Supplementary Material for the Paper

Forgetting in Answer Set Programming – A Survey

published in Theory and Practice of Logic Programming

RICARDO GONÇALVES and MATTHIAS KNORR and JOÃO LEITE
NOVA LINCS, Departamento de Informática, Faculdade de Ciências e Tecnologia, Universidade Nova de Lisboa, Portugal
(*e-mail:* `rjrg|mkn|jleite@fct.unl.pt`)

Appendix

In this section, we prove the results stated in the paper. For Theorem 5, we divide it into several insightful results, grouped by the classes of operators, which we present after the theorem. These intermediate results are also stated as theorems (Theorems 11-23). For the sake of simplicity and in order to highlight the important results, we prove some more technical lemmas only subsequently.

Theorem 1
The following relations hold for all F:

1. **(W)** is equivalent to **(NP)**;
2. **(SP)** implies **(SE)**;
3. **(CP)** and **(SI)** together are equivalent to **(SP)**;
4. **(sC)** and **(wC)** together are equivalent to **(CP)**;
5. **(CP)** implies **(wE)**;
6. **(SE)** and **(SI)** together imply **(PP)**;
7. **(wSP)** and **(sSP)** together are equivalent to **(SP)**;
8. **(sC)** and **(SI)** together imply **(sSP)**;
9. **(wC)** and **(SI)** together imply **(wSP)**;
10. **(W)** and **(PP)** together imply **(SC)**;
11. **(SC)** implies **(SE)**;
12. **(W)** implies **(NC)**;
13. **(wC)** is incompatible with **(W)** for F over $\mathcal{C}$ such that $\mathcal{C}_n \subseteq \mathcal{C}$;
14. **(wC)** and **(UI)** together are incompatible with **(RC)** for F over $\mathcal{C}$ such that $\mathcal{C}_n \subseteq \mathcal{C}$.
15. **(SI)** implies **(UI)**;
16. **(CP)** and **(UI)** together are equivalent to **(UP)**;
17. **(SI)** implies $(\mathbf{SI_u})$;
18. **(UP)** is incompatible with $(\mathbf{SI_u})$.

Proof
The first two results were already proven in the literature, so we focus on the remainder.

To show 3., first, let F be a class of operators satisfying **(SP)**. Let $\mathsf{f} \in \mathsf{F}$, $V \subseteq \mathcal{A}$, and P a program. The fact that F satisfies **(CP)** follows immediately from **(SP)**, by taking $R = \emptyset$. The proof that **(SP)** implies **(SI)** involves reasoning with subsignatures. For a program P' over $\mathcal{A} \setminus V$, we denote by $\mathcal{HT}_{\|V}(P)$ the set of HT-models of P over signature $\mathcal{A} \setminus V$. Let $R \in \mathcal{C}$ with $\mathcal{A}(R) \subseteq \mathcal{A} \setminus V$. To show **(SI)** we aim to prove that $\mathsf{f}(P,V) \cup R \equiv_{\mathsf{HT}} \mathsf{f}(P \cup R, V)$, i.e., $\mathcal{HT}(\mathsf{f}(P,V) \cup R) = \mathcal{HT}(\mathsf{f}(P \cup R, V))$. We first prove that $\mathsf{f}(P,V) \cup R \equiv_{\mathsf{HT}} \mathsf{f}(P \cup R, V)$ for the restricted signature $\mathcal{A} \setminus V$, i.e., $\mathcal{HT}_{\|V}(\mathsf{f}(P,V) \cup R) = \mathcal{HT}_{\|V}(\mathsf{f}(P \cup R, V))$. Let R' be a program such that $\mathcal{A}(R') \subseteq \mathcal{A} \setminus V$. Then, by **(SP)**, we have $\mathcal{AS}(\mathsf{f}(P,V) \cup R \cup R') = \mathcal{AS}(P \cup R \cup R')_{\|V}$. Also by **(SP)**, we have $\mathcal{AS}(\mathsf{f}(P \cup R, V) \cup R') = \mathcal{AS}(P \cup R \cup R')_{\|V}$. Therefore, $\mathcal{AS}(\mathsf{f}(P,V) \cup R \cup R') = \mathcal{AS}(\mathsf{f}(P \cup R, V) \cup R')$. This means that $\mathsf{f}(P,V) \cup R \equiv_{\mathsf{HT}} \mathsf{f}(P \cup R, V)$ for the restricted signature $\mathcal{A} \setminus V$, i.e., $\mathcal{HT}_{\|V}(\mathsf{f}(P,V) \cup R) = \mathcal{HT}_{\|V}(\mathsf{f}(P \cup R, V))$. Now let M be an HT-interpretation over $\mathcal{A}$. Then, since $\mathsf{f}(P,V) \cup R$ is a program over $\mathcal{A} \setminus V$, we have that $M \models_{\mathsf{HT}} \mathsf{f}(P,V) \cup R$ iff $M_{\|V} \models_{\mathsf{HT}} \mathsf{f}(P,V) \cup R$. Since $\mathcal{HT}_{\|V}(\mathsf{f}(P,V) \cup R) = \mathcal{HT}_{\|V}(\mathsf{f}(P \cup R, V))$, we have that $M_{\|V} \models_{\mathsf{HT}} \mathsf{f}(P,V) \cup R$ iff $M_{\|V} \models_{\mathsf{HT}} \mathsf{f}(P \cup R, V)$. Since $\mathsf{f}(P \cup R, V)$ is a program over $\mathcal{A} \setminus V$, we know that $M_{\|V} \models_{\mathsf{HT}} \mathsf{f}(P \cup R, V)$ iff $M \models_{\mathsf{HT}} \mathsf{f}(P \cup R, V)$. Therefore, $\mathcal{HT}(\mathsf{f}(P,V) \cup R) = \mathcal{HT}(\mathsf{f}(P \cup R, V))$.

