The Thirty-Second AAAI Conference
on Artificial Intelligence (AAAI-18)

# Dependence in Propositional Logic: Formula-Formula Dependence and Formula Forgetting — Application to Belief Update and Conservative Extension

**Liangda Fang,[1] Hai Wan,[2*] Xianqiao Liu,[2] Biqing Fang,[2] Zhaorong Lai[1]**
[1]College of Information Science and Technology, Jinan University, Guangzhou 510632, China
{fangld,laizhr}@jnu.edu.cn
[2]School of Data and Computer Science, Sun Yat-sen University, Guangzhou 510006, China
wanhai@mail.sysu.edu.cn, {liuxq35,fangbq3}@mail2.sysu.edu.cn

## Abstract

Dependence is an important concept for many tasks in artificial intelligence. A task can be executed more efficiently by discarding something independent from the task. In this paper, we propose two novel notions of dependence in propositional logic: formula-formula dependence and formula forgetting. The first is a relation between formulas capturing whether a formula depends on another one, while the second is an operation that returns the strongest consequence independent of a formula. We also apply these two notions in two well-known issues: belief update and conservative extension. Firstly, we define a new update operator based on formula-formula dependence. Furthermore, we reduce conservative extension to formula forgetting.

## Introduction

Dependence[1] is an important concept in artificial intelligence (AI). Before performing an intelligent task (*e.g.*, reasoning and planning), it is intuitive to first determine what are irrelevant, and discard them for better efficiency. For example, when a student takes an examination on literature, she does not need to keep the knowledge of mathematics in mind. This process of discarding irrelevant information involves two issues:

1. What is the irrelevant information in the background knowledge base (KB) for the task?
2. How to discard it in the KB?

In the context of propositional logic, various authors (Boutilier 1994; Lakemeyer 1997) studied the first issue by systematically analyzing a dependence relation between a formula and a variable, namely *formula-variable dependence* (FV-dependence). Loosely speaking, a formula $\phi$ depends on a variable $v$ if $v$ always occur in every formula equivalent to $\phi$. Further, Lang, Liberatore, and Marquis (2003) pointed out that, in some applications, we are concerned about not only which variable the formula depends on but also the polarity of the variable. To distinguish the case where a formula conveys some information about a literal but no information about its complement, they proposed a more fine-grained notion, namely *formula-literal dependence* (FL-dependence). For the second issue, *variable forgetting* is highly related to FV-dependence. It yields the strongest consequence independent of a variable. The intuitive meaning of literal forgetting is similar.

As mentioned in (Darwiche 1997), dependence is not only a philosophical notion but also a pragmatic notion. Over the past years, this notion has been widely used in many fields of AI, including automated reasoning (Levy, Fikes, and Sagiv 1997; Kautz, McAllester, and Selman 1997), knowledge compilation (Bryant 1992; Minato 1993; Darwiche 1997), reasoning about actions (Lin and Reiter 1997), and especially belief change (Zhang and Zhou 2009; Oveisi et al. 2017). Belief update, a type of belief change, studies how an agent modifies her belief base $\phi$ in the presence of new information $\psi$ in a dynamically changing environment. Herzig, Lang, and Marquis (2013) suggested that belief update should be based on the notion of dependence, and proposed the dependence-based update scheme that consists of first removing any belief on which the negation of the new information depends, and then adding the new information.

After investigating FL-dependence and literal forgetting, a natural next step is to study two more general notions: formula-formula dependence (FF-dependence) and formula forgetting. This paper is intended to fill this gap. The main contributions are as follows. First of all, we give a formal definition of FF-dependence, which is a dependence relation between formulas. We also provide a model-based characterization result and analyze some properties for FF-dependence. In addition, based on FF-dependence, we introduce formula forgetting as generalizations of literal forgetting. We give various equivalent formulations of this notion, including an axiomatic definition by four postulates, a syntactic definition via conditioning, and a model-theoretic definition. Finally, we apply these two notions in two well-known issues of AI: belief update and conservative extension. Following the dependence-based update scheme, we define an update operator $\diamond_{\mathsf{F}}$ by first forgetting the negation of the new information in the initial belief base, and then conjoining the resulting belief base with the new information. We characterize $\diamond_{\mathsf{F}}$ by a set of postulates and assess it against the well-known KM postulates. We compare it with other operators from three perspectives including informa-

[*] Corresponding author

[1] In this paper, we do not differentiate between "dependence" and "relevance", and use these terms interchangeably.

tion preservation, computational complexity and empirical results. The comparison shows that $\diamond_{\mathsf{F}}$ is a suitable alternative to belief update. We finally give the correspondence between conservative extension and formula forgetting. It turns out that the former can be reduced to the latter.

## Preliminaries

In this section, we first recall some basic concepts of propositional logic, and then present the notions of FL-dependence and literal forgetting. Most contents of the first two subsections originate from (Lang, Liberatore, and Marquis 2003). Finally, we review some background work regarding belief update.

### Propositional logic

We assume the propositional language $\mathcal{L}$ is built from a finite set $P$ of variables, the connectives $\neg$, $\vee$, $\wedge$ and two logical constants $\top$ (true) and $\bot$ (false). We use $\phi$, $\psi$, $\eta$ and $\xi$ to range over formulas. The notation Var($\phi$) denotes the set of variables appearing in $\phi$. A formula is *trivial* if it is equivalent to $\top$ or $\bot$. A *literal* is a variable (positive literal) or a negated one (negative literal). For a literal $l$, $\bar{l}$ denotes the complementary literal of $l$. A *term* is a conjunction of literals. A *disjunctive normal form (DNF)* is a disjunction of terms. We say that a term $t$ is in a DNF formula $\phi$, written $t \in \phi$, if $t$ is a disjunct of $\phi$. For a subset $X$ of $P$, a *minterm* over $X$ is a term where it uses only $X$ and each variable of $X$ appears exactly once. For simplify, we omit $X$ if $X = P$. We use $\Omega_X$ to denote the set of minterms over $X$.

A formula is in *full DNF*, if it is a disjunction of minterms. A formula is in *negation normal form (NNF)*, if $\neg$ is only applied to variables. It is well-known that every formula can be equivalently transformed into full DNF and NNF. A full DNF formula of $\phi$ is the disjunction of all minterms entailing $\phi$. Throughout this paper, we take all variables of $P$ into consideration, and assume that there is a unique full DNF formula equivalent to $\phi$. We call it the *full DNF formula* of $\phi$. An NNF formula of $\phi$ can be acquired by pushing the negation inwards via De Morgan's law and eliminating double negation. For convenience, we call the formula generated by the above process the *NNF formula* of $\phi$.

Let $\phi$ be an NNF formula. The variable $x$ in $\phi$, where $x$ is not in the scope of $\neg$, is called an occurrence of a positive literal $x$ in $\phi$. The symbols $\neg x$ in $\phi$ are called an occurrence of a negative literal $\neg x$ in $\phi$. Note that any occurrence of the negative literal $\neg x$ cannot be considered as an occurrence of its complement $x$. A simple example is that the formula $\neg x$ contains no occurrence of $x$.

An *interpretation* is a subset of $P$. A model of $\phi$ is an interpretation satisfying $\phi$, and $[\![\phi]\!]$ denotes the set of models of $\phi$. A formula $\phi$ is *satisfiable*, if there is a model of $\phi$. A satisfiable term is a term satisfied by at least one interpretation. In other words, it does not contain positive and negative literals of the same variable simultaneously. The following is an operation on an interpretation w.r.t. a satisfiable term.

**Definition 1.** Let $\omega$ be an interpretation and $t$ a satisfiable term. Forcing $\omega$ on $t$, written $\omega_{\rightarrow t}$, is defined as $(\omega \backslash \text{Var}(t)) \cup \{x \mid x \in P \text{ and } t \models x\}$.

Both $\omega$ and $\omega_{\rightarrow t}$ agree on the valuations of all variables except those of Var($t$). The latter is the model of $t$ that is the closest to $\omega$. For instance, let $\omega = \{a, b\}$ and $t = \bar{a} \wedge c$, then $\omega_{\rightarrow t} = \{b, c\}$.

### Formula-literal dependence and literal forgetting

The main intuition FL-dependence aims to capture is that the literal $l$ is an indispensable part of the formula $\phi$: roughly speaking, every NNF formula equivalent to $\phi$ contains the occurrence of $l$.

**Definition 2.** Let $\phi$ be a formula and $l$ a literal. We say $\phi$ is *Lit-dependent* on $l$, written $l \mapsto_{\mathsf{L}} \phi$, if every NNF formula equivalent to $\phi$ contains the occurrence of $l$. Otherwise, $\phi$ is *Lit-independent* from $l$, written $l \not\mapsto_{\mathsf{L}} \phi$.

**Example 1.** Let $\phi = \neg(a \wedge \neg b) \wedge (b \vee \neg b)$, and $\psi = \neg a \vee b$. It is obvious that $\psi$ is an NNF formula equivalent to $\phi$, and does not contain $\neg b$. Hence neither $\phi$ nor $\psi$ depends on $\neg b$.

Although Definition 2 is a syntactic formulation of FL-dependence, it is syntax-independent.

**Proposition 1.** *Let $\phi$ and $\phi'$ be formulas where $\phi \equiv \phi'$, and $l$ a literal. Then, $l \mapsto_{\mathsf{L}} \phi$ iff $l \mapsto_{\mathsf{L}} \phi'$.*

The notion of FV-dependence can be easily defined from that of FL-dependence.

**Definition 3.** Let $\phi$ be a formula and $x$ a variable. We say $\phi$ is *Var-dependent* on $x$, written $x \mapsto_{\mathsf{V}} \phi$, if $x \mapsto_{\mathsf{L}} \phi$ or $\bar{x} \mapsto_{\mathsf{L}} \phi$.

