Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence (IJCAI 2015)

On Forgetting Postulates in Answer Set Programming

Jianmin Ji
University of Science and Technology of China, Hefei, China
jianmin@ustc.edu.cn

Jia-Huai You
University of Alberta
Edmonton, Canada
you@cs.ualberta.ca

Yisong Wang*
Guizhou University
Guiyang, China
csc.yswang@gzu.edu.cn

Abstract

Forgetting is an important mechanism for logic-based agent systems. A recent interest has been in the desirable properties of forgetting in answer set programming (ASP) and their impact on the design of forgetting operators. It is known that some subsets of these properties are incompatible, *i.e.*, they cannot be satisfied at the same time. In this paper, we are interested in the question on the largest set Δ of pairs (Π, V), where Π is a logic program and V is a set of atoms, such that a forgetting operator exists that satisfies all the desirable properties for each (Π, V) in Δ. We answer this question positively by discovering the precise condition under which the knowledge forgetting, a well-established approach to forgetting in ASP, satisfies the property of strong persistence, which leads to a sufficient and necessary condition for a forgetting operator to satisfy all the desirable properties proposed in the literature. We explore computational complexities on checking the condition and present a syntactic characterization which can serve as the basis of computing knowledge forgetting in ASP.

Introduction

It has been well argued that for cognitive robotics the ability of eliminating or hiding irrelevant symbols in a knowledge base, known as *(variable) forgetting*, plays an important role in logic-based agent systems [Lin and Reiter, 1994]. In simple words, forgetting is a process on a logical formula that replaces some logic symbols by *true* on the one hand and by *false* on the other, to produce a formula that no longer contains these symbols. Forgetting has found several interesting applications in Artificial Intelligence, such as regression and progression in databases and planning [Lin and Reiter, 1997; Liu and Wen, 2011; Rajaratnam *et al.*, 2014], abduction and diagnosis [Lin, 2001], conflict resolution [Lang and Marquis, 2010], and abstracting and comparing ontologies [Wang *et al.*, 2010; Konev *et al.*, 2012].

Logic programming under stable model (or answer set) semantics [Gelfond and Lifschitz, 1988; Ferraris, 2005], commonly referred to as *Answer Set Programming* (ASP), is a paradigm for declarative problem solving [Baral, 2003]. In ASP, various notions of equivalence have been proposed: (standard) equivalence, strong equivalence [Lifschitz *et al.*, 2001], uniform equivalence [Eiter *et al.*, 2007], and modular equivalence [Janhunen *et al.*, 2009]. Among these, the first two are most relevant in this paper. Informally, given two logic program Π and Π', they are *equivalent* if they have the same answer sets; they are *strongly equivalent* if $\Pi \cup \Sigma$ and $\Pi' \cup \Sigma$ have the same answer sets for every logic program Σ. The latter allows for "equivalent replacement" in ASP, and can thus be used to simplify logic programs [Lifschitz *et al.*, 2001]. The notion of strong equivalence can be characterized in the logic *here-and-there* (HT) [Pearce, 1996]. Since HT is a monotonic logic, ASP admits a monotonic entailment relation, written $\models_{\text{HT}}$, between logic programs by regarding a logic program as a logical formula.

Recently, researchers have shown a focused interest in forgetting in ASP [Delgrande and Wang, 2015], with a number of varying notions of forgetting proposed, such as the strong and weak forgetting [Zhang and Foo, 2006], the semantic forgetting [Eiter and Wang, 2008], forgetting operators F_W and F_S [Wong, 2009], the knowledge forgetting [Wang *et al.*, 2012; 2014], the SM-forgetting [Wang *et al.*, 2013], and the strong AS-forgetting [Knorr and Alferes, 2014]. Forgetting is also investigated for some nonmonotonic logical systems [Wang *et al.*, 2015]. In the above literature, several desirable properties have been formulated, which we briefly introduce below.

Let $\mathcal{L}$ be an ASP language on a signature $\mathcal{A}$, Π a logic program in $\mathcal{L}$, $V \subseteq \mathcal{A}$, and $\mathsf{f}(\Pi, V)$ the result of *forgetting about* V *in* Π. Let $\text{AS}(\Pi)$ denote the set of all answer sets of Π. The desirable properties about f can be described informally as follows:

- **(E)** Existence: $\mathsf{f}(\Pi, V)$ is expressible in $\mathcal{L}$.
- **(IR)** Irrelevance: $\mathsf{f}(\Pi, V)$ is irrelevant to V in terms of strong equivalence.
- **(W)** Weakening: $\Pi \models_{\text{HT}} \mathsf{f}(\Pi, V)$.
- **(PP)** Positive Persistence: if $\Pi \models_{\text{HT}} \Pi'$ and Π' is irrelevant to V then $\mathsf{f}(\Pi, V) \models_{\text{HT}} \Pi'$.
- **(NP)** Negative Persistence: if $\Pi \not\models_{\text{HT}} \Pi'$ and Π' is irrelevant to V then $\mathsf{f}(\Pi, V) \not\models_{\text{HT}} \Pi'$.

*Corresponding Author.

(SE) Strong Equivalence: If Π and Π' are strongly equivalent, then $\mathsf{f}(\Pi, V)$ and $\mathsf{f}(\Pi', V)$ are strongly equivalent.

(CP) Consequence Persistence: $\mathrm{AS}(\mathsf{f}(\Pi, V)) = \{M \setminus V \mid M \in \mathrm{AS}(\Pi)\}$.

(SP) Strong Persistence: $\mathrm{AS}(\mathsf{f}(\Pi, V) \cup \Pi') = \{M \setminus V \mid M \in \mathrm{AS}(\Pi \cup \Pi')\}$ for all programs Π' over signature $\mathcal{A} \setminus V$.

The intended meanings of the first seven properties are easy to understand. For instance, the property **(IR)** requires that the forgetting result $\mathsf{f}(\Pi, V)$ is strongly equivalent to a logic program containing no variable from V. The property **(SP)** says that the result of forgetting should preserve all the semantic dependencies contained in the original program, for all but the atom(s) to be forgotten [Knorr and Alferes, 2014]. It is evident that **(CP)** is a special case of **(SP)**. It has been shown that these properties together are inconsistent, in the sense that if f satisfies **(IR)**, **(E)** and **(CP)** then it violates **(W)** [Wang *et al.*, 2013].

The first four properties, i.e., **(W)**, **(PP)**, **(NP)** and **(IR)**, were proposed by Zhang and Zhou (2009) for knowledge forgetting in modal logic S5. The property **(CP)** was originally proposed by Eiter and Wang (2008) for a semantical notion of forgetting in ASP, which satisfies **(E)** and **(IR)**, but none of **(W)**, **(PP)**, **(NP)**, **(SE)**, and **(SP)**. Wang *et al.* (2012) adapted **(W)**, **(PP)**, **(NP)**, and **(IR)** for knowledge forgetting in ASP, which satisfies both **(E)** and **(SE)**, but fails for **(CP)** and **(SP)**. Later, Wang *et al.* (2013) proposed SM-forgetting in ASP, which satisfies **(E)**, **(IR)**, **(SE)**, **(CP)**, and **(PP)**, but none of **(W)**, **(NP)**, and **(SP)**. Knorr and Alferes (2014) proposed strong AS-forgetting, which satisfies **(IR)**, **(SE)**, **(CP)**, and **(SP)**, but not **(E)** in general.

In this paper, with the focus on the knowledge forgetting operator in propositional ASP – $\mathsf{Forget}_{\mathrm{HT}}$ – which is known to enjoy the first six properties [Wang *et al.*, 2012; 2014], we investigate possible restrictions for a logic program and a set of forgotten variables under which $\mathsf{Forget}_{\mathrm{HT}}$ also satisfies **(SP)**. This allows us to explore syntactically restricted subclasses of logic programs for which $\mathsf{Forget}_{\mathrm{HT}}$ enjoys all of the well-recognized properties.

In addition, as knowledge forgetting is defined semantically [Wang *et al.*, 2012; 2014] and, to our knowledge, there have been no syntactic characterizations for it, we propose a syntax-based approach for knowledge forgetting, which can be used as a syntactic transformation to compute knowledge forgetting.