Now let F be a class of operators satisfying **(CP)** and **(SI)**. Let $\mathsf{f} \in \mathsf{F}$, $V \subseteq \mathcal{A}$, P a program, and R a program with $\mathcal{A}(R) \subseteq \mathcal{A} \setminus V$. Using **(SI)** we have $\mathcal{AS}(\mathsf{f}(P,V) \cup R) = \mathcal{AS}(\mathsf{f}(P \cup R, V))$. Using **(CP)** we have $\mathcal{AS}(\mathsf{f}(P \cup R, V)) = \mathcal{AS}(P \cup R)_{\|V}$. Putting these together we have $\mathcal{AS}(\mathsf{f}(P,V) \cup R) = \mathcal{AS}(P \cup R)_{\|V}$. Therefore, **(SP)** holds.

Result 4. follows immediately from the fact that **(sC)** and **(wC)** correspond to the two inclusions on the equality condition in **(CP)**.

To show 5., let F be a class of operators satisfying **(CP)**. Let $\mathsf{f} \in \mathsf{F}$, $V \subseteq \mathcal{A}$, and P_1, P_2 be two programs such that $\mathcal{AS}(P_1) = \mathcal{AS}(P_2)$. By **(CP)**, we have the equalities: $\mathcal{AS}(\mathsf{f}(P_1, V)) = \mathcal{AS}(P_1) \setminus V = \mathcal{AS}(P_2) \setminus V = \mathcal{AS}(\mathsf{f}(P_2, V))$. Thus, $\mathcal{AS}(\mathsf{f}(P_1, V)) = \mathcal{AS}(\mathsf{f}(P_2, V))$.

To show 6., let F be a class of operators satisfying **(SE)** and **(SI)**. Let $\mathsf{f} \in \mathsf{F}$, $V \subseteq \mathcal{A}$, and P a program. Consider $R = \{r : P \models_{\mathsf{HT}} r \text{ and } r \text{ does not contain variables from } V\}$. Then clearly $\mathsf{f}(P,V) \cup R \models_{\mathsf{HT}} R$. By **(SI)**, we have that $\mathsf{f}(P,V) \cup R \equiv_{\mathsf{HT}} \mathsf{f}(P \cup R, V)$. We can then conclude that $\mathsf{f}(P \cup R, V) \models_{\mathsf{HT}} R$. Now, by **(SE)** and the fact that $P \cup R \equiv_{\mathsf{HT}} P$, we can conclude that $\mathsf{f}(P \cup R, V) \equiv_{\mathsf{HT}} \mathsf{f}(P,V)$. Therefore, $\mathsf{f}(P,V) \models_{\mathsf{HT}} R$, i.e., $\mathsf{f}(P,V) \models_{\mathsf{HT}} r$ for every rule r such that $P \models_{\mathsf{HT}} r$ and r does not contain V. Thus **(PP)** is satisfied.

Result 7. can be shown straightforwardly due to the fact that the conditions of **(sSP)** and **(wSP)** provide the two directions of the condition of **(SP)**.

To show 8., let $\mathsf{f} \in \mathsf{F}$, $P \in \mathcal{C}(\mathsf{f})$ and $V \subseteq \mathcal{A}$ such that F satisfies **(sC)** and **(SI)**. Consider $M \in \mathcal{AS}(\mathsf{f}(P,V) \cup R)$. By **(SI)**, we have that $M \in \mathcal{AS}(\mathsf{f}(P \cup R, V))$. By **(sC)**, we obtain $M \in \mathcal{AS}(P \cup R)_{\|V}$ which finishes the proof.

To show 9., let $\mathsf{f} \in \mathsf{F}$, $P \in \mathcal{C}(\mathsf{f})$ and $V \subseteq \mathcal{A}$ such that F satisfies **(wC)** and **(SI)**. Consider $M \in \mathcal{AS}(P \cup R)_{\|V}$. By **(wC)**, we obtain $M \in \mathcal{AS}(\mathsf{f}(P \cup R, V))$. Then, by **(SI)**, we have $M \in \mathcal{AS}(\mathsf{f}(P,V) \cup R)$ which finishes the proof.

Results 10.–12. have been shown by Gonçalves et al. (2017).

To show 13. and 14., we rely on the following example which fits the class of programs required. Consider $P = \{a \leftarrow p; p \leftarrow \mathit{not\,not}\, p\}$ from which we want to forget about p. Note that though the program uses double negation, we can easily replace $p \leftarrow \mathit{not\,not}\, p$ by two rules $p \leftarrow \mathit{not}\, q$ and $q \leftarrow \mathit{not}\, p$ and forget both p and q. To ease the presentation, we rely on P. The idea is now to show that in both cases, it is not possible to satisfy both (sets of) properties simultaneously. Note that P has two answer sets, $\emptyset$ and $\{a, p\}$.