We also say $x$ is a *dependent variable* of $\phi$, if $x \mapsto_{\mathsf{V}} \phi$. We use DepLit($\phi$) (resp. DepVar($\phi$)) to denote the set of dependent literals (resp. variables) of $\phi$.

We hereafter present the notion of literal forgetting. We first introduce *term conditioning* (Darwiche 1998) that is an important syntactic operation for literal forgetting.

**Definition 4.** Let $\phi$ be a formula and $t$ a satisfiable term. The *conditioning* of $\phi$ on $t$, written $\phi|_t$, is defined by substituting each variable $x$ of $\phi$ by $\top$ (resp. $\bot$) if $x$ (resp. $\bar{x}$) is a positive (resp. negative) literal of $t$.

For instance, conditioning $(a \wedge \bar{b}) \vee (b \wedge c)$ on $\bar{a} \wedge b$ gives a formula $(\bot \wedge \neg\top) \vee (\top \wedge c)$, which is equivalent to $c$.

According to the Shannon expansion (Shannon 1938), any formula $\phi$ can be decomposed into $(l \wedge \phi|_l) \vee (\bar{l} \wedge \phi|_{\bar{l}})$. From the syntactic point of view, forgetting $l$ just removes the occurrence of $l$ from the expansion. The notion of literal forgetting can be defined in a syntactic way via conditioning.

**Definition 5.** Let $\phi$ be a formula and $l$ a literal. The result of forgetting $l$ in $\phi$, written $\exists_{\mathsf{L}} l.\phi$, is defined as $\phi|_l \vee (\bar{l} \wedge \phi|_{\bar{l}})$.

For example, $\exists_{\mathsf{L}} a.[(a \wedge b) \vee (\bar{a} \wedge c)] \equiv [(\top \wedge b) \vee (\neg\top \wedge c)] \vee \{\bar{a} \wedge [(\bot \wedge b) \vee (\neg\bot \wedge c)]\} \equiv b \vee (\bar{a} \wedge c)$.

One of key propositions of literal forgetting is as follows: Forgetting a literal $l$ generates the strongest consequence that does not depend on $l$.

**Proposition 2.** *Let $\phi$ be a formula and $l$ a literal. $\exists_{\mathsf{L}} l.\phi$ is the strongest consequence of $\phi$ that is Lit-independent of $l$.*

Forgetting a single literal can be extended to a variable, a set of literals or variables. Forgetting a variable $x$ is defined by forgetting the positive literal $x$ and the negative one $\bar{x}$ sequentially. As the order in which literals of $L$ are handled does not matter, forgetting a set $L$ of literals can be sequentially computed by forgetting a single literal $l$ of $L$ one by one. The operation of forgetting a set of variables is similar.

**Definition 6.** Let $L$ be a set of literals, $X$ a set of variables, $l \in L$ and $x \in X$.

- $\exists_{\mathsf{V}} x.\phi = \exists_{\mathsf{L}} \bar{x}.(\exists_{\mathsf{L}} x.\phi)$;
- $\exists_{\mathsf{L}} L.\phi = \exists_{\mathsf{L}} L \setminus \{l\}.(\exists_{\mathsf{L}} l.\phi)$;
- $\exists_{\mathsf{V}} X.\phi = \exists_{\mathsf{V}} (X \setminus \{x\}).(\exists_{\mathsf{V}} x.\phi)$.

Note that if $L = \emptyset$ (resp. $X = \emptyset$), we let $\exists_{\mathsf{L}} L.\phi = \phi$ (resp. $\exists_{\mathsf{V}} X.\phi = \phi$).

Finally, we establish the link between term conditioning and variable forgetting. The conditioning of $\phi$ on $t$ can be computed via forgetting all dependent variables of $t$ in $\phi \wedge t$.

**Proposition 3.** *Let $\phi$ be a formula and $t$ a satisfiable term. Then, $\phi|_t \equiv \exists_{\mathsf{V}} \mathit{DepVar}(t).(\phi \wedge t)$.*

## Belief update

Belief update focuses on the evolution of the belief base in line with the new information, reflecting the modification of the world. Based on the principle of minimal change, Katsuno and Mendelzon (1991) proposed the KM postulates to capture rational belief update operators, which map the initial belief base $\phi$ and the new information $\psi$ to a new belief base $\phi \diamond \psi$. The postulates are as follows:

**U1** $\phi \diamond \psi \models \psi$;

**U2** If $\phi \models \psi$, then $\phi \diamond \psi \equiv \phi$;

**U3** If $\phi$ and $\psi$ are satisfiable, then $\phi \diamond \psi$ is also satisfiable;

**U4** If $\phi \equiv \phi'$ and $\psi \equiv \psi'$, then $\phi \diamond \psi \equiv \phi' \diamond \psi'$;

**U5** $(\phi \diamond \psi) \wedge \eta \models \phi \diamond (\psi \wedge \eta)$;

**U6** If $\phi \diamond \psi \models \psi'$ and $\phi \diamond \psi' \models \psi$, then $\phi \diamond \psi \equiv \phi \diamond \psi'$;

**U7** If $\phi$ is a minterm, then $(\phi \diamond \psi) \wedge (\phi \diamond \psi') \models \phi \diamond (\psi \vee \psi')$;

**U8** $(\phi \vee \phi') \diamond \psi \equiv (\phi \diamond \psi) \vee (\phi' \diamond \psi)$.

According to the postulate (U8), the new belief base $\phi \diamond \psi$ collects the updates of each model of $\phi$ by $\psi$. More formally,

$$[\![\phi \diamond \psi]\!] = \bigcup_{\omega \in [\![\phi]\!]} (\omega \diamond \psi).$$

The Possible Models Approach (PMA) and Forbus operators, introduced by Winslett (1988) and Forbus (1989) respectively, are two famous update operators. Both of them are based on the principle of minimal change, thereby satisfying all of the KM postulates. The PMA operator $\diamond_{\mathsf{PMA}}$ is based on minimization of the distance between interpretations, while the Forbus operator $\diamond_{\mathsf{For}}$ is defined in terms of the cardinality of distances. The definitions of updating an interpretation $\omega$ by $\psi$ of $\diamond_{\mathsf{PMA}}$ and $\diamond_{\mathsf{For}}$ are as follows:

$$\omega \diamond_{\mathsf{PMA}} \psi = \{\nu \in [\![\psi]\!] \mid \forall \mu \in [\![\psi]\!].(\omega \ominus \nu) \subseteq (\omega \ominus \mu)\};$$
$$\omega \diamond_{\mathsf{For}} \psi = \{\nu \in [\![\psi]\!] \mid \forall \mu \in [\![\psi]\!].|\omega \ominus \nu| \leq |\omega \ominus \mu|\}.$$

where $\omega \ominus \nu$ is the symmetric difference between $\omega$ and $\nu$ and $|\omega \ominus \nu|$ is the cardinality of $\omega \ominus \nu$.

On the other hand, Herzig, Lang, and Marquis (2013) suggested that belief update should be based on the notion of dependence, and proposed the dependence-based update scheme, consisting of two steps: (1) forget every piece of $\phi$ which $\neg\psi$ depends on; (2) expand the resulting belief base with $\psi$. This scheme has some resemblance to the so-called Levi Identity (Levi 1977) which defines an update operator from a given erasure operator. Based on the notion of FV-dependence, Herzig and Rifi (1998) proposed an operator $\diamond_{\mathsf{V}}$ which first forgets in $\phi$ all variables on which $\neg\psi$ depends[2]. More precisely,

$$\phi \diamond_{\mathsf{V}} \psi = (\exists_{\mathsf{V}} \mathrm{DepVar}(\neg\psi).\phi) \wedge \psi.$$

Herzig, Lang, and Marquis (2013) criticized the update operator $\diamond_{\mathsf{V}}$ for forgetting too much in the initial belief base, and proposed an update operator $\diamond_L$ by forgetting all dependent literals of $\neg\psi$ rather than dependent variables. The definition of $\diamond_{\mathsf{L}}$ is as follows:

$$\phi \diamond_{\mathsf{L}} \psi = (\exists_{\mathsf{L}} \mathrm{DepLit}(\neg\psi).\phi) \wedge \psi.$$

Besides the above work about dependence-based belief update, Parikh (1999) proposed a postulate (P) of relevance for belief change. The following is a stronger version introduced by Peppas et al. (2015).

**SP** If $\mathrm{Var}(\phi) \cap \mathrm{Var}(\phi') = \emptyset$ and $\mathrm{Var}(\psi) \subseteq \mathrm{Var}(\phi')$, then $(\phi \wedge \phi') \diamond \psi \equiv \phi \wedge (\phi' \diamond \psi)$.

Postulate (SP) says that if a belief base can be split into two disjoint compartments, then only the compartment affected by the new information will be modified.

# Formula-formula dependence

In this section, we introduce the notion of FF-dependence. We first extend the notion of occurrence of literals to that of arbitrary formulas. We then define the notion of FF-dependence in terms of occurrence. We further give a model-theoretical characterization of FF-dependence. Finally, some properties of FF-dependence are given.

Recall that the definition of FL-dependence (cf. Definition 2) says that a formula $\phi$ is Lit-dependent on a literal $l$ if every NNF formula equivalent to $\phi$ contains the occurrence of $l$. It is hard to extend this definition to FF-dependence. We therefore resort to another equivalent definition: $\phi$ does not depend on $l$ if substituting every occurrence of $l$ in the NNF formula of $\phi$ with $\top$ leads to a logically different formula.