The main contributions of the paper are as follows:

- We identify a sufficient and necessary condition for a logic program Π and a set of atoms V for which $\mathsf{Forget}_{\mathrm{HT}}$ satisfies the property **(SP)**, i.e. $\mathrm{AS}(\mathsf{Forget}_{\mathrm{HT}}(\Pi, V) \cup \Pi') = \{M \setminus V \mid M \in \mathrm{AS}(\Pi \cup \Pi')\}$ for every logic program Π' containing no atom from V. This implies that we have found the largest set Δ of pairs (Π, V) such that $\mathsf{Forget}_{\mathrm{HT}}$ satisfies all the properties under Δ (see Definition 2). We also study the complexity on checking whether a logic program and a set of atoms satisfy the condition.
- We obtain a syntactic counterpart of the (semantics-based) knowledge forgetting in ASP. It is substantially different from the syntactic definition for the forgetting in classical propositional logic.

The rest of the paper is organized as follows. Section briefly reviews the necessary concepts about ASP, the logic here-and-there, more details on desirable properties, and knowledge forgetting for ASP. In Section we show a sufficient and necessary condition for knowledge forgetting that satisfies the property **(SP)**, and study the computational complexities on checking the condition. Section presents a syntactic approach for knowledge forgetting. Finally, Section provides concluding remarks along with future directions.

Preliminaries

We assume a propositional language $\mathcal{L}_{\mathcal{A}}$ over a finite set $\mathcal{A}$ of propositional variables (*atoms*), called the *signature* of $\mathcal{L}_{\mathcal{A}}$. The *formulas* of $\mathcal{L}_{\mathcal{A}}$ are inductively constructed using connectives $\bot, \wedge, \vee$ and $\supset$ as the following:

$$\varphi ::= \bot \mid p \mid \varphi \vee \varphi \mid \varphi \wedge \varphi \mid \varphi \supset \varphi \qquad (1)$$

where $p \in \mathcal{A}$. The formula $\neg\varphi$ stands for $\varphi \supset \bot$, while $\top$ for $\bot \supset \bot$. We identify an interpretation with the set of atoms satisfied by it. A set $X \subseteq \mathcal{A}$ is a *model* of a formula φ, written $X \models \varphi$, if X satisfies φ in the sense of classical propositional logic. By $\mathsf{Mod}(\varphi)$ we denote the set of models of φ. A formula φ is *irrelevant* to $V \subseteq \mathcal{A}$, written $\mathsf{IR}(\varphi, V)$, if there is a formula ψ mentioning no atoms from V such that $\mathsf{Mod}(\varphi) = \mathsf{Mod}(\psi)$, *i.e.*, φ is *equivalent to* ψ.

In the following we recall the basic notions about answer sets of propositional formulas [Ferraris, 2005], the logic of here-and-there [Heyting, 1930; Pearce *et al.*, 2009], and the knowledge forgetting (HT-forgetting) for ASP [Wang *et al.*, 2012; 2014].

Answer sets

Let φ, ψ be two formulas and $X \subseteq \mathcal{A}$. The *reduct* of φ relative to X, written φ^X, is defined recursively as follows:

- if $X \not\models \varphi$ then $\varphi^X = \bot$,
- if $X \models p$ then $p^X = p$, and
- if $X \models \varphi \otimes \psi$ then $(\varphi \otimes \psi)^X = \varphi^X \otimes \psi^X$

where $p \in \mathcal{A}$ and $\otimes \in \{\vee, \wedge, \supset\}$. Intuitively, φ^X stands for the formula obtained from φ by replacing every outermost subformula not satisfied by X with $\bot$. The set X is an *answer set* of φ if it is a subset minimal model of φ^X. By $\mathrm{AS}(\varphi)$ we denote the set of answer sets of φ.

Under the answer set semantics, two formulas φ_1 and φ_2 are *equivalent*, denoted by $\varphi_1 \equiv_{AS} \varphi_2$, if they have the same answer sets, viz. $\mathrm{AS}(\varphi_1) = \mathrm{AS}(\varphi_2)$; φ_1 and φ_2 are *strongly equivalent*, denoted by $\varphi_1 \equiv^s_{AS} \varphi_2$, if $\varphi_1 \wedge \psi$ and $\varphi_2 \wedge \psi$ have the same answer sets for every formula ψ, viz. $\mathrm{AS}(\varphi_1 \wedge \psi) = \mathrm{AS}(\varphi_2 \wedge \psi)$ for every formula ψ. A formula φ is *AS-irrelevant* to $V \subseteq \mathcal{A}$, written $\mathsf{IR}_{\mathrm{AS}}(\varphi, V)$, if there exists a formula ψ mentioning no atoms from V such that $\varphi \equiv^s_{AS} \psi$.

A formula ψ is in *normal form*, if it is a conjunction of formulas (also called *rules*[1]) in the following form:

$$\bigwedge(B \cup \neg C \cup \neg\neg D) \supset \bigvee A \tag{2}$$

where A, B, C, D are sets of atoms, and we use the notation, for any $S \subseteq \mathcal{A}$, $\neg S = \{\neg p \mid p \in S\}$ and $\neg\neg S = \{\neg\neg p \mid p \in S\}$. Every formula φ can be translated to a formula ψ in the normal form such that $\varphi \equiv^s_{AS} \psi$ (cf. Theorem 2 of [Cabalar and Ferraris, 2007]).

A *logic program* is a finite set of rules. In the following, we identify a logic program Π with the formula $\bigwedge \Pi$ unless stated otherwise explicitly.

HT-models

An HT-*interpretation* is a pair $\langle X, Y\rangle$ such that $X \subseteq Y \subseteq \mathcal{A}$. That an HT-interpretation $\langle X, Y\rangle$ *satisfies* a formula φ, written $\langle X, Y\rangle \models_{\text{HT}} \varphi$, is defined recursively as follows

- for any atom p, $\langle X, Y\rangle \models_{\text{HT}} p$ if $p \in X$,
- $\langle X, Y\rangle \not\models_{\text{HT}} \bot$,
- $\langle X, Y\rangle \models_{\text{HT}} \varphi \wedge \psi$ if $\langle X, Y\rangle \models_{\text{HT}} \varphi$ and $\langle X, Y\rangle \models_{\text{HT}} \psi$,
- $\langle X, Y\rangle \models_{\text{HT}} \varphi \vee \psi$ if $\langle X, Y\rangle \models_{\text{HT}} \varphi$ or $\langle X, Y\rangle \models_{\text{HT}} \psi$,
- $\langle X, Y\rangle \models_{\text{HT}} \varphi \supset \psi$ if (i) $\langle X, Y\rangle \models_{\text{HT}} \varphi$ implies $\langle X, Y\rangle \models_{\text{HT}} \psi$, and (ii) $Y \models \varphi \supset \psi$.

An HT-interpretation $\langle X, Y\rangle$ is an HT-*model* of a formula φ if $\langle X, Y\rangle \models_{\text{HT}} \varphi$. Two formulas φ and ψ are HT-*equivalent*, denoted by $\varphi \equiv_{\text{HT}} \psi$, if they have the same HT-models; φ HT-*entails* ψ, denoted by $\varphi \models_{\text{HT}} \psi$, if every HT-model of φ is also an HT-model of ψ. An HT-model $\langle Y, Y\rangle$ of φ is an *equilibrium model* of φ if there is no $X \subseteq \mathcal{A}$ such that $X \subset Y$ and $\langle X, Y\rangle \models_{\text{HT}} \varphi$.

As shown in [Lifschitz *et al.*, 2001; Ferraris, 2005; Cabalar and Ferraris, 2007], the notion of HT-model is closely related to answer set.

Proposition 1 *Let φ, ψ be two formulas and $X \subseteq Y \subseteq \mathcal{A}$.*

(i) $\langle X, Y\rangle \models_{\text{HT}} \varphi$ *iff* $X \models \varphi^Y$.

(ii) *X is an answer set of φ iff $\langle X, X\rangle$ is an equilibrium model of φ.*

(iii) $\varphi \equiv^s_{AS} \psi$ *iff* $\varphi \equiv_{\text{HT}} \psi$.