Consider first 13., and suppose both **(wC)** and **(W)** are satisfied. By **(wC)**, we have that $\mathcal{AS}(\mathsf{f}(P, \{p\})) \supseteq \{\emptyset, \{a\}\}$. But there is only one rule over a that has (at least) these two answer sets: $a \leftarrow \mathit{not\,not}\, a$. Now, P has an HT-model $\langle \emptyset, a \rangle$, while a forgetting result $\mathsf{f}(P, \{p\})$ cannot have this HT-model. We obtain a contradiction to **(W)** being satisfied.

Now consider 14., and suppose that **(wC)**, **(UI)**, and **(RC)** are satisfied. Again, by **(wC)**, we have that $\mathcal{AS}(\mathsf{f}(P,\{p\})) \supseteq \{\emptyset,\{a\}\}$, and there is only one rule over a that has (at least) these two answer sets: $a \leftarrow \mathit{not\,not}\,a$. Let this rule be r in **(RC)**. Clearly, $\mathsf{f}(P,\{p\}) \models_{\mathsf{HT}} r$. Consider r' such that $P \models_{\mathsf{HT}} r'$. Note that thus r' cannot be r itself. By **(UI)**, we have $\mathsf{f}(\{r'\},\{p\}) \cup \{a \leftarrow\} \equiv_{\mathsf{HT}} \mathsf{f}(\{r'\} \cup \{a \leftarrow\},\{p\})$. We consider two cases. First, $\mathsf{f}(\{r'\},\{p\})$ has no HT-model $\langle \emptyset, a\rangle$. Then, it cannot have an HT-model $\langle a,a\rangle$. We derive a contradiction to the condition imposed by **(UI)**. Second, $\mathsf{f}(\{r'\},\{p\})$ has an HT-model $\langle \emptyset, a\rangle$ (as well as $\langle a,a\rangle$). But this is a contradiction to $\mathsf{f}(P,\{p\}) \models_{\mathsf{HT}} r$, which finishes the argument.

Then, 15. is a consequence of the definition of the respective properties where the uniform property, **(UI)**, is just a special case of **(SI)**, and the proof of 16. is a precise adaptation of that of 3. replacing all used programs R by sets of facts. Finally, 17 is straightforward by the definitions of **(SI)** and $(\mathbf{SI_u})$, and 18. has been shown by Gonçalves et al. (2021). □

Theorem 2

For all Horn programs P, every $V \subseteq \mathcal{A}(P)$, and all forgetting operators $\mathsf{f}_1, \mathsf{f}_2$ in the classes F_{strong}, F_{weak}, F_{S}, F_{HT}, F_{SM}, F_{Sas}, F_{SE}, F_{SP}, F_{R}, F_{M}, and F_{UP}, it holds that $\mathsf{f}_1(P,V) \equiv_{\mathsf{HT}} \mathsf{f}_2(P,V)$.

Proof

We will base our proof on Theorem 10 by Wang et al. (2014), a representation result for HT-Forgetting. This theorem implies that, whenever F_{HT} is closed for a class of programs, then it coincides with any class of forgetting operators that is closed for the same class and satisfies **(W)** and **(PP)** for that class. Our aim now is to prove that every class of forgetting operators mentioned in the statement of this theorem coincides with F_{HT}. Just for the sake of simplify we will use the results in Theorem 5. We should stress, nevertheless, that there is no circular dependence between these two results.

In order to prove that a class F coincides with F_{HT} on the class of Horn programs, we need to prove that F_{HT} is closed for the class of Horn programs, and: (i) F is closed for the class of Horn programs, (ii) F satisfies **(W)** and **(PP)** on the class of Horn programs.

The fact that F_{HT} is closed for the class of Horn programs is precisely Theorem 8 (Wang et al. 2014). So we now prove that each other class of operators satisfies (i) and (ii).

First, it is straightforward from the definitions of F_{strong} and F_{weak} that they coincide on the class of Horn programs. It is also clear that they are closed for this class. Since F_{strong} satisfies **(W)** (Theorem 5), F_{weak} satisfies **(PP)** (Theorem 5), and they coincide on the class of Horn programs, they both satisfy **(PP)** and **(W)** on that class.

Thm. 4 states that F_{SE} and F_{S} coincide with F_{HT} on the class of disjunctive programs, whenever the result of F_{HT} is a disjunctive program. Since F_{HT} is closed for the class of Horn programs, it follows immediately that F_{SE} and F_{S} coincide with F_{HT} when restricted to Horn programs.

It follows easily from the algorithm presented by Knorr and Alferes (2014) that for $\mathsf{f}_{Sas} \in \mathsf{F}_{Sas}$, $V \subseteq \mathcal{A}$, and P a Horn program $\mathsf{f}_{Sas}(P,V) \equiv_{\mathsf{HT}} \mathsf{f}_{strong}(P,V)$. As a consequence, F_{Sas}, just as F_{strong}, coincides with F_{HT} on the class of Horn programs.

Since F_{HT} coincides with F_{Sas} for Horn programs, we have that F_{HT} satisfies **(CP)** for Horn programs. Therefore, by definition, F_{SM} coincides with F_{HT} for Horn programs.