This definition cannot be directly transferred to the notion of FF-dependence. It is possible that $\phi$ depends on $\psi$ but the latter does not explicitly occur in the former even if both $\phi$ and $\psi$ are in NNF. For example, consider the two equivalent formulas $\phi = \bar{a} \vee b$ and $\psi = (\bar{a} \vee b) \wedge (b \vee \bar{b})$. Because $\phi$ appears in itself, so replacing $\phi$ in itself by $\top$ results in $\top$. We get that $\top \not\equiv \phi$, and hence $\phi$ depends on itself. Based on the principle of syntax-independence, $\phi$ should depend on $\psi$. However, replacing all occurrences of $\psi$ in $\phi$ does not modify $\phi$ since $\psi$ does not explicitly occur in $\phi$. We obtain that $\phi$ does not depend on $\psi$.

To define the notion of FF-dependence, we need to solve two problems:

[2] Two equivalent update operators were proposed in (Hegner 1987) and (Doherty, Łukaszewicz, and Madalinska-Bugaj 1998) respectively.

1. Which normal form is suitable for the notion of FF-dependence?
2. How to define the occurrence of a formula in the normal form of another formula?

For the first problem, we choose full DNF instead of NNF. In the following, we introduce the notion of occurrence of a formula in the full DNF of another formula. This notion should obey two principles: *syntax-independence* and *self-elimination*. Suppose that two formulas $\phi$ and $\phi'$ are equivalent. The former requires that "given a full DNF formula $\psi$, the occurrence of $\phi$ in $\psi$ is also that of $\phi'$ in $\psi$". The latter means that "if $\phi'$ is in full DNF, then replacing the occurrence of $\phi$ in $\phi'$ with $\top$ results in $\top$". To illustrate the definition of occurrence, it is necessary to define the notion of dependent minterms.

**Definition 7.** We say a term $t$ is a *dependent minterm* of $\phi$, if $t$ is a minterm over $\mathrm{DepVar}(\phi)$ and $t \models \phi$.

We use $\Omega_\phi$ to denote the set of dependent minterms of $\phi$. We also let $\Omega_\phi$ be empty if $\phi \equiv \bot$, and $\Omega_\phi$ be $\{\top\}$ if $\phi \equiv \top$. Clearly, the disjunction of $\Omega_\phi$ is equivalent to $\phi$.

**Example 2.** *Let $\psi = (a \vee b) \wedge (c \vee \bar{c})$. The sets of dependent variables and dependent minterms of $\psi$ are as follows: DepVar($\psi$) $= \{a, b\}$ and $\Omega_\psi = \{a \wedge b, a \wedge \bar{b}, \bar{a} \wedge b\}$. Neither $\bar{a} \wedge b \wedge c$ nor $\bar{a} \wedge \bar{b}$ is a dependent minterm of $\psi$ since the former contains the variable $c$ not in DepVar($\phi$), and the latter does not entails $\phi$.*

We hereafter give a definition of occurrence. We first consider a simple case that is the occurrence of a term in a minterm.

**Definition 8.** Let $t$ be a term and $t'$ be a minterm. We say a literal $l$ of $t'$ is an occurrence of $t$ in $t'$, if $l$ is an occurrence of $t$ and $t' \models t$.

**Example 3.** *Consider three terms $t_1 = a \wedge \bar{b}$, $t_2 = a \wedge b$, and $t' = a \wedge \bar{b} \wedge \bar{c}$. The literals $a$ and $\bar{b}$ are occurrences of $t_1$ in $t'$. But neither of them is an occurrence of $t_2$ in $t'$ since $t' \not\models t_2$. Moreover, $t'$ contains no occurrence of $t_2$.*

The notion of occurrence of a formula $\psi$ in a full DNF formula $\phi$ can be defined as that of every dependent minterm of $\psi$ in every disjunct of $\phi$.

**Definition 9.** Let $\phi$ and $\psi$ be two formulas where $\phi$ is in full DNF. Let $t'$ be a minterm of $\phi$, and $l$ a literal of $t'$. We say the occurrence of $l$ in $\phi$ is an occurrence of $\psi$ in $\phi$, if there is a dependent minterm $t$ of $\psi$ s.t. $l$ is an occurrence of $t$ in $t'$.

Since two equivalent formulas have the same set of dependent minterms, Definition 9 satisfies syntax-independence. It is easily verified that Definition 9 satisfies self-elimination.

**Example 4.** *Continued with Example 2, we have $\psi = (a \vee b) \wedge (c \vee \bar{c})$ and $\Omega_\psi = \{a \wedge b, a \wedge \bar{b}, \bar{a} \wedge b\}$. Consider the full DNF formula $\phi = (a \wedge \bar{b} \wedge c) \vee (\bar{a} \wedge b \wedge c) \vee (\bar{a} \wedge \bar{b} \wedge c) \vee (\bar{a} \wedge \bar{b} \wedge \bar{c})$. The occurrences of $a$ and $\bar{b}$ in the first term $a \wedge \bar{b} \wedge c$ of $\phi$ are that of $\psi$ in $\phi$. So are the occurrences of $\bar{a}$ and $b$ in the second term. However, the occurrences of $\bar{a}$ and $\bar{b}$ in the third and fourth disjuncts are not since no dependent minterm of $\psi$ satisfies the condition of Definition 9.*

We now are ready to define the notion of FF-dependence.

**Definition 10.** We say $\phi$ is *Fml-dependent* on $\psi$, written $\psi \mapsto_{\mathsf{F}} \phi$, if substituting every occurrence of $\psi$ in the full DNF formula of $\phi$ with $\top$ leads to another formula that is not equivalent to $\phi$. Otherwise, $\phi$ is Fml-independent from $\psi$, written $\psi \not\mapsto_{\mathsf{F}} \phi$.

We illustrate Definition 10 with the following example.

**Example 5.** *Continued with Example 4, we have $\psi = (a \vee b) \wedge (c \vee \bar{c})$. Let $\phi' = (\bar{a} \wedge \bar{b}) \vee (\bar{a} \wedge c) \vee (\bar{b} \wedge \bar{c})$. It is easily verified that $\phi \equiv \phi'$. Replacing every occurrence of $\psi$ in $\phi$ with $\top$ results in a formula $(\top \wedge \top \wedge c) \vee (\top \wedge \top \wedge c) \vee (\bar{a} \wedge \bar{b} \wedge c) \vee (\bar{a} \wedge \bar{b} \wedge \bar{c}) \equiv c \vee (\bar{a} \wedge \bar{b})$. Clearly, this formula is not equivalent to $\phi$. Hence, both $\phi$ and $\phi'$ depend on $\psi$.*

Definition 10 is a simple approach to define the notion of FF-dependence. To capture this notion comprehensively, we present a model-theoretic characterization.

**Proposition 4.** Let $\phi$ and $\psi$ be formulas, and $X = \mathrm{DepVar}(\psi)$. Then, $\phi$ is *Fml-dependent* on $\psi$ iff there exist an interpretation $\omega$ and two terms $t \in \Omega_\psi$ and $t' \in \Omega_X \setminus \{t\}$ s.t. $\omega \models \phi \wedge t$ and $\omega_{\to t'} \not\models \phi$.

This proposition means that $\phi$ depends on $\psi$, if there exist an interpretation $\omega$ and a dependent minterm $t$ of $\psi$ such that (1) $\omega \models \phi \wedge t$; and (2) $t$ is essential for $\omega$ to satisfy $\phi$, *i.e.*, there is a minterm $t'$ over $\mathrm{DepVar}(\psi)$ such that forcing $\omega$ on $t'$ no longer satisfies $\phi$.

Next, we continue Example 5 to illustrate Proposition 4.

**Example 6.** *We have $\phi' = (\bar{a} \wedge \bar{b}) \vee (\bar{a} \wedge c) \vee (\bar{b} \wedge \bar{c})$, $\psi = (a \vee b) \wedge (c \vee \bar{c})$ and $\Omega_\psi = \{a \wedge b, a \wedge \bar{b}, \bar{a} \wedge b\}$. Then, we let $\omega = \{a, c\}$, $t = a \wedge \bar{b}$ and $t' = a \wedge b$. Clearly, $t \in \Omega_\psi$, $t' \in \Omega_X \setminus \{t\}$ and $\omega \models \phi \wedge t$. Thus, $\omega_{\to t'} = \{a, b, c\}$ that does not satisfy $\phi$. So $\phi$ depends on $\psi$.*

Finally, we analyze some properties of FF-dependence. It is obvious that a formula is Fml-dependent on a literal if and only if the former is Lit-dependent on the latter.

**Proposition 5.** *Let $\phi$ be a formula and $l$ be a literal. Then $l \mapsto_{\mathsf{L}} \phi$ iff $l \mapsto_{\mathsf{F}} \phi$.*

FF-dependence satisfies syntax-independence, symmetry and almost reflexivity (*i.e.*, every non-trivial formula depends on itself). red Trivial formulas do not depend on any formula.

**Proposition 6.**

- *Syntax-independence: $\phi \mapsto_{\mathsf{F}} \psi$ iff $\phi' \mapsto_{\mathsf{F}} \psi'$ when $\phi \equiv \phi'$ and $\psi \equiv \psi'$;*
- *Symmetry: $\phi \mapsto_{\mathsf{F}} \psi$ iff $\psi \mapsto_{\mathsf{F}} \phi$;*
- *Almost reflexivity: $\phi \mapsto_{\mathsf{F}} \phi$ when $\phi$ is not trivial.*
- *For any formula $\phi$, $\phi \not\mapsto_{\mathsf{F}} \top$ and $\phi \not\mapsto_{\mathsf{F}} \bot$.*

We analyze the computational complexity of FF-dependence as follows:

**Proposition 7.** *FF-dependence is in $\Delta_2^{\mathsf{P}}$ and $\mathsf{NP}$-hard.*

*Proof.* Upper bound: First, we compute $\mathrm{DepVar}(\psi)$ by calling an $\mathsf{NP}$ oracle that decides whether $\psi$ depends on $v$ for each variable of $\psi$. Further, we guess an interpretation $\omega$ and two terms $t$ and $t'$, and then check whether they satisfy the condition of Proposition 4. The whole procedure calls $\mathsf{NP}$

oracles $n + 1$ times where $n$ is the number of $P(\psi)$. So, FF-dependence is in $\Delta_2^{\mathsf{P}}$.