Postulates for forgetting in ASP

We recall the desirable properties (postulates) for forgetting in ASP as introduced in [Eiter and Wang, 2008; Wong, 2009; Wang *et al.*, 2012; 2013; Knorr and Alferes, 2014].

Let φ be a formula and $V \subseteq \mathcal{A}$, and f a forgetting operator in ASP, *i.e.*, a formula $\psi = \mathsf{f}(\varphi, V)$ is the result of forgetting V in φ. For $\mathcal{S} \subseteq 2^{\mathcal{A}}$, we denote by $\mathcal{S}_{\|V}$ the set $\{S \setminus V \mid S \in \mathcal{S}\}$. The desirable properties are formally defined as follows:

(E) Existence: ψ is expressible in $\mathcal{L}_{\mathcal{A}}$.

(W) Weakening: $\varphi \models_{\text{HT}} \psi$.

(IR) Irrelevance: $\mathsf{IR}_{\text{AS}}(\psi, V)$.

(PP) Positive Persistence: for any formula ϕ, if $\mathsf{IR}_{\text{AS}}(\phi, V)$ and $\varphi \models_{\text{HT}} \phi$ then $\psi \models_{\text{HT}} \phi$.

(NP) Negative Persistence: for any formula ϕ, if $\mathsf{IR}_{\text{AS}}(\phi, V)$ and $\varphi \not\models_{\text{HT}} \phi$ then $\psi \not\models_{\text{HT}} \phi$.

(CP) Consequence Persistence: $\text{AS}(\psi) = \text{AS}(\varphi)_{\|V}$.

(SE) Strong Equivalence: for any formula ϕ, if $\varphi \equiv^s_{AS} \phi$ then $\mathsf{f}(\varphi, V) \equiv^s_{AS} \mathsf{f}(\phi, V)$.

(SP) Strong Persistence: for any formula ϕ, if $\mathsf{IR}_{\text{AS}}(\phi, V)$ then $\text{AS}(\psi \wedge \phi) = \text{AS}(\varphi \wedge \phi)_{\|V}$.

For every property above, we say that the operator f *satisfies* the property, if for every formula φ and every $V \subseteq \mathcal{A}$, the property holds for f. For instance, if $\text{AS}(\varphi)_{\|V} = \text{AS}(\mathsf{f}(\varphi, V))$ for each formula φ and $V \subseteq \mathcal{A}$ then **(CP)** holds for f, viz., f satisfies **(CP)**.

The first seven properties can be understood easily. For example, the property **(CP)** requires that the answer sets of the forgetting result ψ are exactly the ones of the original formula φ by discarding the forgotten atoms V. The property **(SP)** says that the result of forgetting should preserve all the semantic dependencies contained in the original formula, for all but the atom(s) to be forgotten [Knorr and Alferes, 2014]. The property **(CP)** is a special case of **(SP)**. Note that if f satisfies **(IR)**, **(E)** and **(CP)**, it will violate **(W)** (see Proposition 3). Thus, these properties together are inconsistent.

The next two propositions clarify the relationships among these properties.

Proposition 2 *Given a forgetting operator f in ASP:*

(i) if it satisfies **(SP)** *then it satisfies* **(CP)***;*

(ii) if it satisfies **(W)** *then it satisfies* **(NP)***;*

(iii) if it satisfies both **(IR)** *and* **(NP)** *then it satisfies* **(W)***;*

(iv) if it satisfies both **(IR)** *and* **(SP)** *then it satisfies* **(PP)** *and* **(SE)***.*

Proof sketch: (iv) For any $\langle X, Y\rangle \models_{\text{HT}} \mathsf{f}(\varphi, V)$ with $Y \cap V = \emptyset$, we can construct formula $\phi_1 = \bigwedge Y$ and $\phi_2 = \bigwedge X \wedge \bigwedge_{p,q \in Y \setminus X} p \supset q$. Then Y is the only answer set of $\mathsf{f}(\varphi, V) \wedge \phi_1$ and $X \subset Y$ implies Y is not an answer set of $\mathsf{f}(\varphi, V) \wedge \phi_2$. From **(SP)**, there exists $\langle X^*, Y^*\rangle \models_{\text{HT}} \varphi$ with $X^* \setminus V = X$ and $Y^* \setminus V = Y$. These results imply that **(PP)** and **(SE)** should be satisfied. ■

Proposition 3 (Proposition 3 in [Wang *et al.*, 2013]) *There is no forgetting operator in ASP that satisfies* **(W)** *or* **(NP)** *while it also satisfies* **(IR)***,* **(E)** *and* **(CP)***.*

The next two corollaries follow from Propositions 2 and 3.

Corollary 1 *Given a forgetting operator f in ASP, if it satisfies* **(W)***,* **(IR)***,* **(SP)** *and* **(E)***, then it satisfies the rest of the properties, i.e.,* **(PP)***,* **(NP)***,* **(SE)***, and* **(CP)***.*

Corollary 2 *There is no forgetting operator in ASP that satisfies* **(W)***,* **(IR)***,* **(SP)***, and* **(E)***.*

This motivates the following definition which allows us to talk about restrictions on the domain of a forgetting operator.

[1]A rule of the form (2) is written as $\bigvee A \leftarrow \bigwedge(B \cup \mathit{not}\, C \cup \mathit{not}\,\mathit{not}\, D)$ in [Lifschitz *et al.*, 1999], where $\mathit{not}\, S = \{\mathit{not}\, p \mid p \in S\}$ and $\mathit{not}\,\mathit{not}\, S = \{\mathit{not}\,\mathit{not}\, p \mid p \in S\}$.

Definition 1 *Let Δ be a set of pairs (φ, V) where φ is a formula and $V \subseteq \mathcal{A}$, and* f *a forgetting operator. The operator* f satisfies a property under Δ *if* $\mathsf{f}(\varphi, V)$ *has the corresponding property for every $(\varphi, V) \in \Delta$.*

For instance, we say that f *satisfies* (***SE***) *under* Δ if $\mathsf{f}(\varphi, V) \equiv^s_{AS} \mathsf{f}(\varphi', V)$ for every (φ, V) and (φ', V) in Δ with $\varphi' \equiv^s_{AS} \varphi$; f *satisfies* (***CP***) *under* Δ if $\mathrm{AS}(\mathsf{f}(\varphi, V)) = \mathrm{AS}(\varphi)_{\|V}$ for every $(\varphi, V) \in \Delta$; f *satisfies (**SP**) under* Δ if $\mathrm{AS}(\mathsf{f}(\varphi, V) \wedge \psi) = \mathrm{AS}(\varphi \wedge \psi)_{\|V}$ for every $(\varphi, V) \in \Delta$ and every formula ψ with $\mathsf{IR}_{\mathrm{AS}}(\psi, V)$.

Note that when we say a forgetting operator f satisfies a desirable property, we mean that f satisfies the property under $\Delta^* = \{(\varphi, V) \mid \varphi$ is a formula of $\mathcal{L}_{\mathcal{A}}$, and $V \subseteq \mathcal{A}\}$.

By Proposition 3, there is no forgetting operator in ASP that can satisfy all the desirable properties under Δ^*. For this reason, we are interested in identifying the largest subset Δ of Δ^* such that there is a forgetting operator f in ASP satisfying all the desirable properties under Δ. We will show that this forgetting operator f is the knowledge forgetting one in ASP.

The knowledge forgetting

Let X, X', V be sets of atoms and φ a formula. We define $X \sim_V X'$ if $X \setminus V = X' \setminus V$. Given two HT-interpretations $\langle X, Y\rangle$ and $\langle X', Y'\rangle$, we define that $\langle X, Y\rangle \sim_V \langle X', Y'\rangle$ if $X \sim_V X'$ and $Y \sim_V Y'$.