In the case of F_{SP}, F_{R}, and F_{M}, the result is a consequence of Proposition 6 (Gonçalves et al. 2020), which shows that F_{SP} coincides with F_{HT} when restricted to Horn programs, and Proposition 21 (Gonçalves et al. 2020), which shows that the three classes F_{SP}, F_{R}, and F_{M} coincide when restricted to Horn programs.

Finally, in the case of F_{UP}, Proposition 1 (Gonçalves et al. 2019) shows that F_{UP} coincides with F_{HT} when restricted to Horn programs. □

Theorem 3
Consider the class of disjunctive programs. Then, F_S and F_{SE} coincide.

Proof
First recall that, on the one hand, given a disjunctive program P and $V \subseteq \mathcal{A}$, $P_S(P,V)$ is defined by removing from $Cn(P,a) = \{r \mid r \text{ disjunctive}, P \models_{\mathsf{HT}} r, \mathcal{A}(r) \subseteq \mathcal{A}(P)\}$ all rules in which atoms from V occur. On the other hand, given a disjunctive program P and $V \subseteq \mathcal{A}$, for any $\mathsf{f}_{\mathsf{SE}} \in \mathsf{F}_{\mathsf{SE}}$, we have that $\mathsf{f}_{\mathsf{SE}}(P,V)$ is equivalent to $Cn_{\mathcal{A}}(P) \cap \mathcal{L}_{\mathcal{A}(P)\setminus V}$, where $Cn_{\mathcal{A}}(P) = \{r \in \mathcal{L}_{\mathcal{A}} \mid r \text{ disjunctive}, P \vdash_s r\}$. Since, as Wong (2009) showed, the consequence $\vdash_s$ is sound and complete with respect to $\models_{\mathsf{HT}}$, we have that $Cn(P,a) = Cn_{\mathcal{A}}(P) \cap \mathcal{L}_{\mathcal{A}(P)}$. Therefore, $Cn(P,a) \cap \mathcal{L}_{\mathcal{A}(P)\setminus V} = Cn_{\mathcal{A}}(P) \cap \mathcal{L}_{\mathcal{A}(P)\setminus V}$. This means that $\mathsf{f} \in \mathsf{F}_S$ iff $\mathsf{f}(P,V) \equiv_{\mathsf{HT}} Cn(P,a) \cap \mathcal{L}_{\mathcal{A}(P)\setminus V}$ iff $\mathsf{f}(P,V) \equiv_{\mathsf{HT}} Cn_{\mathcal{A}}(P)\mathcal{L}_{\mathcal{A}(P)\setminus V}$ iff $\mathsf{f} \in \mathsf{F}_{\mathsf{SE}}$. Therefore, $\mathsf{F}_S = \mathsf{F}_{\mathsf{SE}}$. □

Theorem 4
Let P be a disjunctive program, $V \subseteq \mathcal{A}(P)$, $\mathsf{f}_S \in \mathsf{F}_S$, $\mathsf{f}_{\mathsf{HT}} \in \mathsf{F}_{\mathsf{HT}}$, and $\mathsf{f}_{\mathsf{SE}} \in \mathsf{F}_{\mathsf{SE}}$. Then, $\mathsf{f}_S(P,V) \equiv_{\mathsf{HT}} \mathsf{f}_{\mathsf{HT}}(P,V) \equiv_{\mathsf{HT}} \mathsf{f}_{\mathsf{SE}}(P,V)$ whenever $\mathsf{f}_{\mathsf{HT}}(P,V)$ is strongly equivalent to a disjunctive program.

Proof
Let P be a disjunctive program, $V \subseteq \mathcal{A}(P)$, such that $\mathsf{f}_{\mathsf{HT}}(P,V)$ is strongly equivalent to a disjunctive program. We only prove that $\mathsf{f}_{\mathsf{HT}}(P,V) \equiv_{\mathsf{HT}} \mathsf{f}_{\mathsf{SE}}(P,V)$ since Theorem 3 already states that $\mathsf{f}_S(P,V) \equiv_{\mathsf{HT}} \mathsf{f}_{\mathsf{SE}}(P,V)$. Since both F_{HT} and F_{SE} satisfy **(W)**, we have that (i) $P \models_{\mathsf{HT}} \mathsf{f}_{\mathsf{HT}}(P,V)$ and (ii) $P \models_{\mathsf{HT}} \mathsf{f}_{\mathsf{SE}}(P,V)$. Since F_{SE} satisfies **(PP)** (restricted to disjunctives programs) and $\mathsf{f}_{\mathsf{HT}}(P,V)$ is equivalent to a disjunctive program, we can use (i) to conclude that $\mathsf{f}_{\mathsf{SE}}(P,V) \models_{\mathsf{HT}} \mathsf{f}_{\mathsf{HT}}(P,V)$. Since F_{HT} satisfies **(PP)** we can use (ii) to conclude that $\mathsf{f}_{\mathsf{HT}}(P,V) \models_{\mathsf{HT}} \mathsf{f}_{\mathsf{SE}}(P,V)$. Therefore, $\mathsf{f}_{\mathsf{SE}}(P,V) \equiv_{\mathsf{HT}} \mathsf{f}_{\mathsf{HT}}(P,V)$. □

Theorem 5
All results in Table 3 hold.