Lower bound: By Proposition 5, FL-dependence is a restriction of FF-dependence. The complexity of FL-dependence is $\mathsf{NP}$-complete (Lang, Liberatore, and Marquis 2003). Hence, FF-dependence is $\mathsf{NP}$-hard. □

## Formula forgetting

In the previous section, we investigate the notion of FF-dependence. Following this notion, we study the notion of formula forgetting in this section. We first propose a set of postulates that precisely characterize the notion of formula forgetting. We then discuss some properties of formula forgetting. Finally, the computation and model-theoretic characterization of formula forgetting are also studied.

Zhang and Zhou (2009) proposed four postulates (W), (IR), (PP) and (NP) for variable forgetting in modal logic S5. We extend these postulates to formula forgetting in propositional logic.

**Definition 11.** We say $\psi$ is a result of forgetting $\eta$ in $\phi$, if it satisfies the following postulates:

**W** Weakening: $\phi \models \psi$;

**IR** Independence: $\eta \not\mapsto_{\mathsf{F}} \psi$;

**PP** Positive Persistence: for any formula $\xi$, if $\eta \not\mapsto_{\mathsf{F}} \xi$ and $\phi \models \xi$, then $\psi \models \xi$;

**NP** Negative Persistence: for any formula $\xi$, if $\eta \not\mapsto_{\mathsf{F}} \xi$ and $\phi \not\models \xi$, then $\psi \not\models \xi$.

Postulate (W) says that forgetting weakens the original formula. Postulate (IR) requires that after forgetting, the resulting formula should be irrelevant to the formula which we have forgotten. Finally, postulates (PP) and (NP) viewed together state that forgetting $\eta$ does not affect entailment of queries that are Fml-independent from $\eta$.

Similarly to literal forgetting, the result of formula forgetting is unique up to logical equivalence. We use $\exists_{\mathsf{F}}\psi.\phi$ to denote the result of forgetting $\psi$ in $\phi$.

The following proposition reflects strong relationships between formula forgetting and FF-dependence: Forgetting $\psi$ in $\phi$ does not change $\phi$ if and only if $\phi$ does not depend on $\psi$.

**Proposition 8.** *Let $\phi$ and $\psi$ be two formulas. Then, $\phi \equiv \exists_{\mathsf{F}}\psi.\phi$ iff $\psi \not\mapsto_{\mathsf{F}} \phi$.*

The following proposition further illustrates some essential properties of formula forgetting.

**Proposition 9.** *1. $\exists_{\mathsf{F}} l.\phi \equiv \exists_{\mathsf{L}} l.\phi$ for any literal $l$;*

*2. $\phi$ is satisfiable iff $\exists_{\mathsf{F}}\psi.\phi$ is satisfiable.*

*3. If $\phi \equiv \phi'$ and $\psi \equiv \psi'$, then $\exists_{\mathsf{F}}\psi.\phi \equiv \exists_{\mathsf{F}}\psi'.\phi'$.*

*4. $\exists_{\mathsf{F}}\psi.(\phi \vee \phi') \equiv (\exists_{\mathsf{F}}\psi.\phi) \vee (\exists_{\mathsf{F}}\psi.\phi')$;*

*5. If DepVar$(\phi)$ $\cap$ DepVar$(\psi) = \emptyset$, then $\exists_{\mathsf{F}}\psi.(\phi \wedge \phi') \equiv \phi \wedge (\exists_{\mathsf{F}}\psi.\phi')$.*

Firstly, formula forgetting is a generalization of literal forgetting. Secondly, forgetting any formula preserves satisfiability of the original formula. Thirdly, it is syntax-irrelevance, and distributive over disjunction. Finally, forgetting $\psi$ in a conjunction of two formulas does not affect the counterpart that shares no dependent variable of $\psi$.

We next investigate the computation of formula forgetting. Recall the definition of literal forgetting (cf. Definition 5), forgetting a literal $l$ in $\phi$ consists of two steps:

1. Transform $\phi$ into $(l \wedge \phi|_l) \vee (\bar{l} \wedge \phi|_{\bar{l}})$ via the Shannon expansion;
2. Eliminate the occurrence of $l$, *i.e.*, $\phi|_l \vee (\bar{l} \wedge \phi|_{\bar{l}})$.

The Shannon expansion can be generalized to a multi-variable expansion w.r.t. a set $X$ of variables, *i.e.*, $\phi \equiv \bigvee_{t \in \Omega_X} (t \wedge \phi|_t)$. It is natural to imagine that forgetting a formula $\psi$ in $\phi$ should first decompose $\phi$ w.r.t. the set of dependent variables of $\psi$, and then remove every dependent minterm of $\psi$. We therefore obtain a brute-force computation of formula forgetting as follows.

**Proposition 10.** Let $X$ = DepVar$(\psi)$. Then, $\exists_{\mathsf{F}}\psi.\phi \equiv [\bigvee_{t \in \Omega_\psi} (\phi|_t)] \vee [\bigvee_{t \in \Omega_X \setminus \Omega_\psi} (t \wedge \phi|_t)]$.

The computation of formula forgetting via Proposition 10 is computationally expensive when DepVar$(\psi)$ is large. To simplify the computation, we give another approach via conditioning. It is hard to extend the definition of term conditioning (cf. Definition 4) to formula conditioning since a formula may contain a literal and its negation simultaneously, and it may even be unsatisfiable. We hereafter resort to Proposition 3, and formalize the notion of formula conditioning via variable forgetting.

**Definition 12.** The conditioning of $\phi$ on $\psi$, written $\phi|_\psi$, is defined as $\exists_{\mathsf{V}}$DepVar$(\psi).(\phi \wedge \psi)$.

Note that if $\psi \equiv \top$ (resp. $\bot$), then $\phi|_\psi \equiv \phi$ (resp. $\bot$).

From now on, we adopt Definition 12 as the definition of the notation $\phi|_\psi$ when $\psi$ is a satisfiable term $t$.

A semantic characterization of formula conditioning is as follow: the models of the conditioning of $\phi$ on $\psi$ is the union of model $\omega$ where forcing it on any dependent minterm of $\psi$ leads to a model of $\phi$.

**Proposition 11.** $[\![\phi|_\psi]\!] = \bigcup_{t \in \Omega_\psi} \{\omega \mid \omega_{\rightarrow t} \models \phi\}$.

The following proposition states that the result of forgetting $\psi$ in $\phi$ is equivalent to the disjunction of $\phi$ and the conditioning of $\phi$ on $\psi$.

**Proposition 12.** $\exists_{\mathsf{F}}\psi.\phi \equiv \phi \vee \phi|_\psi$.

Now, we use an example to illustrate the computation of formula forgetting.

**Example 7.** *Continued with Example 5. We have $\phi' = (\bar{a} \wedge \bar{b}) \vee (\bar{a} \wedge c) \vee (\bar{b} \wedge \bar{c})$, $\psi = (a \vee b) \wedge (c \vee \bar{c})$, and DepVar$(\psi) = \{a, b\}$.*

*Firstly, the conjunction of $\phi'$ and $\psi$ is as follows:*

$$\begin{aligned}\phi' \wedge \psi &= [(\bar{a} \wedge \bar{b}) \vee (\bar{a} \wedge c) \vee (\bar{b} \wedge \bar{c})) \wedge ((a \vee b) \wedge (c \vee \bar{c})] \\ &\equiv ((a \wedge \bar{b}) \vee (\bar{a} \wedge b)) \wedge c.\end{aligned}$$

*Then, conditioning of $\phi'$ on $\psi$ leads to:*

$$\phi'|_\psi = \exists_{\mathsf{V}}\{a, b\}.[((a \wedge \bar{b}) \vee (\bar{a} \wedge b)) \wedge c] \equiv c.$$

*Hence, the result of forgetting $\psi$ in $\phi'$ is*

$$\exists_{\mathsf{F}}\psi.\phi' = \phi \vee \phi'|_\psi = [(\bar{a} \wedge \bar{b}) \vee (\bar{a} \wedge c) \vee (\bar{b} \wedge \bar{c})] \vee c \equiv (\bar{a} \wedge \bar{b}) \vee c.$$

Theoretically, the computation of formula forgetting via Proposition 12 cause single-exponential blowup in the size of the original formula unless $\mathsf{P} = \mathsf{NP}$. But this is a practical solution by using existing techniques of knowledge compilation, *e.g.*, binary decision diagrams (BDDs) (Bryant 1992). BDD is a compact form of propositional formula, and supports efficient boolean operations. This will be shown in our experimental evaluation.

By Propositions 11 and 12, we get a semantic characterization of formula forgetting. Forgetting $\psi$ in $\phi$ amounts to introduce the models of the conditioning of $\phi$ on $\psi$.

**Corollary 1.** $[\![\exists_{\mathsf{F}}\psi.\phi]\!] = [\![\phi]\!] \cup \bigcup_{t\in\Omega_\psi}\{\omega \mid \omega_{\to t} \models \phi\}$.

## Belief update based on FF-dependence

In this section, following the dependence-based update scheme, we define a new update operator $\diamond_{\mathsf{F}}$ based on FF-dependence. Then, we completely capture our update operator by identifying two extra postulates, show that the update operator satisfy the postulates (U1) - (U4) and (U8) and identify a special case in which all of the KM postulates holds. Finally, we compare $\diamond_{\mathsf{F}}$ with other operators $\diamond_{\mathsf{V}}$, $\diamond_{\mathsf{L}}$, $\diamond_{\mathsf{PMA}}$ and $\diamond_{\mathsf{For}}$ from various perspectives including information preservation, computational complexity and experimental results.

