(x_1, \ldots, x_k)$ can be understood as a definition of

$r(x_1, \ldots, x_k)$, so whenever $r(\ldots)$ occurs in A, arguments of the expression $expr(\ldots)$ substituting this occurrence of $r(\ldots)$ should match the actual arguments of $r(\ldots)$.

For example, let $A \overset{\text{def}}{=} \big(s(x_1, a) \vee r(a, b) \vee r(b, c)\big)$ and $expr(x_1, x_2) \overset{\text{def}}{=} s(x_1, x_2) \wedge t(x_2, d)$. When $r(x_1, x_2) = expr(x_1, x_2)$, we have:

$$r(a, b) = expr(a, b) = s(a, b) \wedge t(b, d) \quad \text{and} \quad r(b, c) = expr(b, c) = s(b, c) \wedge t(c, d).$$

Therefore we have:

$$A\big(r(x_1, x_2) = \underbrace{(s(x_1, x_2) \wedge t(x_2, d))}_{expr(x_1, x_2)}\big) = s(x_1, a) \vee \underbrace{(s(a, b) \wedge t(b, d))}_{r(a,b)=expr(a,b)} \vee \underbrace{(s(b, c) \wedge t(c, d))}_{r(b,c)=expr(b,c)}$$

We can now formulate the first-order version of the Ackermann lemma.

Lemma 6.3 *(First-order Ackermann lemma). Let r be a k-ary relation symbol, A be a first-order formula without occurrences of r, B be a first-order formula and $\bar{x}$ be a k-tuple of distinct variables. Then:*

- *if B is positive wrt r then:* $\exists r\big(\forall \bar{x}(r(\bar{x}) \to A(\bar{x})) \wedge B\big) \;\equiv\; B(r(\bar{x}) = A(\bar{x}))$; (55)
- *if B is negative wrt r then:* $\exists r\big(\forall \bar{x}(A(\bar{x}) \to r(\bar{x})) \wedge B\big) \;\equiv\; B(r(\bar{x}) = A(\bar{x}))$. □ (56)

Observe that Fig. 1 also illustrates Lemma 6.3, where B grows when r grows (respectively shrinks).[7] In the first-order case, Fig. 1 is even more intuitive since growing and shrinking pertains to the tuples that satisfy the relation r (p in the figure), which are maximized (minimized) relative to (55) and (56), respectively.

To transform a formula into a form required in (55) or (56), one can again use the DLS algorithm [11]. Unlike the propositional case, in the first-order case such a transformation is not always doable in general, but it has been shown to work for large classes of formulas [6,11,19].

The following example illustrates the use of Lemma 6.3 in the context of standard and weak forgetting.

Example 6.4. To illustrate the use of the first-order Ackermann lemma, consider the following belief base, where $ms(x)$ stands for "person x has mild symptoms of a disease", $ss(x)$ – for "person x has severe symptoms of the disease", $h(x)$ – for "x should stay home", $t(x)$ – for "x needs a test for the disease", and $ich(x)$ – for "x should immediately consult a health care provider":

$$Th(ms, h, t, ss, ich) = \Big\{\forall x\Big(ms(x) \to \big(h(x) \wedge t(x)\big)\Big), \forall x\Big(\big(ss(x) \vee t(x)\big) \to ich(x)\Big)\Big\}. \tag{57}$$

When a test is not available, it is useful to forget about it, so one can consider standard and weak forgetting operators $F^{NC}\big(Th(ms, h, t, ss, ich); t)\big)$ and $F^{SC}\big(Th(ms, h, t, ss, ich); t)\big)$:

-
$$\begin{aligned} F^{NC}\big(Th\big(ms, h, t, ss, ich\big); t)\big) \equiv \exists t\Big(&\forall x\Big(ms(x) \to \big(h(x) \wedge t(x)\big)\Big) \wedge \\ &\forall x\Big(\big(ss(x) \vee t(x)\big) \to ich(x)\Big)\Big). \end{aligned} \tag{58}$$

  In order to eliminate $\exists t$ from (58) using Lemma 6.3, it suffices to transform it to the equivalent form[8]:

$$\exists t\Big(\forall x\Big(ms(x) \to t(x)\Big) \wedge \forall x\Big(ms(x) \to h(x)\Big) \wedge \forall x\Big(\big(ss(x) \vee t(x)\big) \to ich(x)\Big)\Big). \tag{59}$$

  An application of Lemma 6.3(56) results in:

$$\forall x\Big(ms(x) \to h(x)\Big) \wedge \forall x\Big(\big(ss(x) \vee ms(x)\big) \to ich(x)\Big). \tag{60}$$

  That is, a person with minor symptoms should stay at home. If, under the circumstances, it is not possible to test for the disease, then the severe *or* mild symptoms should suffice to immediately consult (by phone or a visit) the health care provider. Intuitively, both cases are correct:
  - by the second formula of the belief base (57), severe symptoms suffice for a need to immediately consult a health care provider;
  - by the first formula of the belief base (57), mild symptoms imply a need for making a test which, in turn, by the second formula of the belief base, suffices for a need to immediately consult a health care provider.

$$\begin{aligned} F^{SC}\big(Th\big(ms, h, t, ss, ich\big); t)\big) \equiv \forall t\Big(&\forall x\Big(ms(x) \to \big(h(x) \wedge t(x)\big)\Big) \wedge \\ &\forall x\Big(\big(ss(x) \vee t(x)\big) \to ich(x)\Big)\Big). \end{aligned} \tag{61}$$

[7] In Fig. 1, r is represented by p.