Definition 2 (Knowledge forgetting) *A formula ψ is a result of* HT-forgetting $V \subseteq \mathcal{A}$ in *a formula φ if, $\langle X', Y'\rangle \models_{\mathrm{HT}} \psi$ whenever $\langle X', Y'\rangle \sim_V \langle X, Y\rangle$ for some $\langle X, Y\rangle \models_{\mathrm{HT}} \varphi$.*

The result of HT-forgetting always exists and is unique up to the strong equivalence in ASP. Let us denote it by $\mathsf{Forget}_{\mathrm{HT}}(\varphi, V)$. Namely, $\mathsf{Forget}_{\mathrm{HT}}$ is the knowledge forgetting operator in ASP. It has been shown in [Wang *et al.*, 2012] that the $\mathsf{Forget}_{\mathrm{HT}}$ operator can be characterized precisely in terms of the properties (**W**), (**PP**), (**NP**) and (**IR**).

Proposition 4 (Theorem 3 in [Wang *et al.*, 2012]) *Let φ be a formula, $V \subseteq \mathcal{A}$, and* f *a forgetting operator. Then* $\mathsf{f}(\varphi, V) \equiv^s_{AS} \mathsf{Forget}_{\mathrm{HT}}(\varphi, V)$ *iff* f *satisfies the properties (**W**),* (**PP**)*,* (**NP**)*, and* (**IR**).

It can be shown that $\mathsf{Forget}_{\mathrm{HT}}$ also satisfies **(E)** and **(SE)**. They are actually implied by (**W**), (**PP**), (**NP**), and (**IR**). We therefore have the following proposition.

Proposition 5 *Let Δ be a set of pairs (φ, V) where φ is a formula and $V \subseteq \mathcal{A}$. The following statements (i) and (ii) are equivalent to each other.*

(i) There exists a forgetting operator f *in ASP satisfying all eight properties under Δ.*

(ii) $AS(\varphi \wedge \phi)_{\|V} = AS(\mathsf{Forget}_{\mathrm{HT}}(\varphi, V) \wedge \phi)$ for every $(\varphi, V) \in \Delta$ and every formula ϕ with $\mathsf{IR}_{AS}(\phi, V)$, i.e., $\mathsf{Forget}_{\mathrm{HT}}$ *satisfies the property* **(SP)** *under* Δ.

From the above proposition, given a set $\Delta \subseteq \Delta^*$, the problem of deciding whether there exists a forgetting operator that satisfies all desirable properties under Δ is equivalent to the problem of deciding whether $\mathsf{Forget}_{\mathrm{HT}}$ satisfies **(SP)** under Δ.

A Sufficient and Necessary Condition

As mentioned earlier, there is no forgetting operator in ASP that satisfies all the eight desirable properties under Δ^*. In this section, we identify a sufficient and necessary condition under which the HT-forgetting satisfies the property (**SP**).

HT-forgetting an atom

In the following, we write a single set $\{\alpha\}$ as α when it is clear from its context, for convenience. For example, we write $\mathsf{Forget}_{\mathrm{HT}}(\varphi, p)$ for $\mathsf{Forget}_{\mathrm{HT}}(\varphi, \{p\})$, $\mathsf{IR}_{\mathrm{AS}}(\varphi, p)$ for $\mathsf{IR}_{\mathrm{AS}}(\varphi, \{p\})$, $\mathcal{S}_{\|p}$ for $\mathcal{S}_{\|\{p\}}$, and so on.

Proposition 6 *Let φ be a formula and $p \in \mathcal{A}$. It holds that $AS(\varphi \wedge \psi)_{\|p} \subseteq AS(\mathsf{Forget}_{\mathrm{HT}}(\varphi, p) \wedge \psi)$ for every formula ψ with $\mathsf{IR}_{AS}(\psi, p)$, iff, for any* HT*-model $\langle X, Y\rangle$ of φ with $X \subset Y$, the following conditions hold:*

(i) $\langle Y \setminus \{p\}, Y \setminus \{p\}\rangle \models_{\mathrm{HT}} \varphi$ implies $\langle X \setminus \{p\}, Y \setminus \{p\}\rangle \models_{\mathrm{HT}} \varphi$, and

(ii) $\langle Y \cup \{p\}, Y \cup \{p\}\rangle \models_{\mathrm{HT}} \varphi$ implies $\langle Y, Y \cup \{p\}\rangle \models_{\mathrm{HT}} \varphi$ or $\langle X, Y \cup \{p\}\rangle \models_{\mathrm{HT}} \varphi$.

Proof sketch: ($\Leftarrow$) This is easy to verify.

($\Rightarrow$) If (i) or (ii) is not satisfied then we can construct the following formula

$$\psi' = \bigwedge(X \setminus \{p\}) \wedge \bigwedge_{q, q' \in Y \setminus (X \cup \{p\})} (q \supset q').$$

One can verify that $\mathrm{AS}(\varphi \wedge \psi')_{\|p} \not\subseteq \mathrm{AS}(\mathsf{Forget}_{\mathrm{HT}}(\varphi, p) \wedge \psi')$. ∎

The intuition behind (i) and (ii) in the above proposition is as follows. If an HT-interpretation $\langle X, Y\rangle \models_{\mathrm{HT}} \varphi$ then $\langle X \setminus \{p\}, Y \setminus \{p\}\rangle \models_{\mathrm{HT}} \mathsf{Forget}_{\mathrm{HT}}(\varphi, p)$. Once $X \setminus \{p\} \subset Y \setminus \{p\}$, then there exists some formula ψ with $\mathsf{IR}_{\mathrm{AS}}(\psi, p)$ such that $Y \setminus \{p\}$ is not an answer set of $\mathsf{Forget}_{\mathrm{HT}}(\varphi, p) \wedge \psi$. Thus, the conditions are to ensure that neither $Y \cup \{p\}$ nor $Y \setminus \{p\}$ is an answer set of $\varphi \wedge \psi$.

Proposition 7 *Let φ be a formula and $p \in \mathcal{A}$. It holds that $AS(\mathsf{Forget}_{\mathrm{HT}}(\varphi, p) \wedge \psi) \subseteq AS(\varphi \wedge \psi)_{\|p}$ for every formula ψ with $\mathsf{IR}_{AS}(\psi, p)$, iff, for every $Y \subseteq \mathcal{A}$,*

(i) $\langle Y \setminus \{p\}, Y\rangle \models_{\mathrm{HT}} \varphi$ implies $\langle Y \setminus \{p\}, Y \setminus \{p\}\rangle \models_{\mathrm{HT}} \varphi$.

Proof sketch: ($\Leftarrow$) Easy to show.

($\Rightarrow$) If the condition (i) is not satisfied then we can construct the following formula

$$\psi' = \bigwedge(Y \setminus \{p\}).$$

One can verify that $Y \setminus \{p\} \in \mathrm{AS}(\mathsf{Forget}_{\mathrm{HT}}(\varphi, p) \wedge \psi')$ and neither $Y \setminus \{p\}$ nor $Y \cup \{p\}$ is an answer set of $\varphi \wedge \psi'$. ∎

Intuitively, the condition (i) in Proposition 7 is to avoid the case that $Y \setminus \{p\}$ is an answer set of $\mathsf{Forget}_{\mathrm{HT}}(\varphi, p) \wedge \psi$. If $\langle Y \setminus \{p\}, Y\rangle \models_{\mathrm{HT}} \varphi \wedge \psi$ and $\langle Y \setminus \{p\}, Y \setminus \{p\}\rangle \not\models_{\mathrm{HT}} \varphi \wedge \psi$ then neither $Y \setminus \{p\}$ nor $Y \cup \{p\}$ is an answer set of $\varphi \wedge \psi$.

From Propositions 6 and 7, when forgetting just one atom, we could identify a necessary and sufficient condition under which the HT-forgetting satisfies all of the desirable properties as indicated by the next theorem.

Theorem 3 *Let φ be a formula and $p \in \mathcal{A}$. The following statements (i) and (ii) are equivalent to each other.*

(i) $AS(\mathsf{Forget}_{\text{HT}}(\varphi, p) \wedge \psi) = AS(\varphi \wedge \psi)_{\|p}$ for every formula ψ with $\mathsf{IR}_{AS}(\psi, p)$.