### Belief update via formula forgetting

The update operator based on FF-dependence is defined in terms of formula forgetting.

**Definition 13.** Let $\phi$ and $\psi$ be formulas. The update operator $\diamond_{\mathsf{F}}$, is defined as $\phi \diamond_{\mathsf{F}} \psi = [\exists_{\mathsf{F}}(\neg\psi).\phi] \wedge \psi$.

The following example illustrates the mechanism of the update operator $\diamond_{\mathsf{F}}$.

**Example 8.** *Continued Example 5, we have* $\phi' = (\bar{a} \wedge \bar{b}) \vee (\bar{a} \wedge c) \vee (\bar{b} \wedge \bar{c})$, $\psi = (a \vee b) \wedge (c \vee \bar{c})$. *We let* $\eta = \bar{a} \wedge \bar{b}$ *that is equivalent to* $\neg\psi$. *The procedure of updating* $\phi'$ *by* $\eta$ *via* $\diamond_{\mathsf{F}}$ *consists of two steps:*

1. *Forget* $\neg\eta$ *in* $\phi'$: $\exists_{\mathsf{F}}(\neg\eta).\phi' \equiv (\bar{a} \wedge \bar{b}) \vee c$.
2. *Conjoin the result of formula forgetting with* $\eta$: $[\exists_{\mathsf{F}}(\neg\eta).\phi'] \wedge \eta = [(\bar{a} \wedge \bar{b}) \vee c] \wedge (\bar{a} \wedge \bar{b}) \equiv \bar{a} \wedge \bar{b}$.

In the following, we give the model-theoretical characterization of $\diamond_{\mathsf{F}}$. We first provide the definition of the update of an interpretation by the new information.

**Definition 14.** Let $\omega$ be an interpretation. The update $\omega \diamond_{\mathsf{F}} \psi$ based on FF-dependence is defined as

$$\omega \diamond_{\mathsf{F}} \psi = \begin{cases} \{\omega\}, & \text{if } \omega \models \psi; \\ \{\omega_{\to t} \mid t \in \Omega_\psi\}, & \text{otherwise.} \end{cases}$$

If an interpretation $\omega$ satisfies the new information $\psi$, then we do not modify the interpretation. Otherwise, we force $\omega$ on each dependent minterm $t$ of $\psi$ such that the new interpretation $\omega_{\to t}$ satisfies $\psi$.

The FF-dependence based update of a formula $\phi$ by $\psi$ collects the updates of each model of $\phi$ by $\psi$:

**Proposition 13.** $[\![\phi \diamond_{\mathsf{F}} \psi]\!] = \bigcup_{\omega\in[\![\phi]\!]}(\omega \diamond_{\mathsf{F}} \psi)$.

Now, we give a representation result for the operator $\diamond_{\mathsf{F}}$.

**Theorem 1.** *An operator* $\diamond$: $\mathcal{L} \times \mathcal{L} \mapsto \mathcal{L}$ *is equal to* $\diamond_{\mathsf{F}}$ *iff it satisfies (U2), (U8), and*

**UP** *If* $DepVar(\phi) \cap DepVar(\psi) = \emptyset$, *then* $(\phi \wedge \phi') \diamond \psi \equiv \phi \wedge (\phi' \diamond \psi)$;

**UF** *If* $DepVar(\phi) \subseteq DepVar(\psi)$, $\phi \not\equiv \bot$, *and* $\phi \wedge \psi \equiv \bot$, *then* $\phi \diamond \psi \equiv \psi$.

*Proof.* ($\Rightarrow$): By Definition 14 and Proposition 13, $\diamond_{\mathsf{F}}$ satisfies (U2) and (U8). Suppose that $\mathrm{DepVar}(\phi) \cap \mathrm{DepVar}(\psi) = \emptyset$, by Proposition 9, we have $(\phi \wedge \phi') \diamond_{\mathsf{F}} \psi \equiv [\exists_{\mathsf{F}}(\neg\psi).(\phi \wedge \phi')] \wedge \psi \equiv \phi \wedge [\exists_{\mathsf{F}}(\neg\psi).\phi'] \wedge \psi \equiv \phi \wedge (\phi' \diamond_{\mathsf{F}} \psi)$. The above implies that $\diamond_{\mathsf{F}}$ satisfies (UP). Suppose that $\mathrm{DepVar}(\phi) \subseteq \mathrm{DepVar}(\psi)$, $\phi \not\equiv \bot$ and $\phi \wedge \psi \equiv \bot$. It is easily verified that $\phi|_{\neg\psi} \equiv \top$. Thus, $\phi\diamond\psi \equiv [\exists_{\mathsf{F}}(\neg\psi).\phi]\wedge\psi \equiv (\top\vee\phi)\wedge\psi \equiv \psi$. Hence, $\diamond_{\mathsf{F}}$ satisfies (UF).

($\Leftarrow$) : Suppose that $\diamond$ satisfies (U2), (U8), (UP) and (UF). By postulate (U8), $\phi \diamond \psi = \bigcup_{\omega\in[\![\phi]\!]} \omega \diamond \psi$. In the following, we only prove that, for any interpretation $\omega$, $\omega \diamond \psi = \omega \diamond_{\mathsf{F}} \psi$. It follows from (U2) that if $\omega \models \phi$, then $\omega \diamond \psi = \omega$.

Suppose that $\omega \not\models \phi$. Let $X = \mathrm{DepVar}(\psi)$. Based on $\mathrm{DepVar}(\psi)$ and $\omega$, we construct two terms $t$ and $t'$ as follows:

- $t = \bigwedge_{p\in X \text{ and } \omega\models p} p \wedge \bigwedge_{p\in X \text{ and } \omega\not\models p} \neg p$;
- $t' = \bigwedge_{p\in P\setminus X \text{ and } \omega\models p} p \wedge \bigwedge_{p\in P\setminus X \text{ and } \omega\not\models p} \neg p$.

Obviously, $t \wedge t'$ is the minterm corresponding to $\omega$. By postulates (UP) and (UF), we get that $(t \wedge t') \diamond \psi \equiv (t \diamond \psi) \wedge t' \equiv \psi \wedge t'$. So $\omega \diamond \psi = \{\omega_{\to t} \mid t \in \Omega_\psi\}$. □

Postulate (UP) is analogous to the postulate (SP) except use dependent variables instead of variables. It says that if the belief base can be divided into two disjoint compartments, then the compartment, which is not related to the new information, remains unchanged. Postulate (UF) means that if any dependent variable of $\phi$ is also a dependent variable of $\psi$, and $\phi$ is satisfiable and conflicts with $\psi$, then the belief is simply replaced by the new information after updating.

By Theorem 1 and Proposition 13, it is easily verified that $\diamond_{\mathsf{F}}$ satisfies some of KM postulates.

**Theorem 2.** *The operator* $\diamond_{\mathsf{F}}$ *satisfies (U1)-(U4) and (U8).*

It is easy to construct counterexamples to show that none of (U5)-(U7) are satisfied. Due to space limitations, we do not provide the counterexamples. The reason why the operator $\diamond_{\mathsf{F}}$ fails to satisfy them is that $\diamond_{\mathsf{F}}$ does not follow the principle of minimal change to which the three postulates correspond. From the semantic perspective, this principle requires that the update of $\omega$ by $\psi$ should be a set of models of $\psi$ that are the closest to $\omega$. However, $\omega \diamond_{\mathsf{F}} \psi$ generally involves some models that violate this condition. For example, consider $\omega = \{a, b, c\}$ and $\psi = \bar{a} \vee \bar{b}$. Then $\omega \diamond_{\mathsf{F}} \psi = \{\{a, c\}, \{b, c\}, \{c\}\}$. Clearly, both $\{a, c\}$ and $\{b, c\}$ are closer to $\omega$ than $\{c\}$, and hence $\{c\}$ is not the closest model.

Interestingly, when we restrict $\psi$ to be a satisfiable term, $\omega \diamond_f \psi$ contains only one model $\omega_{\to\psi}$. We take it for granted that it is the only model of $\psi$ that is the closest to $\omega$. Under this restriction, $\diamond_{\mathsf{F}}$ obeys the principle of minimal change, and hence satisfies (U5)-(U7).

**Theorem 3.** *If any formula* $\psi$ *appearing in* $\phi \diamond \psi$ *must be a term in the KM postulates, then the operator* $\diamond_{\mathsf{F}}$ *satisfies (U1)-(U8).*

To prove the theorem, we now give the following lemma which means that if the update of an interpretation $\omega$ by a term $t$ satisfies another one $t'$, then it is the same as the update of $\omega$ by the conjunction of $t$ and $t'$

**Lemma 1.** *Let $t$ and $t'$ are two terms. If $\omega \diamond_{\mathsf{F}} t \models t'$, then $\omega \diamond_{\mathsf{F}} t = \omega \diamond_{\mathsf{F}} (t \wedge t')$.*

We now prove Theorem 3.

*Proof.* **(U5)** Let $\omega$ be a model of $(\phi \diamond_{\mathsf{F}} \psi) \wedge \eta$. There is a model $\nu$ of $\phi$ s.t. $\omega = \nu \diamond_{\mathsf{F}} \psi$. Thus, $\nu \diamond_{\mathsf{F}} \psi \models \eta$. This, together with Lemma 1, imply that $\omega = \nu \diamond_{\mathsf{F}} (\psi \wedge \eta)$. Hence, $(\phi \diamond_{\mathsf{F}} \psi) \wedge \eta \models \phi \diamond_{\mathsf{F}} (\psi \wedge \eta)$.