[8] Notice that in this case, the form can be automatically obtained using the DLS algorithm.

After a transformation similar to (59), and distributing $\forall t$ over conjunctions, one obtains the following equivalent formula:

$$\forall t \forall x\Big(ms(x) \to t(x)\Big) \wedge \forall t \forall x\Big(ms(x) \to h(x)\Big) \wedge \forall t \forall x\Big(\big(ss(x) \vee t(x)\big) \to ich(x)\Big). \tag{62}$$

One can now eliminate $\forall t$ in each conjunct separately:

- – $\forall t \forall x\Big(ms(x) \to t(x)\Big) \equiv \forall x \neg \exists t\Big(ms(x) \wedge \neg t(x)\Big) \equiv \forall x \neg \exists t\Big(\forall x\big(\mathbb{F} \to t(x)\big) \wedge ms(x) \wedge \neg t(x)\Big)$.
  Notice that subformula $\forall x\big(\mathbb{F} \to t(x)\big)$ is added artificially to make the Ackermann Lemma work. It is actually a tautology so, in conjunction with the other formulas, it does not affect their truth value. Now an application of Lemma 6.3(56), results in $\forall x \neg\Big(ms(x) \wedge \neg \mathbb{F}\Big)$, which is equivalent to $\forall x\Big(\neg ms(x)\Big)$;
- – in $\forall t \forall x\Big(ms(x) \to h(x)\Big)$ the quantifier $\forall t$ is redundant, so can simply be removed;
- – $\forall t \forall x\Big(\big(ss(x) \vee t(x)\big) \to ich(x)\Big) \equiv \forall x \neg \exists t\Big(\big(ss(x) \vee t(x)\big) \wedge \neg ich(x)\Big)$. We add an artificial conjunct $\forall x\Big(t(x) \to \mathbb{T}\Big)$ which is a tautology, apply Lemma 6.3(55), and obtain an equivalent formula $\forall x \neg\Big(\big(ss(x) \vee \mathbb{T}\big) \wedge \neg ich(x)\Big)$. This is equivalent to $\forall x\Big(ich(x)\Big)$.

Thus, we obtain the following first-order result, equivalent to (62):

$$\forall x\Big(\neg ms(x)\Big) \wedge \forall x\Big(ms(x) \to h(x)\Big) \wedge \forall x\Big(ich(x)\Big), \tag{63}$$

which is equivalent to:

$$\forall x\Big(\neg ms(x)\Big) \wedge \forall x\Big(ich(x)\Big). \tag{64}$$

Indeed, without access to a test, $t(x)$, one can only make sure that the first formula of (57) is true by assuring that $\forall x\Big(\neg ms(x)\Big)$ is true. The second formula of (57) can only be guaranteed when $\forall x\Big(ich(x)\Big)$ is true.

Though (64) looks rather useless, it can actually be used in querying and reasoning. The following points illustrate the ideas:

- (64) can be equivalently transformed to $\forall x\Big(\neg ms(x) \wedge ich(x)\Big)$, and used to select persons satisfying $\neg ms(x) \wedge ich(x)$, thus also satisfying (57). That is, when one reduces the domain to objects satisfying $\big(\neg ms(x) \wedge ich(x)\big)$, the initial theory is guaranteed to hold;
- given other formulas in the underlying belief base, one can make use of (64) in further reasoning. For example, assuming that severe symptoms imply mild symptoms, $\forall x\big(ss(x) \to ms(x)\big)$, from $\forall x\Big(\neg ms(x)\Big)$ one can infer $\forall x\Big(\neg ss(x)\Big)$, and so on. □

The use of $F^{SC}()$ in Example 6.4 has interesting generic potential in that it allows one to isolate a subdomain of individuals ensuring the validity of a restricted part of the original theory in question.

6.3. The fixpoint lemma

Notice that Lemma 6.3 requires that formula A does not contain occurrences of the eliminated relation symbol. On the other hand, when it does, one may still obtain useful results for a large class of formulas. To see this, consider once again the cases shown in Fig. 1, where rather than A we consider $A(p)$. As is well known (see, e.g., [1]), given that $A(p)$ is positive wrt p, there exists a least and greatest fixpoint of $A(p)$ wrt p, where by a *fixpoint of* $A(p)$ *wrt* p we understand any formula B satisfying $\models B \equiv A(B)$.

The above observation is all that is required to generalize the first-order extension. The idea is formulated in the following lemma, proved in [28] (see also, e.g., [13,19]), where:

- $\mathrm{Lfp}\, p.\big(A(p)\big)$ denotes the *least* (wrt implication) fixpoint of $A(p)$;
- $\mathrm{Gfp}\, p.\big(A(p)\big)$ denotes the *greatest* (wrt implication) fixpoint of $A(p)$.