(ii) for any HT*-model $\langle X, Y\rangle$ of φ with $X \subset Y$,*

(a) $\langle Y \setminus \{p\}, Y \setminus \{p\}\rangle \models_{\text{HT}} \varphi$ implies $\langle X \setminus \{p\}, Y \setminus \{p\}\rangle \models_{\text{HT}} \varphi$, and

(b) $\langle Y \cup \{p\}, Y \cup \{p\}\rangle \models_{\text{HT}} \varphi$ implies $\langle Y, Y \cup \{p\}\rangle \models_{\text{HT}} \varphi$ or $\langle X, Y \cup \{p\}\rangle \models_{\text{HT}} \varphi$,

(c) $\langle Y \setminus \{p\}, Y\rangle \models_{\text{HT}} \varphi$ implies $\langle Y \setminus \{p\}, Y \setminus \{p\}\rangle \models_{\text{HT}} \varphi$.

The following theorem shows that it is intractable to check whether the property (**SP**) holds for HT-forgetting when just one atom is forgotten.

Theorem 4 *Let φ be a formula and $p \in \mathcal{A}$. Each of the following decision problems is* co-NP*-complete.*

(i) Deciding whether $AS(\varphi \wedge \psi)_{\|p} \subseteq AS(\mathsf{Forget}_{\text{HT}}(\varphi, p) \wedge \psi)$ for every formula ψ with $\mathsf{IR}_{AS}(\psi, p)$.

(ii) Deciding whether $AS(\mathsf{Forget}_{\text{HT}}(\varphi, p) \wedge \psi) \subseteq AS(\varphi \wedge \psi)_{\|p}$ for every formula ψ with $\mathsf{IR}_{AS}(\psi, p)$.

(iii) Deciding whether $AS(\varphi \wedge \psi)_{\|p} = AS(\mathsf{Forget}_{\text{HT}}(\varphi, p) \wedge \psi)$ for every formula ψ with $\mathsf{IR}_{AS}(\psi, p)$.

Proof sketch: The memberships are easy. For hardness, let ϕ be a formula, $p \in \mathcal{A}$ but not occurring in ϕ, and $\varphi_1 = (\neg\phi \vee \neg\neg p \vee q) \wedge ((\neg q \supset \neg\phi) \vee \neg\neg p)$, $\varphi_2 = \neg\phi \vee \neg\neg p$, and $\varphi_3 = \varphi_1 \wedge \varphi_2$. We can show that, for every formula ψ with $\mathsf{IR}_{\text{AS}}(\psi, p)$, ϕ is unsatisfiable iff $\text{AS}(\varphi_1 \wedge \psi)_{\|p} \subseteq \text{AS}(\mathsf{Forget}_{\text{HT}}(\varphi_1, p) \wedge \psi)$ iff $\text{AS}(\mathsf{Forget}_{\text{HT}}(\varphi_2, p) \wedge \psi) \subseteq \text{AS}(\varphi_2 \wedge \psi)_{\|p}$ iff $\text{AS}(\varphi_3 \wedge \psi)_{\|p} = \text{AS}(\mathsf{Forget}_{\text{HT}}(\varphi_3, p) \wedge \psi)$. ■

HT-forgetting a set of atoms

We are now in the position to identify a sufficient and necessary condition under which the HT-forgetting satisfies the property (**SP**) in general.

Proposition 8 *Let φ be a formula and $V \subseteq \mathcal{A}$. It holds that, for every formula ψ with $\mathsf{IR}_{AS}(\psi, V)$, $AS(\varphi \wedge \psi)_{\|V} \subseteq AS(\mathsf{Forget}_{\text{HT}}(\varphi, V) \wedge \psi)$, iff, for each* HT*-model $\langle X, Y\rangle$ of φ with $X \setminus V \subset Y \setminus V$, if there exists a set Y' with $Y \setminus V \subseteq Y' \subseteq Y \cup V$ and $\langle Y', Y'\rangle \models_{\text{HT}} \varphi$, then*

(i) there exists a set Y'' with $Y \setminus V \subseteq Y'' \subset Y'$ such that $\langle Y'', Y'\rangle \models_{\text{HT}} \varphi$, or

(ii) there exists a set X' with $X' \subseteq Y'$ and $X' \setminus V = X \setminus V$ such that $\langle X', Y'\rangle \models_{\text{HT}} \varphi$.

Proof sketch: ($\Leftarrow$) Easy.

($\Rightarrow$) If both conditions (i) and (ii) are not satisfied then we can construct the formula

$$\psi' = \bigwedge(X \setminus V) \wedge \bigwedge_{q,q' \in Y \setminus (X \cup V)} (q \supset q').$$

One can verify that $\text{AS}(\varphi \wedge \psi')_{\|V} \not\subseteq \text{AS}(\mathsf{Forget}_{\text{HT}}(\varphi, V) \wedge \psi')$. ■

Intuitively, if $\langle Y', Y'\rangle \models_{\text{HT}} \varphi$ then either (i) or (ii) in the above proposition should be satisfied, which ensures that Y' cannot be an answer set of $\varphi \wedge \psi$ for some formula ψ with $\mathsf{IR}_{\text{AS}}(\psi, V)$.

Proposition 9 *Let φ be a formula and $V \subseteq \mathcal{A}$. It holds that, for every formula ψ with $\mathsf{IR}_{AS}(\psi, V)$, $AS(\mathsf{Forget}_{\text{HT}}(\varphi, V) \wedge \psi) \subseteq AS(\varphi \wedge \psi)_{\|V}$, iff, for each $Y \subseteq \mathcal{A}$, if there exists a set Y' with $Y \setminus V \subseteq Y' \subset Y$ and $\langle Y', Y\rangle \models_{\text{HT}} \varphi$, then*

(i) there exists a set Y'' with $Y \setminus V \subseteq Y'' \subseteq Y \cup V$ such that $\langle Y'', Y''\rangle \models_{\text{HT}} \varphi$ and there does not exist a set Y''' with $Y \setminus V \subseteq Y''' \subset Y''$ and $\langle Y''', Y''\rangle \models_{\text{HT}} \varphi$.

Proof sketch: ($\Leftarrow$) Easy.

($\Rightarrow$) If the condition (i) is not satisfied then we can construct the formula

$$\psi' = \bigwedge(Y \setminus V).$$

It is not difficult to verify that $Y \setminus V \in \text{AS}(\mathsf{Forget}_{\text{HT}}(\varphi, V) \wedge \psi')$ and there does not exist a set Y' with $Y' \sim_V Y$ such that $Y' \in \text{AS}(\varphi \wedge \psi')$. ■

Intuitively, the condition (i) in Proposition 9 is to avoid the case that $Y \setminus V$ is an answer set of $\mathsf{Forget}_{\text{HT}}(\varphi, V) \wedge \psi$ while there does not exist a set Y' with $Y' \sim_V Y$ and Y' is an answer set of $\varphi \wedge \psi$.

The next theorem follows from Propositions 8 and 9.

Theorem 5 (Main theorem) *Let φ be a formula and $V \subseteq \mathcal{A}$. The statements (i) and (ii) are equivalent to each other.*

(i) $AS(\mathsf{Forget}_{\text{HT}}(\varphi, V) \wedge \psi) = AS(\varphi \wedge \psi)_{\|V}$ for every formula ψ with $\mathsf{IR}_{AS}(\psi, V)$.

(ii) The following conditions (a) and (b) hold:

(a) for each HT*-model $\langle X, Y\rangle$ of φ with $X \setminus V \subset Y \setminus V$, if there exists a set Y' with $Y \setminus V \subseteq Y' \subseteq Y \cup V$ and $\langle Y', Y'\rangle \models_{\text{HT}} \varphi$, then*
- *there exists a set Y'' with $Y \setminus V \subseteq Y'' \subset Y'$ such that $\langle Y'', Y'\rangle \models_{\text{HT}} \varphi$, or*
- *there exists a set X' with $X' \subseteq Y'$ and $X' \setminus V = X \setminus V$ such that $\langle X', Y'\rangle \models_{\text{HT}} \varphi$;*

(b) for each $Y \subseteq \mathcal{A}$, if there exists a set Y' with $Y \setminus V \subseteq Y' \subset Y$ and $\langle Y', Y\rangle \models_{\text{HT}} \varphi$, then
- *there exists a set Y'' with $Y \setminus V \subseteq Y'' \subseteq Y \cup V$ such that $\langle Y'', Y''\rangle \models_{\text{HT}} \varphi$ and there does not exist a set Y''' with $Y \setminus V \subseteq Y''' \subset Y''$ and $\langle Y''', Y''\rangle \models_{\text{HT}} \varphi$.*

From Proposition 5, the condition (ii) in Theorem 5 specifies the largest set Δ of (φ, V) such that there exists a forgetting operator in ASP satisfying all eight properties under Δ. In the following, we use Δ° to denote the largest set.