**(U6)** Let $\omega$ be a model of $\phi$. By the assumption, we have $\omega \diamond_{\mathsf{F}} \psi \models \psi'$ and $\omega \diamond_{\mathsf{F}} \psi' \models \psi$. By Lemma 1, we get that $\omega \diamond_{\mathsf{F}} \psi = \omega \diamond_{\mathsf{F}} (\psi \wedge \psi') = \omega \diamond_{\mathsf{F}} \psi'$. Hence, $\phi \diamond_{\mathsf{F}} \psi \equiv \phi \diamond_{\mathsf{F}} \psi'$.

**(U7)** Let $\omega$ be a model of $(\phi \diamond_{\mathsf{F}} \psi) \wedge (\phi \diamond_{\mathsf{F}} \psi')$. Since $\phi$ is a minterm, there is a unique model $\nu$ of $\phi$. Because both $\psi$ and $\psi'$ are terms, so $\omega = \nu \diamond_{\mathsf{F}} \psi$ and $\omega = \nu \diamond_{\mathsf{F}} \psi'$. By postulate (U1), we get that $\omega \models \psi'$. This, together with Lemma 1, imply that $\omega = \nu \diamond_{\mathsf{F}} (\psi \wedge \psi')$. We construct a term $t$ as follows:

$$\bigwedge\nolimits_{p \in P \text{ and } (\psi \models p \text{ or } \psi' \models p)} p \wedge \bigwedge\nolimits_{p \in P \text{ and } (\psi \models \neg p \text{ or } \psi' \models \neg p)} \neg p.$$

It is easily verified that $t \equiv \psi \wedge \psi'$ and $t$ is a dependent term of $\psi \vee \psi'$. Hence, $\omega \diamond_{\mathsf{F}} (\psi \wedge \psi') \subseteq \omega \diamond_{\mathsf{F}} (\psi \vee \psi')$, and $(\phi \diamond_{\mathsf{F}} \psi) \wedge (\phi \diamond_{\mathsf{F}} \psi') \models \phi \diamond_{\mathsf{F}} (\psi \vee \psi')$.

□

From the postulational point of view, the essential difference between revision and update is as follows: revision satisfies the conjunction property (R2), proposed in (Katsuno and Mendelzon 1991): if the new information $\psi$ does not contradict the initial base $\phi$, then the revised belief base should be equivalent to the conjunction of $\phi$ and $\psi$. On the contrary, update satisfies the distribution property (U8): update is distributive over the initial base. The operator $\diamond_{\mathsf{F}}$ satisfies (U8) but not (R2). Hence, we consider $\diamond_{\mathsf{F}}$ as an update operator, not a revision operator.

We next give the upper and lower bounds of the inference problem of the update operator $\diamond_{\mathsf{F}}$.

**Proposition 14.** *Deciding whether $\phi \diamond_{\mathsf{F}} \psi \models \eta$ is in $\Delta_2^{\mathsf{P}}$ and* coNP*-hard.*

## Comparison with other update operators

In this subsection, we compare our belief update operator with other operators from different perspectives.

**Information preservation** The first perspective we focus on is how much information of the initial base is preserved after updating. From the semantic perspective, the less difference between the models of the initial base and those of the updated one, the more information that the update procedure preserves.

**Definition 15.** Let $\diamond$ and $\diamond'$ be two update operators. We say $\diamond$ preserves at least as much information as $\diamond'$, written $\diamond \leq_i \diamond'$, if $[\![\phi \diamond \psi]\!] \ominus [\![\phi]\!] \subseteq [\![\phi \diamond' \psi]\!] \ominus [\![\phi]\!]$ for any formulas $\phi$ and $\psi$. The notation $<_i$ is obtained as usual by taking the asymmetric parts of $\leq_i$.

Table 1: Comparison of various update operators

| Update operator | V/C | Min | Max | Avg |
|---|---|---|---|---|
| $\diamond_{\mathsf{F}}$ | 20/91 | 0.024 | 0.146 | 0.051 |
| | 50/218 | 0.072 | 92.681 | 1.201 |
| $\diamond_{\mathsf{V}}$ | 20/91 | 0.030 | 0.199 | 0.068 |
| | 50/218 | 0.070 | 91.647 | 1.206 |
| $\diamond_{\mathsf{L}}$ | 20/91 | 0.015 | 0.076 | 0.029 |
| | 50/218 | 0.027 | 64.218 | 0.164 |
| $\diamond_{\mathsf{For}}$ | 20/91 | 9.205 | 1,227 | 632.867 |
| | 50/218 | 9.555 | 1,350 | 857.459 |
| $\diamond_{\mathsf{PMA}}$ | 20/91 | 11.279 | 1,700 | 876.926 |
| | 50/218 | 11.461 | 1,895 | 1,225 |

The following proposition says that update operators based on dependence preserve more information than those based on the principle of minimal change. Among three dependence-based update operators, $\diamond_{\mathsf{F}}$ preserves the most information.

**Proposition 15.** $\diamond_{\mathsf{For}} <_i \diamond_{\mathsf{PMA}} <_i \diamond_{\mathsf{F}} <_i \diamond_{\mathsf{L}} <_i \diamond_{\mathsf{V}}$.

**Computational complexity** We next make a comparison from the perspective of the computational complexity. The complexity results of inference problems of $\diamond_{\mathsf{PMA}}$, $\diamond_{\mathsf{For}}$, $\diamond_{\mathsf{V}}$ and $\diamond_{\mathsf{L}}$ are as follows.

**Proposition 16.** *(Eiter and Gottlob 1992) Deciding whether $\phi \diamond_{\mathsf{PMA}} \psi \models \eta$ (resp. $\phi \diamond_{\mathsf{For}} \psi \models \eta$) is $\Pi_2^{\mathsf{P}}$-complete.*

**Proposition 17.** *(Herzig and Rifi 1999; Herzig, Lang, and Marquis 2013) Deciding whether $\phi \diamond_{\mathsf{V}} \psi \models \eta$ (resp. $\phi \diamond_{\mathsf{L}} \psi \models \eta$) is* coNP*-complete.*

**Definition 16.** Let $\diamond$ and $\diamond'$ be two update operators. We say $\diamond$ is at least as computational complexity as $\diamond'$, written $\diamond \leq_c \diamond'$, iff the upper bound the complexity of $\diamond$ is contained in the lower bound of the complexity of $\diamond'$. The notations $<_c$ and $=_c$ are obtained by taking the asymmetric and symmetric part of $\leq_c$ respectively.

By Propositions 14, 16 and 17, we get the following corollary. Because the computational complexity of $\diamond_{\mathsf{V}}$ and $\diamond_{\mathsf{L}}$ are the same, so $\diamond_{\mathsf{V}} =_c \diamond_{\mathsf{L}}$ holds. Similarly, $\diamond_{\mathsf{PMA}} =_c \diamond_{\mathsf{For}}$ also holds. The lower bound of $\diamond_{\mathsf{F}}$, which is coNP, is the same as the upper bound of $\diamond_{\mathsf{L}}$. Thus, $\diamond_{\mathsf{L}} \leq_c \diamond_{\mathsf{F}}$ holds. However, we cannot get that $\diamond_{\mathsf{F}} \leq_c \diamond_{\mathsf{L}}$ since the lower bound of $\diamond_{\mathsf{L}}$ does not contain the upper bound of $\diamond_{\mathsf{F}}$, which is $\Delta_2^{\mathsf{P}}$. Hence, we only obtain that $\diamond_{\mathsf{L}} \leq_c \diamond_{\mathsf{F}}$. In addition, $\diamond_{\mathsf{F}} \leq_c \diamond_{\mathsf{PMA}}$ holds while $\diamond_{\mathsf{PMA}} \leq_c \diamond_{\mathsf{F}}$ does not. This is because the upper bound of $\diamond_{\mathsf{PMA}}$, which is $\Pi_2^P$, is not a subset of the lower bound of $\diamond_{\mathsf{F}}$ even if it is $\Delta_2^P$-complete. So $\diamond_{\mathsf{F}} <_c \diamond_{\mathsf{PMA}}$ holds. The computational complexity of $\diamond_{\mathsf{F}}$ is between those of $\diamond_{\mathsf{V}}/\diamond_{\mathsf{L}}$ and $\diamond_{\mathsf{PMA}}/\diamond_{\mathsf{For}}$.

**Corollary 2.** $\diamond_{\mathsf{V}} =_c \diamond_{\mathsf{L}} \leq_c \diamond_{\mathsf{F}} <_c \diamond_{\mathsf{PMA}} =_c \diamond_{\mathsf{For}}$.

**Empirical results** We have shown that the computational complexity of $\diamond_{\mathsf{F}}$ is higher than the other two dependence-based update operators in theory, but practice might be another matter. To assess the latter, we conduct an experiment of computing the new belief base. The benchmarks used in (Marchi, Bittencourt, and Perrussel 2010) is from SATLIB that is available in http://www.cs.ubc.ca/∼hoos/SATLIB/benchm.html. We test two scales of test-sets: 91 clauses with 20 variables and 218 clauses with 50 variables. Each test-set has 1000 instances. For each instance, we use its corresponding theory as the initial belief base, and the negation of the first 4 clauses as the new contradictory information. In this experiment, we use BDDs to represent the initial KB and the new information and compute the updated KB via some operations of BDDs.

We illustrate the computation of each update operator in the following. Based on Definitions 12 and 13 and Proposition 12, computing $\phi \diamond_{\mathsf{F}} \psi$ consists of (1) generating the set DepVar$(\neg\psi)$; (2) forgetting all variables of DepVar$(\neg\psi)$ in $\phi \wedge \neg\psi$; (3) conjoining $\exists_{\mathsf{F}}(\neg\psi).\phi$ with $\psi$. The computation of $\phi \diamond_{\mathsf{V}} \psi$ (resp. $\phi \diamond_{\mathsf{L}} \psi$) is similar to the above except conjoin $\exists_{\mathsf{V}}$DepVar$(\neg\psi).\phi$ (resp. $\exists_{\mathsf{L}}$DepLit$(\neg\psi).\phi$) with $\psi$. We implement $\diamond_{\mathsf{PMA}}$ and $\diamond_{\mathsf{For}}$ according to the approach proposed by Gorogiannis and Ryan (2002).