Proof
The result is an immediate consequence of Thms. 11 to 23. □

Theorem 6
For Horn programs, the following holds:

- F_{strong}, F_{weak}, F_S, F_{HT}, F_{SM}, F_{Sas}, F_{SE}, F_{SP}, F_{R}, F_{M}, and F_{UP} satisfy **(W)**, **(RC)**, **(SP)**, and **(PI)**;
- F_{sem} satisfies **(CP)** and **(PI)**;
- F_W satisfies the same properties as in the general case.

Proof
The fact that F_{strong}, F_{weak}, F_{S}, F_{HT}, F_{SM}, F_{Sas}, F_{SE}, F_{SP}, F_{R}, F_{M}, and F_{UP}, when restricted to Horn programs, satisfy **(W)**, **(RC)**, **(SP)**, and **(PI)** follows easily from Thm. 2, and the fact that F_{Sas} satisfies all these properties.

The fact that F_{sem}, when restricted to Horn programs, additionally satisfies **(CP)** follows easily from the fact that any Horn program has at most one answer set, i.e., $\mathcal{AS}(P) = \{A\}$ or $\mathcal{AS}(P) = \{\}$. Then, for every $\mathsf{f}_{sem} \in \mathsf{F}_{sem}$, by definition we have $\mathcal{AS}(\mathsf{f}_{sem}(P,V)) = \mathcal{MIN}(\mathcal{AS}(P)_{\|V})$. Since $\mathcal{AS}(P)$ has at most one element we have $\mathcal{AS}(\mathsf{f}_{sem}(P,V)) = \{A_{\|V}\} = \mathcal{AS}(P)_{\|V}$ or $\mathcal{AS}(\mathsf{f}_{sem}(P,V)) = \{\} = \mathcal{AS}(P)_{\|V}$. Therefore, F_{sem} satisfies **(CP)** when restricted to Horn programs. □

Theorem 7
If a class F of operators over a class $\mathcal{C}$ of logic programs satisfies property **(CP)**, then every operator of that class can be used to obtain uniform interpolants w.r.t. $\mid\!\sim$.

Proof
For that, assume that f satifies **(CP)**, i.e, given P and $V \subseteq \mathcal{A}$, we have that $\mathcal{AS}(\mathsf{f}(P,V)) = \mathcal{AS}(P)_{\|V}$. Condition (i) of uniform interpolation follows easily from the fact that every $M \in \mathcal{AS}(P)$ is such that $M_{\|V} \in \mathcal{AS}(\mathsf{f}(P,V))$, and therefore $M \models \mathsf{f}(P,V)$. For condition (ii), we let R be a program such that $P \vdash R$ and $\mathcal{A}(R) \subseteq \mathcal{A}\backslash V$, and we aim to conclude that $\mathsf{f}(P,V) \mid\!\sim R$. Let $M \in \mathcal{AS}(\mathsf{f}(P,V))$. Then, since we are assuming that $\mathcal{AS}(\mathsf{f}(P,V)) = \mathcal{AS}(P)_{\|V}$, we know that there is $M^* \in \mathcal{AS}(P)$ such that $M = M^*\backslash V$. Given that $P \vdash R$, we can conclude that $M^* \models R$, and since R is a program over $\mathcal{A}\backslash V$, we can conclude that $M \models R$. □

Theorem 8
Every forgetting operator of the classes F_{SM}, F_{M}, F_{UP}, and F_{R} can be used to obtain uniform interpolants with respect to $\mid\!\sim$.

Proof
The case of F_{SM}, F_{M}, and F_{UP} follows from Thm.7 and the fact that these classes satisfy **(CP)**.

In the case of F_{R} let P be a program and $V \subseteq \mathcal{A}$. We start by considering $M \in \mathcal{AS}(P)$. Taking into account the definition of F_{R}, it is easy to see that, for every $\mathsf{f} \in \mathsf{F}_{\mathsf{R}}$, we have that $\langle M\backslash V, M\backslash V\rangle \in \mathcal{HT}(\mathsf{f}(P,V))$. This implies that $M\backslash V \models \mathsf{f}(P,V)$, and since $\mathsf{f}(P,V)$ is over $\mathcal{A}\backslash V$, we also have that $M \models \mathsf{f}(P,V)$, meaning that condition (i) of uniform interpolation is satisfied. Condition (ii) is a consequence of F_{R} satisfying **(sC)**, i.e., $\mathcal{AS}(\mathsf{f}(P,V)) \subseteq \mathcal{AS}(P)_{\|V}$. □

Theorem 9
A class F of forgetting operators can be used to obtain uniform interpolants w.r.t. $\models_{\mathsf{HT}}$ iff F satisfies both **(W)** and **(PP)**.

Proof
The result follows immediately, since the conditions (i) and (ii) of uniform interpolation precisely coincide with **(W)** and **(PP)**, respectively. □

Theorem 10
Every forgetting operator of the classes F_{HT} and F_{S} can be used to obtain uniform interpolants with respect to $\models_{\mathsf{HT}}$.

Proof

The proof follows immediately from Thm. 9 and the fact that F_{HT} and F_S satisfy **(W)** and **(PP)**. □

We now present several intermediate results, each presenting the set of properties satisfied and not satisfied by each class of operators considered in the paper, that we will combine to prove Theorem 5.