The next example shows a possibility of $(\varphi, V) \notin \Delta^{\circ}$ even if V contains all atoms occurring in the formula φ.

Example 1 *Let $\mathcal{A} = \{p, q\}$, $\varphi = \neg p \supset p$, $V = \{p\}$ and $\psi = q$. One can check that $\varphi \wedge \psi$ has no answer set and $\mathsf{Forget}_{\text{HT}}(\varphi, V) = \top$. Thus, $\mathsf{Forget}_{\text{HT}}(\varphi, V) \wedge \psi$ has a unique answer set $\{q\}$. It follows that $AS(\varphi \wedge \psi)_{\|V} \neq AS(\mathsf{Forget}_{\text{HT}}(\varphi, V) \wedge \psi)$.*

Actually, one can further verify that $\langle \{p\}, \{p, q\}\rangle \models_{\text{HT}} \varphi$ and $\langle \{q\}, \{p, q\}\rangle \models_{\text{HT}} \varphi$, but $\langle \{q\}, \{q\}\rangle \not\models_{\text{HT}} \varphi$, i.e., the condition (c) in Theorem 3 does not hold.

Proposition 10 *Let Δ be a set of pairs (φ, V) where φ is a formula and $V \subseteq \mathcal{A}$. The statements (i) and (ii) are equivalent to each other.*

(i) There exists a forgetting operator f in ASP satisfying all either properties under Δ.

(ii) For each pair $(\varphi, V) \in \Delta$, φ and V satisfy the condition (ii) in Theorem 5.

The following theorem indicates that it is difficult to check whether $\mathsf{Forget}_{\text{HT}}$ satisfies (**SP**) in general.

Theorem 6 *Let φ be a formula and $V \subseteq \mathcal{A}$. Each of the following decision problems is Π_2^{P}-complete.*

(i) Deciding whether $AS(\varphi \wedge \psi)_{\|V} \subseteq AS(\mathsf{Forget}_{\text{HT}}(\varphi, V) \wedge \psi)$ for every formula ψ with $\mathsf{IR}_{AS}(\psi, V)$.

(ii) Deciding whether $AS(\mathsf{Forget}_{\text{HT}}(\varphi, V) \wedge \psi) \subseteq AS(\varphi \wedge \psi)_{\|V}$ for every formula ψ with $\mathsf{IR}_{AS}(\psi, V)$.

(iii) Deciding whether $AS(\varphi \wedge \psi)_{\|V} = AS(\mathsf{Forget}_{\text{HT}}(\varphi, V) \wedge \psi)$ for every formula ψ with $\mathsf{IR}_{AS}(\psi, V)$.

Proof sketch: The memberships are easy. The hardness follows from the following fact: Given two formulas ψ' and φ', the problem of deciding whether $\psi' \models_{\text{HT}} \mathsf{Forget}_{\text{HT}}(\varphi', V)$ is Π_2^{P}-complete (cf. Theorem 14 in [Wang *et al.*, 2014]). ■

Though it is in general difficult to verify if $(\varphi, V) \in \Delta^{\circ}$ for a formula φ and $V \subseteq \mathcal{A}$, there exist some trivial syntactic conditions as shown in the next proposition.

Proposition 11 *Let φ be a formula and $V \subseteq \mathcal{A}$. If $\varphi = \bigwedge(A \cup \neg B)$ then $(\varphi, V) \in \Delta^{\circ}$, where A, B are sets of atoms.*

Proof sketch: Let $A' = A \setminus V$ and $B' = B' \setminus V$ and ψ a formula with $\mathsf{IR}_{\mathrm{AS}}(\psi, V)$. Note that $\mathsf{Forget}_{\text{HT}}(\varphi, V) \equiv_{\text{HT}} \bigwedge(A' \cup \neg B')$. We can show that $\mathrm{AS}(\bigwedge(A \cup \neg B) \wedge \psi) = \mathrm{AS}(\bigwedge(A \cup \neg B) \wedge (\psi|_{A \to \top})|_{B \to \bot}) = \mathrm{AS}(\bigwedge(A \cup \neg B) \wedge (\psi|_{A' \to \top})|_{B' \to \bot})$ due to $\mathsf{IR}_{\mathrm{AS}}(\psi, V)$. Please see the next section for the definition of $\psi|_{V \to \star}$ for $\star \in \{\top, \bot\}$. ■

As mentioned in Introduction, knowledge forgetting based on the operator $\mathsf{Forget}_{\text{HT}}$ is defined semanticaly and no syntactic characterizations are known. As it is Π_2^{P}-complete to check whether $\psi \equiv_{\text{HT}} \mathsf{Forget}_{\text{HT}}(\varphi, V)$ holds for two given formulas φ and ψ and a set $V \subseteq \mathcal{A}$ [Wang *et al.*, 2014], it is intractable to compute the results of knowledge forgetting. In the next section, we present a syntactic approach for knowledge forgetting. Similar to the syntactic definition of forgetting in classical propositional logic, it may inevitably result in exponential explosion.

A Syntactic Approach

In this section, we provide a syntactic characterization of HT-forgetting and a corresponding algorithm for computing knowledge forgetting, for formulas in normal form.

First, we introduce some notations. Let φ be a formula and $p \in \mathcal{A}$. By $\varphi|_{p \to \star}$ we mean the formula obtained from φ by replacing every occurrence of the atom p by $\star$, where $\star \in \{\top, \bot\}$. Let $V = \{p_1, \ldots, p_n\} \subseteq \mathcal{A}$. By $\varphi|_{V \to \star}$ we denote the formula $(\cdots(\varphi|_{p_1 \to \star}) \cdots)|_{p_n \to \star}$. Please note that the forgetting in propositional logic can be syntactically defined as $\mathsf{Forget}(\varphi, p) = \varphi|_{p \to \top} \vee \varphi|_{p \to \bot}$ and $\mathsf{Forget}(\varphi, V \cup \{p\}) = \mathsf{Forget}(\mathsf{Forget}(\varphi, p), V)$ [Lang *et al.*, 2003].

Definition 3 *Let φ be a formula in normal form and $X \subseteq \mathcal{A}$. The* semi-reduct *of φ w.r.t. X, written φ_X, is the formula obtained from φ by replacing every occurrence of an atom $p \in X$ in the range of $\neg$ with $\top$.*

Please note that, φ_X is slightly different from the GL-reduction [Lifschitz *et al.*, 1999] in that the GL-reduction also handles the negative occurrence of the atoms not in X.

Example 2 *Consider the formula φ:*

$$(\neg p \supset p) \wedge (\neg\neg p \supset p) \wedge (\neg p \supset r) \wedge (\neg q \supset r) \wedge (\neg q \supset p).$$

Let $X = \{p\}$. Then φ_X is the formula:

$$(\neg\top \supset p) \wedge (\neg\neg\top \supset p) \wedge (\neg\top \supset r) \wedge (\neg q \supset r) \wedge (\neg q \supset p)$$

which is strongly equivalent to

$$p \wedge (\neg q \supset r) \wedge (\neg q \supset p).$$

It has a unique answer set $\{p, r\}$. One should note here that the GL-reduct of φ w.r.t.X is $\varphi^X = p \wedge r$ whose unique answer set is $\{p, r\}$. It is not difficult to verify that $\varphi^X \models_{\text{HT}} \varphi_X$, but not vice versa.

The next theorem identifies an alternative sufficient and necessary condition under which $\mathsf{Forget}_{\text{HT}}$ satisfies (**SP**) when forgetting one atom.