In Table 1, the update operator and the numbers of variables and clauses are reported in columns 1 and 2 respectively. Columns 3 - 5 indicate the minimum, maximum and average time (in ms) of updating the KB. We can make two observations from Table 1. Firstly, all of approaches based on dependence are much more efficient than anyone based on the principle of minimal change. Secondly, for three dependence-based update operators, the maximum updating times of three dependence-based update operators are less than 100ms while the average ones are less than 1.5ms in the benchmarks with 218 clauses and 50 variables. The difference can be negligible, and hence the updating times of them are almost the same.

We close this section by noting that $\diamond_{\mathsf{F}}$ is a suitable alternative to update operator. It is the dependence-based update operator preserving the most information. In practice, the computational efficiency of $\diamond_{\mathsf{F}}$ are almost the same as those via $\diamond_{\mathsf{V}}$ and $\diamond_{\mathsf{L}}$, albeit that theoretically the computational complexity of $\diamond_{\mathsf{F}}$ is higher than those of them. Compared to the two operators based on the principle of minimal change, it preserves less information, but it is much more efficient.

## Conservative extension via formula forgetting

Conservative extension plays a prominent role in AI and logics (Ghilardi, Lutz, and Wolter 2006; Lutz, Walther, and Wolter 2007; Jung et al. 2017). Generally speaking, a conservative extension is a supertheory of a theory that proves no new theorems about the language of the original theory. We next give a syntax-independent definition of conservative extension in terms of dependent variables.

**Definition 17.** We say that $\phi \wedge \psi$ is a *syntax-independent conservative extension* of $\phi$, if for every formula $\eta$ with DepVar$(\eta) \subseteq$ DepVar$(\phi)$, $\phi \wedge \psi \models \eta$ implies that $\phi \models \eta$.

The above definition is slightly different from the original version that is based on variables. Each syntax-independent conservative extension also is an original one, but the converse does not hold. For example, $\phi = a \wedge (b \vee \bar{b})$ and $\psi = b$. The formula $\phi \wedge \psi$ is not a conservative extension of $\phi$ since $\phi \wedge \psi \models \psi$ but $\phi \not\models \psi$. However, it is a syntax-independent conservative extension of $\phi$. This difference does not impede the widespread use of syntax-independent version. In many practical applications, the background KB and query are firstly simplified, *i.e.*, they merely contain their dependent variables (Levy, Fikes, and Sagiv 1997; Lang, Liberatore, and Marquis 2003). Based on this assumption, two definitions of conservative extension are the same.

Finally, we obtain that deciding if $\phi \wedge \psi$ is a syntax-independent conservative extension of $\phi$ can be reduced to determining if forgetting $\phi \wedge \psi$ in each dependent minterm of $\phi$ leads to a tautology.

**Theorem 4.** *$\phi \wedge \psi$ is a syntax-independent conservative extension of $\phi$ iff $\exists_{\mathsf{F}}(\phi \wedge \psi).t \equiv \top$ for every $t \in \Omega_\phi$.*

*Proof.* ($\Rightarrow$): Suppose that there is $t \in \Omega_\phi$ s.t. $\exists_{\mathsf{F}}(\phi \wedge \psi).t \not\equiv \top$. Let $\Omega'$ be the set $\{t \mid t \in \Omega_\phi$ and $\exists_{\mathsf{F}}(\phi \wedge \psi).t \not\equiv \top\}$. It is easy to verify that $t \wedge \psi \equiv \bot$ for $t \in \Omega'$. Hence, $\phi \wedge \psi \equiv [\bigvee_{t\in(\Omega_\phi\setminus\Omega')} \vee \bigvee_{t\in\Omega'}] \wedge \psi \equiv \bigvee_{t\in(\Omega_\phi\setminus\Omega')} \wedge\psi$. Let $\eta = \bigvee_{t\in(\Omega_\phi\setminus\Omega')} t$. Obviously, $\phi \wedge \psi \models \eta$ and $\phi \not\models \eta$. This contradicts the assumption.

($\Leftarrow$): Suppose that there is $\eta$ s.t. DepVar$(\eta) \subseteq$ DepVar$(\phi)$, $\phi \wedge \psi \models \eta$ and $\phi \not\models \eta$. Hence, there is $t \in \Omega_\phi$ s.t. $t \not\models \eta$. Since DepVar$(\eta) \subseteq$ DepVar$(\phi)$, we get that $t \models \neg\eta$. Because $\phi \wedge \psi \models \eta$, so $t \wedge \psi \models \eta$. Thus, $t \wedge \psi \equiv \bot$. We get that $\exists_{\mathsf{F}}(\phi \wedge \psi).t \not\equiv \top$. This contradicts the assumption. □

**Example 9.** *Let $\phi = (a\wedge b)\vee(\bar{a}\wedge\bar{b})$ and $\psi = (a\wedge\bar{d})\vee(\bar{b}\wedge c)$. Then, $\phi\wedge\psi \equiv (a\wedge b\wedge\bar{d})\vee(\bar{a}\wedge\bar{b}\wedge c)$ and $\Omega_\phi = \{a \wedge b, \bar{a} \wedge \bar{b}\}$. Thus, $\exists_{\mathsf{F}}(\phi\wedge\psi).(a\wedge b) \equiv \top$ and $\exists_{\mathsf{F}}(\phi\wedge\psi).(\bar{a}\wedge\bar{b}) \equiv \top$. So $\phi \wedge \psi$ is a syntax-independent conservative extension of $\phi$.*

## Related Work

Dependence is well-known as a fundamental concept in many fields of artificial intelligence, particularly belief change. Several authors axiomatized the notion of dependence by postulates, and connected it to belief contraction. Belief contraction, a type of belief change, which is removal of existing beliefs. Del Cerro and Herzig (1996) gave postulates for a dependence relation between formulas, and established the correspondence between the dependence relation and belief contraction. In (del Cerro and Herzig 1996), a belief state is represented as a belief set, *i.e.*, an infinite set of formulas closed under implication. Oveisi et al. (2017) pointed out that a belief base, which need not be deductively closed and is often finite, is a practical alternative for representing belief states. They also identified a similar connection between dependence and base contraction.

The main difference between our work and the above approaches to dependence relations between formulas, is that our dependence relation corresponds to formula forgetting while their relations correspond to belief contraction. Belief contraction gets rid of as little as possible from the initial belief $\phi$ in order that the new belief state does not entail $\psi$. By contrast, formula forgetting eliminates all parts of $\phi$ relevant to $\psi$ even if $\phi$ does not entail $\psi$.

The above works focus on axiomatizations of dependence. Besides, there are several works that define the notion of dependence by the ideas of language splitting and variable sharing. Parikh (1999) showed *the finest splitting theorem*, which says that any finite set $\Sigma$ of formulas has a unique finest splitting (*i.e.*, a partition $\{P_1, \ldots, P_n\}$ of $P$ that refines every other splitting of $\phi$). $\Sigma$ can be decomposed into a set $\{\phi_1, \ldots, \phi_n\}$ of formulas where every $\phi_i$ uses only variables of $P_i$. Based on this theorem, a formula $\psi$ is said to be *canonically cell-relevant* to $\eta$ w.r.t. $\Sigma$, if there is a cell $P_i$ of the finest splitting of $\Sigma$ such that $P_i \cap \mathrm{DepVar}(\psi) \neq \emptyset$ and $P_i \cap \mathrm{DepVar}(\eta) \neq \emptyset$. Parikh (1999) also proposed a postulate (P) of relevance for belief change, and showed that the postulate (P) is consistent with the basic AGM postulates for belief revision.

Kourousias and Makinson (2007) extended the above work to the infinite case. Later, Peppas et al. (2015) demonstrated that the revision operator, proposed by Dalal (1988), satisfied all AGM postulates as well as the stronger version (SP) of postulate (P).

Rodrigues (1997) proposed another definition of relevance, called *path-relevance*. A formula $\psi$ is said to be *path-relevant* to $\eta$ w.r.t. $\Sigma$, if there is a finite sequence $\xi_0, \cdots, \xi_{m+1}$ of formulas such that $\xi_0 = \psi$, $\xi_{m+1} = \eta$, $\xi_1, \cdots, \xi_m \in \Sigma$, and $\mathrm{Var}(\xi_i) \cap \mathrm{Var}(\xi_{i+1}) \neq \emptyset$ for $0 \leq i \leq m$. The notion of path-relevance is not syntax-independence. To repair this defect, Makinson (2009) proposed a new definition of path-relevance via language splitting and dependent variables, and proved that it is indeed equivalent to the notion of canonical cell-relevance. A formula $\psi$ is said to be *canonically path-relevant* to $\eta$ w.r.t. $\Sigma$, if there is a finite sequence $\xi_0, \cdots, \xi_{m+1}$ of formulas such that $\xi_0 = \psi$, $\xi_{m+1} = \eta$, $\xi_1, \cdots, \xi_m$ are cells of the finest splitting of $\Sigma$, and $\mathrm{DepVar}(\xi_i) \cap \mathrm{DepVar}(\xi_{i+1}) \neq \emptyset$ for $0 \leq i \leq m$.

The above works assume that the underlying logic contains classical propositional logic. Horn logic is a very useful fragment of propositional logic, whose satisfiability is tractable. For this logic, relevance-sensitive belief contraction and revision were investigated by Wu, Zhang, and Zhang (2011) and Delgrande and Peppas (2015) respectively.