Theorem 11

F_{strong} satisfies **(W)**, **(SI)**, **(RC)**, **(NC)**, **(PI)**, **(UI)**, $(\mathbf{SI_u})$, $(\mathbf{E}_{\mathcal{C}_H})$, $(\mathbf{E}_{\mathcal{C}_n})$, but not **(sC)**, **(wE)**, **(SE)**, **(PP)**, **(SC)**, **(CP)**, **(SP)**, **(wC)**, **(wSP)**, **(sSP)** and **(UP)**.

Proof

To prove **(W)**, let P be a normal logic program over a signature $\mathcal{A}$, and $v \subseteq \mathcal{A}$. In the first step of the definition of $\mathsf{f}_{strong}(P,V)$ we obtain an intermediate program P' by adding to P, for each $v \in V$, the rules $a \leftarrow B_1, B_2, not\,C_1, not\,C_2$ such that $a \leftarrow B_1, v, not\,C_1$ and $v \leftarrow B_2, not\,C_2$ are in P. By Lemma 7 every HT model of both $a \leftarrow B_1, v, not\,C_1$ and $v \leftarrow B_2, not\,C_2$ is also a model of $a \leftarrow B_1, B_2, not\,C_1, not\,C_2$. Thus, $\mathcal{HT}(P) \subseteq \mathcal{HT}(P')$. Therefore, $\mathcal{HT}(P \cup P') = \mathcal{HT}(P) \cap \mathcal{HT}(P') = \mathcal{HT}(P)$, i.e., $P \cup P'$ is strongly equivalent to P. In the second and last step of the definition of $\mathsf{f}_{strong}(P,V)$, all rules from $P \cup P'$ containing some $v \in V$ are eliminated, thus obtaining $\mathsf{f}_{strong}(P,V)$. Since $\mathsf{f}_{strong}(P,V) \subseteq P \cup P'$, we have that $\mathcal{HT}(P) = \mathcal{HT}(P \cup P') \subseteq \mathcal{HT}(\mathsf{f}_{strong}(P,V))$. Note that we proved the result for forgetting one variable. The general result then follows from (F6) (Wong 2009), which was shown to hold for both Strong and Weak Forgetting.

For **(SI)**, let P be a normal logic program over a signature $\mathcal{A}$ and let $V \subseteq \mathcal{A}$. Let $P_{\|V}$ be the set of rules of P that do not mention any $v \in V$. It follows from the definition of Strong Forgetting that $P_{\|V} \subseteq \mathsf{f}_{strong}(P,V)$. Let R be a normal logic program over $\mathcal{A} \setminus V$. We now prove that $\mathsf{f}_{strong}(P \cup R, V) = \mathsf{f}_{strong}(P,V) \cup R$. Let us prove the two inclusions.

For the left to right inclusion, let $r \in \mathsf{f}_{strong}(P \cup R, V)$. Then, we have two cases:

a) $r \in P_{\|V} \cup R$. Then using the above observation we have that $r \in \mathsf{f}_{strong}(P,V) \cup R$.

b) $r = A \leftarrow B, B', not\,C, not\,C'$ such that there exist $r_1, r_2 \in P \cup R$ and $v \in V$ such that $r_1 = A \leftarrow B, v, not\,C$ and $r_2 = v \leftarrow B', not\,C'$. In this case, $r_1, r_2 \in P$ since R does not contain rules with atoms from V. Therefore, we can immediately conclude that $r \in \mathsf{f}_{strong}(P,V)$, and then $r \in \mathsf{f}_{strong}(P,V) \cup R$.

For the converse inclusion, let $r \in \mathsf{f}_{strong}(P,V) \cup R$. Then, we have two cases:

c) $r \in R$. Then using the above observation we have that $r \in \mathsf{f}_{strong}(P \cup R, V)$.

d) $r \in \mathsf{f}_{strong}(P,V)$. If $r \in P_{\|V}$, then using the above observation we have $r \in \mathsf{f}_{strong}(P \cup R, V)$. Otherwise, if $r = A \leftarrow B, B', not\,C, not\,C'$ such that there exist $r_1, r_2 \in P$ and $v \in V$ such that $r_1 = A \leftarrow B, v, not\,C$ and $r_2 = v \leftarrow B', not\,C'$. In this case, also $r_1, r_2 \in P \cup R$, and we can immediately conclude that $r \in \mathsf{f}_{strong}(P \cup R, V)$.

The fact that F_{strong} satisfies **(RC)**, **(NC)** and **(PI)** was proved by Gonçalves et al. (2017).

The fact that F_{strong} satisfies **(UI)** follows from the already proved fact that F_{strong} satisfies **(SI)**, and item 15 of Thm. 1.

The fact that F_{strong} satisfies $(\mathbf{SI_u})$ follows from the already proved fact that F_{strong} satisfies **(SI)**, and item 17 of Thm. 1.

The fact that $(\mathbf{E}_{\mathcal{C}_H})$ is satisfied follows from the simple observation that, when restricted to Horn programs, f_{strong} and f_{weak} coincide and the result is by definition always Horn.

Finally, the definition of the concrete operator f_{strong} by Zhang and Foo (2006) implicitly implies that F_{strong} satisfy $(\mathbf{E}_n)$.