Intuitively, when the extension of a formula *shrinks*, it becomes *stronger* because it implies more formulas. Likewise, when the extension of a formula *grows*, it becomes *weaker* because it is implied by more formulas. Therefore, *least wrt implication* means *strongest* and *greatest wrt implication* means *weakest*. Thus, $\mathrm{Lfp}\, p.\big(A(p)\big)$ is the strongest formula being a fixpoint of $A(p)$, and $\mathrm{Gfp}\, p.\big(A(p)\big)$ is the weakest formula being a fixpoint of $A(p)$.

Lemma 6.5 *(Fixpoint lemma). Let r be a k-ary relation symbol, $A(r)$ be a first-order formula with positive occurrences of r only, B be a first-order formula, and $\bar{x}$ be a k-tuple of distinct variables. Then:*

– *if B is positive wrt r then:* $$\exists r\big(\forall \bar{x}(r(\bar{x}) \to A(r)) \wedge B\big) \;\equiv\; B\big(r(\bar{x}) = \mathrm{Gfp}\, r(\bar{x}).\big(A(r)\big)\big); \tag{65}$$

– *if B is negative wrt r then:* $$\exists r\big(\forall \bar{x}(A(r) \to r(\bar{x})) \wedge B\big) \;\equiv\; B\big(r(\bar{x}) = \mathrm{Lfp}\, r(\bar{x}).\big(A(r)\big)\big). \quad \square \tag{66}$$

Notice that Fig. 1 applies also to the fixpoint case formulated in Lemma 6.5, where A grows when r grows (respectively shrinks).[9] Therefore we look for the greatest (respectively least) r such that $r \equiv A(r)$, i.e., r is a (greatest or least) fixpoint of $A(r)$.

The following example illustrates an application of this lemma.

Example 6.6. A communication network is being designed. Due to a specific application area, the designers have to consider special security requirements. In particular they consider two networks: an internal and an external one. The internal network nodes should not be externally reachable unless they are protected by a specialized expensive security component. A part of the underlying belief base contains, among others, the following formula characterizing reachability, where $con(x, y)$ stands for "nodes x and y are directly connected", and $r(x, y)$ stands for "y is reachable from x"[10]:

$$\forall x \forall y \Big(\big(con(x, y) \vee \exists z \big(con(x, z) \wedge r(z, y) \big) \big) \rightarrow r(x, y) \Big). \tag{67}$$

Due to the security requirements, the following integrity constraint has to be preserved when choosing direct network connections, where $ex(x)$ and $in(x)$ denote that node x belongs to the external or internal network, respectively, and $sec(x)$ denotes that node x is equipped with the security component:

$$\forall y \Big(\exists x \big(ex(x) \wedge r(x, y) \big) \rightarrow \big(in(y) \rightarrow sec(y) \big) \Big). \tag{68}$$

That is, when y is reachable from a node x of an external network ($\exists x \big(ex(x) \wedge r(x, y) \big)$) then whenever y is an internal node ($in(y)$) then it is to be equipped with the security protecting component ($sec(y)$). To focus on the design of $con()$, the designers prefer to abstract from $r()$ for the time being, so forget about r.

Let us first compute $F^{NC}\big(Th(con, r, ex, in, sec); r\big)$, where $Th(con, r, ex, in, sec)$ denotes the conjunction $(67) \wedge (68)$:

$$F^{NC}\big(Th(con, r, ex, in, sec); r\big) \equiv$$

$$\exists r \Big(\forall x \forall y \Big(\big(con(x, y) \vee \exists z \big(con(x, z) \wedge r(z, y) \big) \big) \rightarrow r(x, y) \Big) \wedge \tag{69}$$

$$\forall y \Big(\exists x \big(ex(x) \wedge r(x, y) \big) \rightarrow \big(in(y) \rightarrow sec(y) \big) \Big) \Big). \tag{70}$$

Observe that Lemma 6.3 cannot be applied due to (69). However, Lemma 6.5(66) can still be used and results in:

$$\begin{aligned} &\forall y \Big(\exists x \big(ex(x) \wedge \mathrm{Lfp}\, r(x, y).\big(con(x, y) \vee \exists z \big(con(x, z) \wedge r(z, y) \big) \big) \big) \\ &\qquad\qquad\qquad\qquad \rightarrow \big(in(y) \rightarrow sec(y) \big) \Big). \end{aligned} \tag{71}$$

The least fixpoint in (71) actually defines $r()$ as the transitive closure of $con()$.