In other areas of artificial intelligence, some authors proposed different definitions of dependence relation between formulas. To speed up inferences from large KBs, Levy, Fikes, and Sagiv (1997) proposed a proof-theoretic framework for analyzing which components of a KB $\Sigma$ are irrelevant to a query $\psi$. A formula $\phi$ of a given KB $\Sigma$ is irrelevant to a query $\psi$ if $\psi$ can be derived without exploiting $\phi$. In the area of knowledge acquisition and machine learning, *novelty* is an important dependence relation between formulas given a background KB (Greiner and Genesereth 1983). It is used to decide whether a fact is new to some concepts w.r.t. the background KB. Greiner and Genesereth (1983) first formalized semantics for this relation. Later, Marquis (1991) proposed two equivalent characterizations in terms of prime implicant and abduction.

## Conclusions

The focus of this paper is on dependence in propositional logic. The main contributions are as follows:

First of all, this paper sheds light on the theoretical underpinnings of dependence in propositional logic. We have generalized the notions of FL-dependence and literal forgetting to FF-dependence and formula forgetting respectively. We have provided several equivalent formulations of FF-dependence and formula forgetting. Furthermore, some properties of them have been given.

In addition, we have applied these notions in belief update and conservative extension. We have defined the update operator $\diamond_{\mathsf{F}}$ in terms of formula forgetting. We have completely characterized this operator by identifying some extra postulates and have assessed them against the KM postulates. Compared to the other dependence-based update operators $\diamond_{\mathsf{V}}$ and $\diamond_{\mathsf{L}}$, $\diamond_{\mathsf{F}}$ preserves more information, and has almost the same efficiency. In contrast to the operators based on the principle of minimal change, $\diamond_{\mathsf{F}}$ is much more efficient than them although it preserves less information. Finally, we show that conservative extension can be reduced to formula forgetting.

## Acknowledgments

We thank the anonymous reviewers for helpful comments. We are grateful to Andreas Herzig, Fangzhen Lin, Yongmei Liu, Quilin Qi, Kewen Wang, Yisong Wang, Zhe Wang, Heng Zhang, Xiaowang Zhang, Yi Zhou and Zhiqiang Zhuang for their helpful discussions on the paper. This work was partially supported by the Natural Science Foundation of China (Nos. 61463044, 61472369, 61572234, 61573386, 61603152 and 61703182), Natural Science Foundation of Guangdong Province (No. 2016A030313292), Guangdong Province Science and Technology Plan project (Nos. 2016B030305007 and 2017B010110011), Guangzhou Science and Technology Plan project (No. 705241369105), the Talent Introduction Foundation of Jinan University (Nos. 88016534 and 88016653), Guangxi Key Laboratory of Trusted Software (Nos. kx201604 and kx201606), the Fundamental Research Funds for the Central Universities (No. 11617347), and Sun Yat-sen University Cultivation Project (No. 16lgpy40). Liangda Fang is also affiliated to Guangxi Key Laboratory of Trusted Software, Guilin University of Electronic Technology, Guilin, China.

## References

Boutilier, C. 1994. Toward a logic for qualitative decision theory. In *Proceedings of the Fourth International Conference on Principles of Knowledge Representation and Reasoning (KR-1994)*, 75–86.

Bryant, R. E. 1992. Symbolic Boolean Manipulation with Ordered Binary-Decision Diagrams. *ACM Computing Surveys* 24(3):293–318.

Dalal, M. 1988. Investigations into a theory of knowledge base revision: Preliminary report. In *Proceedings of the Seventh National Conference on Artificial Intelligence (AAAI-1988)*, 475–479.

Darwiche, A. 1997. A logical notion of conditional independence: properties and applications. *Artificial Intelligence* 97(1):45–82.

Darwiche, A. 1998. Compiling Devices: A Structure-Based Approach. In *Proceedings of the Eighth International Joint Conference on Principles of Knowledge Representation and Reasoning (KR-1998)*, 156–166.

del Cerro, L. F., and Herzig, A. 1996. Belief change and dependence. In *Proceedings of the Sixth Conference on Theoretical Aspects of Rationality and Knowledge (TARK-VI)*, 147–161.

Delgrande, J. P., and Peppas, P. 2015. Belief revision in horn theories. *Artificial Intelligence* 218:1–22.

Doherty, P.; Łukaszewicz, W.; and Madalinska-Bugaj, E. 1998. The PMA and Relativizing Change for Action Update. In *Proceedings of the Sixth International Conference on Principles of Knowledge Representation and Reasoning (KR-1998)*, 258–269.

Eiter, T., and Gottlob, G. 1992. On the complexity of propositional knowledge base revision, updates, and counterfactuals. *Artificial Intelligence* 57(2-3):227–270.

Forbus, K. D. 1989. Introducing Actions into Qualitative Simulation. In *Proceedings of the Eleventh International Joint Conference on Artificial Intelligence (IJCAI-1989)*, 1273–1278.

Ghilardi, S.; Lutz, C.; and Wolter, F. 2006. Did I Damage My Ontology? A Case for Conservative Extensions in Description Logics. In *Proceedings of the Tenth International Conference on Principles of Knowledge Representation and Reasoning (KR-2006)*, 187–197.

Gorogiannis, N., and Ryan, M. D. 2002. Implementation of Belief Change Operators Using BDDs. *Studia Logica* 70(1):131–156.

Greiner, R., and Genesereth, M. R. 1983. What's New? A Semantic Definition of Novelty. In *Proceedings of the Eighth International Joint Conference on Artificial Intelligence (IJCAI-1983)*, 450–454.

Hegner, S. J. 1987. Specification and Implementation of Programs for Updating Incomplete Information Databases. In *Proceedings of the Sixth ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems (PODS-1987)*, 146–158.

Herzig, A., and Rifi, O. 1998. Update operations: a review. In *Proceedings of Thirteenth European Conference on Artificial Intelligence (ECAI-1998)*, 13–17.

Herzig, A., and Rifi, O. 1999. Propositional belief base update and minimal change. *Artificial Intelligence* 115(1):107–138.

Herzig, A.; Lang, J.; and Marquis, P. 2013. Propositional Update Operators Based on Formula/Literal Dependence. *ACM Transactions on Computational Logic* 14(3):24:1–24:31.

Jung, J. C.; Lutz, C.; Martel, M.; Schneider, T.; and Wolter, F. 2017. Conservative Extensions in Guarded and Two-Variable Fragments. In *Proceedings of the Forty-Fourth International Colloquium on Automata, Languages, and Programming (ICALP-2017)*, 108:1–108:14.

Katsuno, H., and Mendelzon, A. O. 1991. On the Difference between Updating a Knowledge Base and Revising it. In *Proceedings of the Second International Joint Conference on Principles of Knowledge Representation and Reasoning (KR-1991)*, 387–394.

Kautz, H.; McAllester, D.; and Selman, B. 1997. Exploiting Variable Dependency in Local Search. In *Proceedings of the Fifteenth International Joint Conference on Artificial Intelligence (IJCAI-1997)*.

Kourousias, G., and Makinson, D. 2007. Parallel interpolation, splitting, and relevance in belief change. *The Journal of Symbolic Logic* 72(03):994–1002.

Lakemeyer, G. 1997. Relevance from an epistemic perspective. *Artificial Intelligence* 97(1-2):137–167.

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

Levi, I. 1977. Subjunctives, dispositions and chances. *Synthese* 34:423–455.

Levy, A. Y.; Fikes, R. E.; and Sagiv, Y. 1997. Speeding up inferences using relevance reasoning: a formalism and algorithms. *Artificial Intelligence* 97(1):83–136.

Lin, F., and Reiter, R. 1997. How to progress a database. *Artificial Intelligence* 92(1-2):131–167.

Lutz, C.; Walther, D.; and Wolter, F. 2007. Conservative Extensions in Expressive Description Logics. In *Proceedings of the Twentieth International Joint Conference on Artificial Intelligence (IJCAI-2007)*, 453–458.

Makinson, D. 2009. Propositional relevance through letter-sharing. *Journal of Applied Logic* 7:377–387.

Marchi, J.; Bittencourt, G.; and Perrussel, L. 2010. Prime forms and minimal change in propositional belief bases. *Annals of Mathematics and Artificial Intelligence* 59(1):1–45.

Marquis, P. 1991. Novelty revisited. In *Proceesings of the Sixth International Syposium on Methodologies for Intelligent Systems (ISMIS-1991)*, 550–559.

Minato, S. 1993. Zero-Suppressed BDDs for Set Manipulation in Combinatorial Problems. In *Proceedings of the Thirtieth ACM/IEEE Design Automation Conference (DAC-1993)*, 272–277.

Oveisi, M.; Delgrande, J. P.; Pelletier, F. J.; and Popowich, F. 2017. Kernel contraction and base dependence. *Journal of Artificial Intelligence Research* 60:97–148.

Parikh, R. 1999. Beliefs, belief revision, and splitting languages. *Logic, language and computation* 2(96):266–278.

Peppas, P.; Williams, M.-A.; Chopra, S.; and Foo, N. 2015. Relevance in belief revision. *Artificial Intelligence* 229:126–138.

Rodrigues, O. T. 1997. *A methodology for iterated information change*. Ph.D. Dissertation, Imperial College London.

Shannon, C. E. 1938. A symbolic analysis of relay and switching circuits. *Transactions of the American Institute of Electrical Engineers* 57(12):713–723.

Winslett, M. 1988. Reasoning about action using a possible models approach. In *Proceedings of the Seventh National Conference on Artificial Intelligence (AAAI-1988)*, 89–93.

Wu, M.; Zhang, D.; and Zhang, M. 2011. Language Splitting and Relevance-Based Belief Change in Horn Logic. In *Proceedings of the Twenty-Fifth AAAI Conference on Artificial Intelligence (AAAI-2011)*, 268–274.

Zhang, Y., and Zhou, Y. 2009. Knowledge forgetting: Properties and applications. *Artificial Intelligence* 173(16-17):1525–1537.