Theorem 7 *Let φ be a formula in normal form and p an atom. Then, $AS(\mathsf{Forget}_{\text{HT}}(\varphi, p) \wedge \psi) = AS(\varphi \wedge \psi)_{\|p}$ for every formula ψ with $\mathsf{IR}_{AS}(\psi, p)$, iff the following conditions hold:*

(a) $\varphi \wedge \neg\neg\varphi|_{p \to \bot} \models_{\text{HT}} \varphi|_{p \to \bot}$,

(b) $\varphi \wedge \neg\neg\varphi|_{p \to \top} \models_{\text{HT}} \varphi_{\{p\}}|_{p \to \bot} \vee \neg\neg(\varphi|_{p \to \top} \wedge \varphi_{\{p\}}|_{p \to \bot})$, *and*

(c) $\varphi|_{p \to \top} \wedge \varphi_{\{p\}}|_{p \to \bot} \models \varphi|_{p \to \bot}$.

Proof sketch: It is not difficult to verify that the condition (a) (resp. (b) and (c)) in the theorem is equivalent to the condition (a) (resp. (b) and (c)) in Theorem 3. ■

Proposition 12 *Let φ be a formula in normal form, $p \in \mathcal{A}$, and $\langle X, Y \rangle$ an* HT*-interpretation with $p \notin Y$. The following hold:*

(i) $\langle X, Y \rangle \models_{\text{HT}} \varphi$ *iff* $\langle X, Y \rangle \models_{\text{HT}} \varphi|_{p \to \bot}$;

(ii) $\langle X \cup \{p\}, Y \cup \{p\} \rangle \models_{\text{HT}} \varphi$ *iff* $\langle X, Y \rangle \models_{\text{HT}} \varphi|_{p \to \top}$;

(iii) $\langle X, Y \cup \{p\} \rangle \models_{\text{HT}} \varphi$ *or* $\langle X \cup \{p\}, Y \cup \{p\} \rangle \models_{\text{HT}} \varphi$ *iff* $\langle X, Y \rangle \models_{\text{HT}} (\varphi_{\{p\}}|_{p \to \bot} \vee \varphi_{\{p\}}|_{p \to \top}) \wedge \neg\neg\varphi|_{p \to \top}$.

Proof sketch: (i) and (ii) follows from the definition of HT-satisfiability. (iii) follows from the following properties

- $\langle X, Y \rangle \models_{\text{HT}} \neg\neg\varphi|_{p \to \top}$ iff $Y \cup \{p\} \models \varphi$;
- $\langle X, Y \cup \{p\} \rangle \models_{\text{HT}} \varphi$ implies $\langle X, Y \rangle \models_{\text{HT}} \varphi_{\{p\}}|_{p \to \bot} \vee \varphi_{\{p\}}|_{p \to \top}$;
- $\langle X, Y \rangle \models_{\text{HT}} \varphi_{\{p\}}|_{p \to \bot} \vee \varphi_{\{p\}}|_{p \to \top}$ and $Y \cup \{p\} \models \varphi$ implies $\langle X, Y \cup \{p\} \rangle \models_{\text{HT}} \varphi$ or $\langle X \cup \{p\}, Y \cup \{p\} \rangle \models_{\text{HT}} \varphi$. ■

Algorithm 1: HT-forgetting

input : A formula φ and $V \subseteq \mathcal{A}$
output: A result of HT-forgetting V in φ

$\varphi' \leftarrow$ the normal form of φ;
foreach $p \in V$ **do**
  $\varphi' \leftarrow \varphi|_{p\to\top} \vee \varphi|_{p\to\bot}$
    $\vee(\varphi_{\{p\}}|_{p\to\bot} \vee \varphi_{\{p\}}|_{p\to\top}) \wedge \neg\neg\varphi|_{p\to\top}$;
  $\varphi' \leftarrow$ the normal form of φ';
end
return φ';

Theorem 8 *Let φ be a formula in normal form and $p \in \mathcal{A}$. It holds that*

$$\mathsf{Forget}_{\text{HT}}(\varphi, p) \equiv^s_{AS} \varphi|_{p\to\top} \vee \varphi|_{p\to\bot} \vee ((\varphi_{\{p\}}|_{p\to\bot} \vee \varphi_{\{p\}}|_{p\to\top}) \wedge \neg\neg\varphi|_{p\to\top}).$$

Proof sketch: Let $\langle X, Y\rangle$ be an HT-interpretation with $p \notin Y$. Then $\langle X, Y\rangle \models_{\text{HT}} \mathsf{Forget}_{\text{HT}}(\varphi, p)$ iff $\langle X, Y\rangle \models_{\text{HT}} \varphi$, $\langle X, Y \cup \{p\}\rangle \models_{\text{HT}} \varphi$, or $\langle X \cup \{p\}, Y \cup \{p\}\rangle \models_{\text{HT}} \varphi$. Thus the claim follows from Proposition 12. ■

Recall that, for any formula φ, atom p, and $V \subseteq \mathcal{A}$, $\mathsf{Forget}_{\text{HT}}(\varphi, V \cup \{p\}) \equiv^s_{AS} \mathsf{Forget}_{\text{HT}}(\mathsf{Forget}_{\text{HT}}(\varphi, p), V)$ (cf. Corollary 7 of [Wang *et al.*, 2014]). Moreover, every formula φ can be translated to a formula ψ in normal form such that $\varphi \equiv^s_{AS} \psi$. Therefore the above theorem implies a syntactic approach to computing the result of HT-forgetting for a formula φ and $V \subseteq \mathcal{A}$. The details are given in Algorithm 1.

Corollary 9 *Algorithm 1 outputs* $\mathsf{Forget}_{\text{HT}}(\varphi, V)$.

Let NF(φ) be a formula in normal form strongly equivalent to the formula φ. The syntactic knowledge forgetting is formally defined below.

Definition 4 (Syntactic knowledge forgetting) *Let φ be a formula. We define:*

- $\mathsf{Forget}_{\text{HT}}(\varphi, p) = NF(\varphi)|_{p\to\top} \vee NF(\varphi)|_{p\to\bot} \vee ((NF(\varphi)_{\{p\}}|_{p\to\bot} \vee NF(\varphi)_{\{p\}}|_{p\to\top}) \wedge \neg\neg NF(\varphi)|_{p\to\top})$,
- $\mathsf{Forget}_{\text{HT}}(\varphi, \{p\} \cup V) = \mathsf{Forget}_{\text{HT}}(\mathsf{Forget}_{\text{HT}}(\varphi, p), V)$

where $p \in \mathcal{A}$ and $V \subseteq \mathcal{A}$.

Example 3 (Continued from Example 2) *Note that,*

$$\begin{aligned}\varphi|_{p\to\top} &\equiv^s_{AS} \neg q \supset r,\\ \varphi|_{p\to\bot} &\equiv^s_{AS} \bot,\\ \varphi_{\{p\}}|_{p\to\top} &\equiv^s_{AS} \neg q \supset r,\\ \varphi_{\{p\}}|_{p\to\bot} &\equiv^s_{AS} \bot.\end{aligned}$$

Then, $\mathsf{Forget}_{\text{HT}}(\varphi, p)$ *is strongly equivalent to $\neg q \supset r$. Its unique answer set is $\{r\}$.*

Concluding Remarks

Lately, the literature on forgetting has shown extensive interest in the desirable properties of forgetting operators in ASP. In this paper, we have identified a precise condition for a formula φ and $V \subseteq \mathcal{A}$ under which $\mathsf{Forget}_{\text{HT}}(\varphi, V)$ satisfies the property (**SP**). This leads to the largest set Δ° of pairs (φ, V) where $\varphi \in \mathcal{L}_\mathcal{A}$ and $V \subseteq \mathcal{A}$ such that $\mathsf{Forget}_{\text{HT}}$ enjoys all the eight properties under Δ°. This condition provides a guideline to explore subclasses of logic programs for which the HT-forgetting enjoys all of the desirable properties. Though a trivial subclass of logic programs occurring in Δ° is identified, cf. Proposition 11, it is still worthy to identify more interesting subclasses of logic programs for which the HT-forgetting enjoys (**SP**). It is also interesting to investigate the property (**SP**) for forgetting in other nonmonotonic logical systems, such as in circumscription [Wang *et al.*, 2015].