Using the well-known Knaster and Tarski theorem, the least fixpoint $\mathrm{Lfp}\, r(x, y).\big(A(r)\big)$, where $A(r) \overset{\text{def}}{\equiv} con(x, y) \vee \exists z \big(con(x, z) \wedge r(z, y) \big)$, is equivalent to a disjunction $\bigvee_i A^i(\mathbb{F})$, where A^i stands for applying A i times. Therefore, the formula (71) can be represented by the conjunction:

$$\begin{array}{l} \forall y \Big(\\ \qquad \exists x \big(ex(x) \wedge A^0(\mathbb{F}) \big) \rightarrow \big(in(y) \rightarrow sec(y) \big) \wedge \\ \qquad \exists x \big(ex(x) \wedge A^1(\mathbb{F}) \big) \rightarrow \big(in(y) \rightarrow sec(y) \big) \wedge \\ \qquad \ldots \\ \qquad \exists x \big(ex(x) \wedge A^i(\mathbb{F}) \big) \rightarrow \big(in(y) \rightarrow sec(y) \big) \wedge \\ \qquad \ldots \\ \quad \Big). \end{array}$$

To compute $F^{SC}\big(Th(con, r, ex, in, sec); r\big)$, we consider:

$$F^{SC}\big(Th(con, r, ex, in, sec); r\big) \equiv$$

$$\forall r \Big(\forall x \forall y \Big(\big(con(x, y) \vee \exists z \big(con(x, z) \wedge r(z, y) \big) \big) \rightarrow r(x, y) \Big) \wedge \tag{72}$$

$$\forall y \Big(\exists x \big(ex(x) \wedge r(x, y) \big) \rightarrow \big(in(y) \rightarrow sec(y) \big) \Big) \Big). \tag{73}$$

The quantifier $\forall r$ can be distributed over the conjunction, so one can consider:

$$\forall r \forall x \forall y \Big(\big(con(x, y) \vee \exists z \big(con(x, z) \wedge r(z, y) \big) \big) \rightarrow r(x, y) \Big) \wedge \tag{74}$$

$$\forall r \forall y \Big(\exists x \big(ex(x) \wedge r(x, y) \big) \rightarrow \big(in(y) \rightarrow sec(y) \big) \Big). \tag{75}$$

[9] As in the first-order case, in Fig. 1 r is represented by p.

[10] Notice that formula (67) can be obtained as a translation of a rule defining $r()$ as a transitive closure of $con()$.

To eliminate $\forall r$ from (74), we transform it to the following equivalent form:

$$\neg\exists x\exists y\exists r\Big(\big(con(x,y)\vee\exists z\big(con(x,z)\wedge r(z,y)\big)\big)\wedge\neg r(x,y)\Big),\tag{76}$$

equivalent to:

$$\neg\exists x\exists y\exists r\Big(\underbrace{\forall u\forall w\big(r(u,w)\to(u\neq x\vee w\neq y))\big)}_{\neg r(x,y)}\wedge\big(con(x,y)\vee\exists z\big(con(x,z)\wedge r(z,y)\big)\big)\Big).\tag{77}$$

Applying Ackermann's Lemma 6.3(55) to (77) one obtains:

$$\neg\exists x\exists y\Big(\big(con(x,y)\vee\exists z\big(con(x,z)\wedge(z\neq x\vee y\neq y)\big)\big)\Big),\tag{78}$$

equivalent to:

$$\forall x\forall y\Big(\neg con(x,y)\wedge\forall z\big(con(x,z)\to z=x\big)\Big),\tag{79}$$

which in turn is equivalent to:

$$\forall x\forall z\Big(\neg con(x,z)\Big).\tag{80}$$

To eliminate $\forall r$ from (75), we transform it to the following equivalent form:

$$\neg\exists y\exists r\Big(\exists x\big(ex(x)\wedge r(x,y)\big)\wedge in(y)\wedge\neg sec(y)\Big).\tag{81}$$

Adding an artificial conjunct $\forall x\forall y\big(r(x,y)\to\mathbb{T}\big)$, equivalent to $\mathbb{T}$ (to ensure the right syntactic structure), and applying Ackermann's Lemma 6.3(55) to (81), we obtain:

$$\forall y\Big(\exists x\big(ex(x)\big)\to\big(in(y)\to sec(y)\big)\Big).\tag{82}$$

Combining (80) and (82) we obtain the following:

$$F^{SC}\big(Th(con,r,ex,in,sec);r\big)\equiv\forall x\forall z\Big(\neg con(x,z)\Big)\wedge\forall y\Big(\exists x\big(ex(x)\big)\to\big(in(y)\to sec(y)\big)\Big).$$

That is, to make sure that $Th(con,r,ex,in,sec)$ remains true when $r()$ is forgotten, nodes cannot be connected to any other nodes (including themselves), and, if there is an external node, then every internal node is to be equipped with the security component. Though this guarantees that the initial security requirements are satisfied, the resulting theory is rather strong. On the other hand, it is the weakest theory on the reduced language that guarantees the requirements. This indicates that the reachability predicate $r()$ is vital from the point of view of reasoning about sufficient conditions, and should not, in this case, be forgotten from the initial specification.