For the negative results, Eiter and Wang (2008) provided counterexamples showing that F_{strong} does not satisfy **(sC)** and **(wE)**; Wong (2009) provided counterexamples showing that F_{strong} does not satisfy **(SE)**; Wang et al. (2014) provided counterexamples showing that F_{strong} does not satisfy **(PP)**, and Gonçalves et al. (2017) provided a counterexample to show that F_{strong} does not satisfy **(SC)**; Wang et al. (2013) presented a counterexample showing that F_{strong} does not satisfy **(CP)**; and Knorr and Alferes (2014) provided a counterexample showing that F_{strong} does not satisfy **(SP)**.

For **(wC)**, consider forgetting about b from $P = \{a \leftarrow not\, b,\ b \leftarrow not\, a\}$. Since $\mathcal{AS}(P) = \{\{a\},\{b\}\}$, to satisfy **(wC)**, the result must have the two answer sets $\{a\}$ and $\emptyset$, which is not possible for disjunctive programs. Therefore, F_{strong} does not satisfy **(wC)**.

Gonçalves et al. (2020) showed that F_{strong} does not satisfy **(wSP)** and **(sSP)**, and that it does not satisfy **(UP)** (2019). □

Theorem 12

F_{weak} satisfies **(PP)**, **(SI)**, **(RC)**, **(NC)**, **(PI)**, **(UI)**, $(\mathbf{SI_u})$, $(\mathbf{E}_{\mathcal{C}_H})$, $(\mathbf{E}_{\mathcal{C}_n})$, but not **(sC)**, **(wE)**, **(SE)**, **(W)**, **(SC)**, **(CP)**, **(SP)**, **(wC)**, **(wSP)**, **(sSP)** and **(UP)**.

Proof

For **(PP)** consider the inferential system given by Wong (2008), which is sound and complete with respect to HT-consequence for disjunctive programs. Suppose that $P \models_{\mathsf{HT}} r$, where $\mathcal{A}(r) \subseteq \mathcal{A}\backslash V$. Now recall that every rule of P that does not contain the atoms to be forgotten is in the result of forgetting. The rule (WGPPE) is the only inference rule in Wong's system that allows to derive a rule without the atoms to be forgotten from rules that have those atoms. Since, by definition of Weak Forgetting, the result of applying (WGPPE) belongs to the result of forgetting, we have that r must also be a consequence of the result of forgetting.

For **(SI)**, let P be a normal logic program over a signature $\mathcal{A}$ and let $V \subseteq \mathcal{A}$. Let $P_{\|V}$ be the set of rules of P that do not mention any $v \in V$. It follows from the definition of Weak Forgetting that $P_{\|V} \subseteq \mathsf{f}_{weak}(P,V)$. Let R be a normal logic program over $\mathcal{A} \setminus V$. We now prove that $\mathsf{f}_{weak}(P \cup R, V) = \mathsf{f}_{weak}(P,V) \cup R$. Let us prove the two inclusions.

For the left to right inclusion, let $r \in \mathsf{f}_{weak}(P \cup R, V)$. Then, we have three cases:

a) $r \in P_{\|V} \cup R$. Then using the above observation we have that $r \in \mathsf{f}_{weak}(P,V) \cup R$.

b) $r = A \leftarrow B, B', not\, C, not\, C'$ such that there exist $v \in V$, $r_1, r_2 \in P \cup R$ such that $r_1 = A \leftarrow B, v, not\, C$ and $r_2 = v \leftarrow B', not\, C'$. In this case, $r_1, r_2 \in P$ since R does not contain any $v \in V$. Therefore, we can immediately conclude that $r \in \mathsf{f}_{weak}(P,V)$, and then $r \in \mathsf{f}_{weak}(P,V) \cup R$.

c) $r = A \leftarrow B, not\, C$ such that there exists $v \in V$ and $r_1 = A \leftarrow B, not\, v, not\, C \in P \cup R$. Then, $r_1 \in P$ since R does not contain any $v \in V$. Therefore, we can conclude that $r \in \mathsf{f}_{weak}(P,V)$, and then $r \in \mathsf{f}_{weak}(P,V) \cup R$.

For the converse inclusion, let $r \in \mathsf{f}_{weak}(P,V) \cup R$. Then, we have two cases:

e) $r \in R$. Then using the above observation we have that $r \in \mathsf{f}_{weak}(P \cup R, V)$.

f) $r \in \mathsf{f}_{weak}(P,V)$.

i) If $r \in P_{\|V}$, then using the above observation we have $r \in \mathsf{f}_{weak}(P \cup R, V)$.

ii) If $r = A \leftarrow B, B', not\, C, not\, C'$ such that there exist $v \in V$, and $r_1, r_2 \in P$ such that $r_1 = A \leftarrow B, v, not\, C$ and $r_2 = v \leftarrow B', not\, C'$. In this case, also $r_1, r_2 \in P \cup R$, and we can immediately conclude that $r \in \mathsf{f}_{weak}(P \cup R, V)$.

iii) If $r = A \leftarrow B, not\, C$ such that there exist $v \in V$, and $r_1 = A \leftarrow B, not\, v, not\, C \in P$. Then, $r_1 \in P \cup R$, and we can conclude that $r \in \mathsf{f}_{weak}(P, V)$. Therefore, $r \in \mathsf{f}_{weak}(P, V) \cup R$.

The fact that F_{weak} satisfies **(RC)**, **(NC)** and **(PI)** was shown by Gonçalves et al. (2017).

In the case of **(UI)** the result follows from the already proved fact that F_{weak} satisfies **(SI)**, and item 15 of Thm. 1.