Secondly, we have proposed a syntactic approach to computing HT-forgetting results. Using this approach to compute the result of HT-forgetting a set $V \subseteq \mathcal{A}$ from a formula φ, one needs to compute the normal form of φ. This can be time consuming as an exponential explosion in the worst case is inevitable. To extend the syntactic approach for arbitrary formulas is a challenging future task.

Acknowledgement

We thank reviewers for their helpful comments. We are grateful to Fangzhen Lin for many helpful and informative discussions. We would also like to thank Xiaoping Chen and his research group for their useful discussions. Jianmin Ji's research was partially supported by the National Natural Science Foundation of China under grand 61175057, the National Natural Science Foundation for the Youth of China under grand 61403359, as well as the USTC Key Direction Project and the USTC 985 Project. Yisong Wang was partially supported by the National Natural Science Foundation of China under grand 61370161, the Stadholder Foundation of Guizhou Province under grant (2012)62 and the Natural Science Foundation of Guizhou Province under grant [2014]7640.

References

[Baral, 2003] Chitta Baral. *Knowledge representation, reasoning and declarative problem solving*. Cambridge university press, 2003.

[Bonet and Koenig, 2015] Blai Bonet and Sven Koenig, editors. *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence, January 25-30, 2015, Austin, Texas, USA*. AAAI Press, 2015.

[Cabalar and Ferraris, 2007] Pedro Cabalar and Paolo Ferraris. Propositional theories are strongly equivalent to logic programs. *Theory and Practice of Logic Programming*, 7(6):745–759, 2007.

[Delgrande and Wang, 2015] James P. Delgrande and Kewen Wang. A syntax-independent approach to forgetting in disjunctive logic programs. In Bonet and Koenig [2015], pages 1482–1488.

[Eiter and Wang, 2008] Thomas Eiter and Kewen Wang. Semantic forgetting in answer set programming. *Artificial Intelligence*, 172(14):1644–1672, 2008.

[Eiter *et al.*, 2007] Thomas Eiter, Michael Fink, and Stefan Woltran. Semantical characterizations and complexity of equivalences in answer set programming. *ACM Transaction of Computional Logic*, 8(3), 2007.

[Ferraris, 2005] Paolo Ferraris. Answer sets for propositional theories. In *Proceedings of the 8th International Conference on Logic Programming and Nonmonotonic Reasoning (LPNMR-05), Diamante, Italy, September 5-8, 2005*, pages 119–131, 2005.

[Gelfond and Lifschitz, 1988] Michael Gelfond and Vladimir Lifschitz. The stable model semantics for logic programming. In *Proceedings of the Fifth International Conference and Symposium on Logic Programming*, pages 1070–1080, Seattle, Washington, 1988. MIT Press.

[Heyting, 1930] Arend Heyting. Die formalen regeln der intuitionistischen logik. *Deütsche Akademie der Wissenschaften zu Berlin, Mathematisch-Naturwissenschaftliche Klasse*, pages 42–56, 1930.

[Janhunen *et al.*, 2009] Tomi Janhunen, Emilia Oikarinen, Hans Tompits, and Stefan Woltran. Modularity aspects of disjunctive stable models. *Journal of Artificial Intelligence Research*, 35:813–857, 2009.

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

[Konev *et al.*, 2012] Boris Konev, Michel Ludwig, Dirk Walther, and Frank Wolter. The logical difference for the lightweight description logic EL. *Journal of Artificial Intelligence Research*, 44:633–708, 2012.

[Lang and Marquis, 2010] Jérôme Lang and Pierre Marquis. Reasoning under inconsistency: A forgetting-based approach. *Artificial Intelligence*, 174(12):799–823, 2010.

[Lang *et al.*, 2003] Jérôme Lang, Paolo Liberatore, and Pierre Marquis. Propositional independence. *Journal of Artificial Intelligence Research*, 18:391–443, 2003.

[Lifschitz *et al.*, 1999] Vladimir Lifschitz, Lappoon R. Tang, and Hudson Turner. Nested expressions in logic programs. *Annals of Mathematics and Artificial Intelligence*, 25(3-4):369–389, 1999.

[Lifschitz *et al.*, 2001] Vladimir Lifschitz, David Pearce, and Agustín Valverde. Strongly equivalent logic programs. *ACM Transactions on Computational Logic (TOCL)*, 2(4):526–541, 2001.

[Lin and Reiter, 1994] Fangzhen Lin and Ray Reiter. Forget it. In *Working Notes of AAAI Fall Symposium on Relevance*, pages 154–159, 1994.

[Lin and Reiter, 1997] Fangzhen Lin and Ray Reiter. How to progress a database. *Artificial Intelligence*, 92(1):131–167, 1997.

[Lin, 2001] Fangzhen Lin. On strongest necessary and weakest sufficient conditions. *Artificial Intelligence*, 128(1):143–159, 2001.

[Liu and Wen, 2011] Yongmei Liu and Ximing Wen. On the progression of knowledge in the situation calculus. In *IJCAI 2011, Proceedings of the 22nd International Joint Conference on Artificial Intelligence*, pages 976–982, Barcelona, Catalonia, Spain, 2011. IJCAI/AAAI.

[Pearce *et al.*, 2009] David Pearce, Hans Tompits, and Stefan Woltran. Characterising equilibrium logic and nested logic programs: Reductions and complexity. *Theory and Practice of Logic Programming*, 9(5):565–616, 2009.

[Pearce, 1996] David Pearce. A new logical characterisation of stable models and answer sets. In *Non-Monotonic Extensions of Logic Programming, NMELP'96*, volume 1216 of *Lecture Notes in Computer Science*, pages 57–70, Bad Honnef, Germany, 1996. Springer.

[Rajaratnam *et al.*, 2014] David Rajaratnam, Hector J. Levesque, Maurice Pagnucco, and Michael Thielscher. Forgetting in action. In Chitta Baral, Giuseppe De Giacomo, and Thomas Eiter, editors, *KR*. AAAI Press, 2014.

[Wang *et al.*, 2010] Zhe Wang, Kewen Wang, Rodney W. Topor, and Jeff Z. Pan. Forgetting for knowledge bases in dl-lite. *Annuals of Mathematics and Artificial Intelligence*, 58(1-2):117–151, 2010.

[Wang *et al.*, 2012] Yisong Wang, Yan Zhang, Yi Zhou, and Mingyi Zhang. Forgetting in logic programs under strong equivalence. In *Proceedings of the 13th International Conference on Principles of Knowledge Representation and Reasoning (KR-12)*, pages 643–647, 2012.

[Wang *et al.*, 2013] Yisong Wang, Kewen Wang, and Mingyi Zhang. Forgetting for answer set programs revisited. In *Proceedings of the 23rd International Joint Conference on Artificial Intelligence (IJCAI-13), Beijing, China, August 3-9, 2013*, pages 1162–1168, 2013.

[Wang *et al.*, 2014] Yisong Wang, Yan Zhang, Yi Zhou, and Mingyi Zhang. Knowledge forgetting in answer set programming. *Journal of Artificial Intelligence Research*, 50:31–70, 2014.

[Wang *et al.*, 2015] Yisong Wang, Kewen Wang, Zhe Wang, and Zhiqiang Zhuang. Knowledge forgetting in circumscription: A preliminary report. In Bonet and Koenig [2015], pages 1649–1655.

[Wong, 2009] Ka-Shu Wong. *Forgetting in Logic Programs*. PhD thesis, The University of New South Wales, 2009.

[Zhang and Foo, 2006] Yan Zhang and Norman Y Foo. Solving logic program conflict through strong and weak forgettings. *Artificial Intelligence*, 170(8):739–778, 2006.

[Zhang and Zhou, 2009] Yan Zhang and Yi Zhou. Knowledge forgetting: Properties and applications. *Artificial Intelligence*, 173(16):1525–1537, 2009.