Notice that $F^{NC}\big(Th(con,r,ex,in,sec);r\big)$ additionally provides a useful definition of reachability, expressed as a least fixpoint formula. □

6.4. Computational aspects

Forgetting is typically applied to finite domain knowledge or belief bases and rule languages. As indicated by the first points of Theorems 3.1 and 3.2, computing propositional queries expressed by $F^{NC}()$ is NP-complete, and those expressed by $F^{SC}()$ is co-NP-complete. On the other hand, the data complexity of first-order queries obtained using Lemma 6.3 is PTime and LogSpace [1]. Data complexity of fixpoint queries, thus also queries obtained from Lemma 6.5, is PTime [1]. Therefore, the approach based on Ackermann's Lemma and its fixpoint extension is computationally friendly when reasoning-by-querying is concerned.

The following approaches to second-order quantifier elimination that have previously been formulated and implemented are:

- using Lemma 6.3 as a basis for the DLS algorithm [11], which first attempts to transform an arbitrary formula into a form suitable for applying this lemma, and then uses this lemma as a basis for eliminating quantifiers;
- using Lemma 6.5 as a basis for the DLS* algorithm [12], which extends the DLS algorithm to work with formulas that are suitable for an application of the fixpoint lemma.

In fact, all calculations carried out in this paper that involve elimination of second-order quantifiers reflect selected steps used in the DLS or DLS* algorithms.

As has been shown in Section 3.5, computing propositional equivalents for the weak forgetting operator is often more efficient than computing such equivalents for standard forgetting and for weakest sufficient and strongest necessary conditions. This reasoning can also be extended to the first-order case as follows, where we consider eliminating the quantifier $\forall r$ from the formula:

$$\forall r\forall\bar{x}\Big(r(\bar{x}_1)\vee\ldots\vee r(\bar{x}_m)\vee\neg r(\bar{x}_{m+1})\vee\ldots\vee\neg r(\bar{x}_n)\vee A(\bar{z})\Big),\tag{83}$$

where $0 \le m \le n$, $\bar{x}$ contains at least all variables in $\bar{x}_1, \dots, \bar{x}_n$, and formula A does not contain relation symbol r. Formula (83) is equivalent to:

$$\forall \bar{x} \neg \exists r \Big(\neg r(\bar{x}_1) \wedge \dots \wedge \neg r(\bar{x}_m) \wedge r(\bar{x}_{m+1}) \wedge \dots \wedge r(\bar{x}_n) \wedge \neg A(\bar{z}) \Big), \tag{84}$$

which can be transformed into the equivalent form (e.g., using the Dls algorithm):

$$\forall \bar{x} \neg \exists r \Big(\forall \bar{y} \big(r(\bar{y}) \to (\bar{y} \neq \bar{x}_1 \wedge \dots \wedge \bar{y} \neq \bar{x}_m) \big) \wedge r(\bar{x}_{m+1}) \wedge \dots \wedge r(\bar{x}_n) \wedge \neg A(\bar{z}) \Big). \tag{85}$$

After applying Ackermann's Lemma (6.3) we obtain the following formula equivalent to (85), thus also equivalent to (83):

$$\forall \bar{x} \neg \Big((\bar{x}_{m+1} \neq \bar{x}_1 \wedge \dots \wedge \bar{x}_{m+1} \neq \bar{x}_m) \wedge \dots \wedge (\bar{x}_n \neq \bar{x}_1 \wedge \dots \wedge \bar{x}_n \neq \bar{x}_m) \wedge \neg A(\bar{z}) \Big). \tag{86}$$

Formula (86) can be simplified to:

$$\forall \bar{x} \Big(\bar{x}_{m+1} = \bar{x}_1 \vee \dots \vee \bar{x}_{m+1} = \bar{x}_m \vee \dots \vee \bar{x}_n = \bar{x}_1 \vee \dots \vee \bar{x}_n = \bar{x}_m \vee A(\bar{z}) \Big). \tag{87}$$

That is, the length of the resulting formula (87) is at most quadratic in the length of the input formula (83). Given variables $\bar{x}_1, \dots, \bar{x}_m, \bar{x}_{m+1}, \dots, \bar{x}_n$, one typically can still substantially simplify the result. Notice that in similar cases, the fixpoint lemma is not needed (but may be needed for other shapes of formulas).

When dealing with a conjunction of formulas of the form (83), the quantifier $\forall r$ can be distributed over conjunction (as, e.g., in (62)), and each conjunct can be processed separately. This provides a sharp contrast in comparison with computing $F^{NC}()$, $wsc()$ and $snc()$, where such a distribution is generally not possible. This is indicated by their syntactic second-order characterizations given by point 1 of Theorem 3.1, and formulas (47), (48) of Lemma 5.1, respectively.

It is worth emphasizing that the problem as to whether a second-order formula is equivalent to a first-order (or fixpoint) formula, is highly undecidable. So algorithms like Dls, Dls* as well as any other algorithms attempting to do the same quantifier elimination, have to fail for some input formulas. However, pragmatic application has shown that these algorithms work for large classes of formulas (see Section 7 for a more detailed discussion).

7. Related work

Standard forgetting, *forget*(), has been introduced in the foundational paper [26], where model theoretical definitions and analysis of properties of the standard *forget*() operator are provided. Its second-order characterization as well as its entailment preserving property, as quoted in Theorem 2.2 above, are consequences of this definition. The paper [26] opened a research subarea, summarized, e.g., in [10] or more recently, in [14]. In the approach used in the current paper, one begins with an entailment-based, inferential perspective, as expressed by (10) and (11). This is directly beneficial for pragmatic application, since this approach leads directly to algorithms and implementations of the dual forgetting operators. In addition to introducing a new weak forgetting operator, standard forgetting is also characterized in this context.

The paper [10] concentrates on *introspective* forgetting ("forgetting as becoming ignorant"). It extends modal epistemic logic with modal operators allowing one to express what is known before and what remains known after forgetting. It therefore essentially deals with necessary conditions, corresponding to standard forgetting.

In [14] two types of forgetting are distinguished: (1) forgetting part of the signature, corresponding to standard forgetting, and (2) forgetting a formula, related to *contracting* in the AGM theory of belief change [3,18]. In the current paper, both dual operators, weak and strong forgetting, belong to the first type of forgetting. The extension of this work to forgetting formulas is an interesting research direction.

A general framework, covering a range of belief changes involving forgetting, is proposed in [5]. In particular, a commonsense perspective based on belief change, involving contraction, ignorance introduction, revision/update, (deductive) abstraction at the level of rules, marginalization, etc., is proposed. Another general framework for standard-like forgetting is investigated in [9], where forgetting is regarded as a belief change operator, independent of the underlying logic. Forgetting is achieved by reducing a part of a signature in a theory. In that approach, forgetting is specified by the set of logical consequences of the input theory over the reduced language. Though providing very interesting characterizations and contexts of forgetting, the frameworks of [5] and [9] deal with consequences (necessary conditions) of the considered theories and do not deal with sufficient conditions, expressed by the weak forgetting operator in this paper.

Papers which consider relations between forgetting and weakest sufficient and strongest necessary conditions include [16,17]. In the first of these papers, forgetting is applied in the context of strongest necessary and weakest sufficient conditions in Computation Tree Logic (CTL). Axiomatic characterization of forgetting and an algorithm for computing forgetting are provided. In addition, weakest sufficient and strongest necessary conditions are characterized using forgetting (Theorem 9 in [16]). The characterization is similar in spirit to the Corollaries (49) and (50) in the current paper. In [17] the results of [16] are transferred into the context of μ-calculus. Both in CTL and μ-calculus, typical reasoning problems are intractable. Forgetting in multiagent modal logics, sharing a similar methodology, has been addressed in [15]. Unlike in [15–17], in the current paper, both standard and weak forgetting are considered. The approach isolates broad classes of formulas for which one can compute first-order or fixpoint equivalents of both forgetting operators. This in turn, leads to a tractable reasoning-by-querying machinery.

In [32] a *weak* forgetting operator has been introduced which differs from the weak forgetting operator, $F^{SC}()$, used in this paper, and it has characteristics much more related to the standard forgetting operator, $F^{NC}()$. While standard forgetting is not always first-order expressible, the weak operator of [32] has been developed to make sure that the result of forgetting can always be expressed in first-order logic. Weak forgetting in the sense of [32] and standard forgetting differ only in the cases where the result of standard forgetting is not first-order expressible. This operator does not coincide with the dual weak forgetting operator $F^{SC}()$. The terminology used in the current paper, reflects the fact that $F^{SC}()$ and $F^{NC}()$ are dual operators that formally show the relationship of $F^{SC}()$ to weakest sufficient conditions and $F^{NC}()$ to strongest necessary conditions.

A series of papers [8,33,34] (see also references there) has addressed forgetting in description logics. Though description logics are strongly related to modal logics [4], the papers [8,33,34] are methodologically closer to the approach used in this paper. In [8], forgetting-based abduction is investigated using weakest sufficient and strongest necessary conditions. The idea, inspired by [13], is implemented and experimentally verified. Though directly dealing with abduction, the approach in [8] concentrates on (valuable and important) experimental research and does not separate the weak forgetting operator from the standard forgetting operator.

As in the approach proposed in the current paper, the papers [33,34] use the Ackermann [2] and fixpoint [28] Lemmas, adjusted to the description logic formalism. Forgetting in description logics has also been addressed in [27] via uniform interpolants, where mixed model-theoretic and automata-theoretic approaches are applied. It is shown that computing uniform interpolants in the context of the description logics considered, is generally highly intractable. In [22], a resolution-based calculus is used to compute abducibles and eliminate (forget about) names outside of a given signature.

Forgetting is particularly useful in rule-based languages, when one simplifies a belief base to improve querying performance, or protect its parts [20,30,31]. This is especially useful for Answer Set Programs, where the corresponding entailment tasks, centering around necessary conditions, are typically intractable. In [30] *semantic forgetting* is proposed. It preserves skeptical and credulous consequences on unforgotten variables, as well as strong equivalence of ASP programs. It is shown that computing the forgetting result is intractable even for Horn logic programs. The paper [20], provides a comprehensive survey of the area as pertains to Answer Set Programs.

In summary, while the topic of forgetting in knowledge representation has gained considerable attention with numerous publications, the dual weak forgetting operator which is proposed and investigated in this paper, has not been explicitly considered in the literature. Additionally, the fundamental principles of forgetting founded on the entailment-based, inferential perspective, which is the starting point of this paper, is somewhat unique in comparison to other papers, where theoretical results are a consequence of model-theoretical, or other considerations. This perspective, which results in entailment preservation on a respective sublanguage, naturally leads to the consideration of both standard and weak forgetting operators.

Second-order quantifier elimination is used in this paper as the main logical tool for computing propositional, first-order and fixpoint equivalents of both standard and weak forgetting operators. Although there are several alternative techniques that could be used (see, e.g., [19]), Ackermann-like approaches have been selected for their expressive power and pragmatic application. Combining Ackermann-like lemmas with tautology-preserving formula transformations, as exhibited in this paper, has been shown to be very powerful and useful in many different contexts, including that of computing dual forgetting operators.

In particular, as shown in [11], the DLS algorithm subsumes most other known techniques that have been developed for computing circumscription. The paper [6], shows that the DLS algorithm covers all Sahlqvist formulas, an important class of formulas used in modal correspondence theory. The fixpoint lemma of [28], has been used originally for application to modal correspondence theory. This in turn has led to the development of the DLS* algorithm for fixpoint computations, which has been used for computing various forms of domain circumscription [12]. Both the DLS and DLS* algorithms have been shown to be useful in computing weakest sufficient and strongest necessary conditions [13]. The previously mentioned works [8,33], in addition to other papers by these authors, have shown that Ackermann-like techniques are also powerful tools for computing forgetting in description logics. The experimental results of these papers also show that such approaches are very efficient and applicable to real-world problems. For other uses of Ackermann-like approaches, including use of the DLS and DLS* algorithms, see [19].

8. Conclusions

This paper provides a general characterization of forgetting founded on an entailment-based, inferential framework. In doing so, an interesting forgetting operator, weak forgetting, is identified which is dual to the standard forgetting operator studied in [26]. Due to the entailment-based, inferential framework presented, it is shown how quantifier elimination techniques based on Ackermann's Lemma, using existing algorithms, DLS and DLS^*, can be used to compute output of both forgetting operators. Additionally, the tight relationship between weak forgetting and weakest sufficient conditions, and standard forgetting and strongest necessary conditions is characterized in a straightforward manner.

This paper first approaches the topics using the propositional logic case and then generalizes these results for the first-order and fixpoint logic case, offering new expressivity in the pragmatic use of such operators. Throughout, the paper provides examples justifying the introduction of the weak forgetting operator and also shows how the computational framework is used to derive inferences from application of both forgetting operators. Similarities and distinctions between related work and work presented in this paper are provided. This work opens up a new way to think about forgetting operators and their pragmatic application in general, by integrating diverse threads of previous research in this area and new results from this paper, in one uniform, formal, entailment-based, inferential framework.

Based on this approach, it is also shown in Theorems 3.1(p.3), 3.2(p.2), 6.1(p.3) and 6.2(p.2) that from the perspective of entailment, the operators $F^{SC}()$ and $F^{NC}()$, generate the formally best possible (maximal) results of forgetting, as expressed by the restricted target language in any application.[11]

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

No data was used for the research described in the article.

Acknowledgements

We are grateful to the anonymous reviewers for the careful reading of our paper and providing many valuable comments and suggestions. The first author has been partially supported by a research grant from the ELLIIT Network Organization for Information and Communication Technology, Sweden, in addition to a research grant from the Mahasarakham Development Fund, Mahasarakham University, Thailand. The second author has been supported by research grant 2017/27/B/ST6/02018 of the National Science Centre Poland.

References

[1] S. Abiteboul, R. Hull, V. Vianu, Foundations of Databases, Addison-Wesley, 1995.
[2] W. Ackermann, Untersuchungen über das eliminationsproblem der mathematischen logik, Math. Ann. 110 (1935) 390–413.
[3] C. Alchourrón, P. Gärdenfors, D. Makinson, On the logic of theory change: partial meet contraction and revision functions, J. Symb. Log. 50 (1985) 510–530.
[4] C. Areces, M. de Rijke, From description to hybrid logics, and back, in: F. Wolter, H. Wansing, M. de Rijke, M. Zakharyaschev (Eds.), Proc. Advances in Modal Logic, vol. 3, 2000, pp. 17–36.
[5] C. Beierle, G. Kern-Isberner, K. Sauerwald, T. Bock, M. Ragni, Towards a general framework for kinds of forgetting in common-sense belief management, Künstl. Intell. 33 (2019) 57–68.
[6] W. Conradie, On the strength and scope of DLS, J. Appl. Non-Class. Log. 16 (2006) 279–296.
[7] D. van Dalen, Logic and Structure, Universitext, Springer, 2013.
[8] W. Del-Pinto, R. Schmidt, Abox abduction via forgetting in ALC, in: The 33rd AAAI Conf. on AI, 2019, pp. 2768–2775.
[9] J. Delgrande, A knowledge level account of forgetting, J. Artif. Intell. Res. 60 (2017) 1165–1213.
[10] H. van Ditmarsch, A. Herzig, J. Lang, P. Marquis, Introspective forgetting, Synthese 169 (2009) 405–423.
[11] P. Doherty, W. Łukaszewicz, A. Szałas, Computing circumscription revisited, J. Autom. Reason. 18 (1997) 297–336.
[12] P. Doherty, W. Łukaszewicz, A. Szałas, General domain circumscription and its effective reductions, Fundam. Inform. 36 (1998) 23–55.
[13] P. Doherty, W. Łukaszewicz, A. Szałas, Computing strongest necessary and weakest sufficient conditions of first-order formulas, in: B. Nebel (Ed.), Proc. 17th IJCAI, 2001, pp. 145–154.
[14] T. Eiter, G. Kern-Isberner, A brief survey on forgetting from a knowledge representation and reasoning perspective, Künstl. Intell. 33 (2019) 9–33.
[15] L. Fang, Y. Liu, H. van Ditmarsch, Forgetting in multi-agent modal logics, Artif. Intell. 266 (2019) 51–80.
[16] R. Feng, E. Acar, Y. Wang, W. Liu, S. Schlobach, W. Ding, Computing sufficient and necessary conditions in CTL: a forgetting approach, Inf. Sci. 616 (2022) 474–504.
[17] R. Feng, Y. Wang, R. Qian, L. Yang, P. Chen, Knowledge forgetting in propositional μ-calculus, Ann. Math. Artif. Intell. (2023) 1–43.
[18] E. Fermé, S.O. Hansson, AGM 25 years: twenty-five years of research in belief change, J. Philos. Log. 40 (2011) 295–331.
[19] D. Gabbay, R. Schmidt, A. Szałas, Second-Order Quantifier Elimination: Foundations, Computational Aspects and Applications, Studies in Logic. College Pub., vol. 12, 2008.
[20] R. Gonçalves, M. Knorr, J. Leite, Forgetting in answer set programming – a survey, Theory Pract. Log. Program. (2021) 1–43.
[21] P. Hinman, Fundamentals of Mathematical Logic, Taylor & Francis, 2005.
[22] P. Koopmann, W. Del-Pinto, S. Tourret, R. Schmidt, Signature-based abduction for expressive description logics, in: D. Calvanese, E. Erdem, M. Thielscher (Eds.), Proc. 17th Int. Conf. KR'2020, 2020, pp. 592–602.
[23] J. Lang, P. Liberatore, P. Marquis, Propositional independence: formula-variable independence and forgetting, J. Artif. Intell. Res. 18 (2003) 391–443.
[24] J. Lang, P. Marquis, Reasoning under inconsistency: a forgetting-based approach, Artif. Intell. 174 (2010) 799–823.
[25] F. Lin, On strongest necessary and weakest sufficient conditions, in: A. Cohn, F. Giunchiglia, B. Selman (Eds.), Proc. 7th Int. Conf. KR'2000, 2000, pp. 167–175.
[26] F. Lin, R. Reiter, Forget it!, in: Proc. of the AAAI Fall Symp. on Relevance, 1994, pp. 154–159.
[27] C. Lutz, F. Wolter, Foundations for uniform interpolation and forgetting in expressive description logics, in: T. Walsh (Ed.), Proc. IJCAI'11, 2011, pp. 989–995.
[28] A. Nonnengart, A. Szałas, A fixpoint approach to second-order quantifier elimination with applications to correspondence theory, in: E. Orłowska (Ed.), Logic at Work: Essays Dedicated to the Memory of Helena Rasiowa, Springer Physica-Verlag, 1998, pp. 307–328.
[29] P. Peppas, Belief revision, in: F. van Harmelen, V. Lifschitz, B. Porter (Eds.), Handbook of KR, Elsevier, 2008, pp. 317–359.
[30] Y. Wang, K. Wang, M. Zhang, Forgetting for answer set programs revisited, in: F. Rossi (Ed.), Proc. IJCAI'2013, 2013, pp. 1162–1168.
[31] Y. Zhang, N. Foo, Solving logic program conflict through strong and weak forgettings, Artif. Intell. 170 (2006) 739–778.
[32] Y. Zhang, Y. Zhou, Forgetting revisited, in: F. Lin, U. Sattler, M. Truszczynski (Eds.), Proc. 12th Int. Conf. KR'2010, 2010, pp. 602–604.
[33] Y. Zhao, R. Schmidt, Role forgetting for $\mathcal{ALCOQH}$ (universal role)-ontologies using an Ackermann-based approach, in: C. Sierra (Ed.), Proc. IJCAI'17, 2017, pp. 1354–1361.
[34] Y. Zhao, R.A. Schmidt, Forgetting concept and role symbols in $\mathcal{ALCOIH}\mu^{+}(\nabla,\sqcap)$-ontologies, in: S. Kambhampati (Ed.), Proc. IJCAI'2016, 2016, pp. 1345–1352.

[11] See also Fig. 2 and the explanations in Section 3.4.