The fact that F_{weak} satisfies $(\mathbf{SI_u})$ follows from the already proved fact that F_{weak} satisfies **(SI)**, and item 17 of Thm. 1.

The fact that $(\mathbf{E}_{\mathcal{C}_H})$ is satisfied follows from the simple observation that, when restricted to Horn programs, f_{strong} and f_{weak} coincide and the result is by definition always Horn.

Finally, the definition of the concrete operator f_{weak} (Zhang and Foo 2006) implicitly implies that F_{weak} satisfy $(\mathbf{E}_n)$.

For the negative results, Eiter and Wang (2008) provided counterexamples showing that F_{weak} does not satisfy **(sC)** and **(wE)**; Wong (2009) provides counterexamples showing that F_{weak} does not satisfy **(SE)**; (Wang et al. 2014) provided a counterexample showing that F_{weak} does not satisfy **(W)**; the fact that F_{weak} does not satisfy **(SC)** was proved by Gonçalves et al. (2017); Wang et al. (2013) presented a counterexample showing that F_{weak} does not satisfy **(CP)**; and Knorr and Alferes (2014) provided a counterexample showing that F_{weak} does not satisfy **(SP)**.

For **(wC)**, consider forgetting about b from $P = \{a \leftarrow not\, b,\ b \leftarrow not\, a\}$. Since $\mathcal{AS}(P) = \{\{a\}, \{b\}\}$, to satisfy **(wC)**, the result must have the two answer sets $\{a\}$ and $\emptyset$, which is not possible for disjunctive programs. Therefore, F_{weak} does not satisfy **(wC)**.

Gonçalves et al. (2020) showed that F_{weak} does not satisfy **(wSP)** and **(sSP)**, and that it does not satisfy **(UP)**(Gonçalves et al. 2019). □

Theorem 13

F_{sem} satisfies **(sC)**, **(wE)**, **(PI)**, $(\mathbf{E}_{\mathcal{C}_H})$, $(\mathbf{E}_{\mathcal{C}_n})$, $(\mathbf{E}_{\mathcal{C}_d})$, but not **(SE)**, **(W)**, **(PP)**, **(SI)**, **(SC)**, **(RC)**, **(NC)**, **(CP)**, **(SP)**, **(wC)**, **(wSP)**, **(sSP)**, **(UP)**, **(UI)** and $(\mathbf{SI_u})$.

Proof

Eiter and Wang (2008) proved that F_{sem} satisfies **(sC)** (Proposition 6) and **(wE)** (Proposition 4), and, by the definitions of concrete operators, it is implicit that F_{sem} satisfies $(\mathbf{E}_{\mathcal{C}_n})$ and $(\mathbf{E}_{\mathcal{C}_d})$. The fact that F_{sem} satisfies **(PI)** was proved by Gonçalves et al. (2017). Finally, in the case of $(\mathbf{E}_{\mathcal{C}_H})$ the result follows from the observation that a variant of forget_1 (Eiter and Wang 2008) can be simplified which constructs a single program consisting of facts given that there does exist only at most one answer set in this case.

Regarding negative results, counterexamples have been presented in the literature for **(SE)** (Eiter and Wang 2008), for **(W)** (Wang et al. 2013), for **(PP)** (Wang et al. 2014), and for **(SI)** (Wong 2009). Also, the fact that F_{sem} does not satisfy **(SC)**, **(RC)**, and **(NC)** was proved by Gonçalves et al. (2017). Wang et al. (2013) showed that F_{sem} does not satisfy **(CP)**, and Knorr and Alferes (2014) that it does not satisfy **(SP)**. The counterexample for **(wC)** is similar to other cases, just by considering forgetting about b from $P = \{a \leftarrow not\, b, b \leftarrow not\, a\}$. Since $\mathcal{AS}(P) = \{\{a\}, \{b\}\}$, to satisfy **(wC)**, the result must have the two answer sets $\{a\}$ and $\emptyset$, which is not possible for disjunctive programs. Therefore, F_{sem} does not satisfy **(wC)**. Gonçalves et al. (2020) showed that F_{sem} does not satisfy **(wSP)** and **(sSP)**, and that it does not satisfy **(UP)** (Gonçalves et al. 2019). For the case of **(UI)**, consider forgetting about b from $P = \{a \leftarrow b\}$. In this case, an operator f of F_{sem} is such that $\mathsf{f}(P,b) = \{\}$, but adding $b \leftarrow$ before and after forgetting gives rise to results with different sets of answer sets. Finally, for the case of $(\mathbf{SI_u})$, consider $P = \{a \leftarrow b\}$ over the signature $\mathcal{A} = \{a,b,c\}$, and the program $R = \{b \leftarrow\}$ over $\mathcal{A}\backslash\{c\}$. Since $\mathcal{AS}(P) = \{\{\}\}$ and $\mathcal{AS}(P \cup R) = \{\{a,b\}\}$, then, according to the definition of $\mathsf{forget}_1 \in \mathsf{F}_{sem}$ in (Eiter and Wang 2008), we have that $\mathsf{forget}_1(P,\{c\}) = \{\}$ and $\mathsf{forget}_1(P \cup R,\{c\}) = \{a \leftarrow, b \leftarrow\}$. We can now easily see that $\mathcal{AS}(\mathsf{forget}_1(P,\{c\}) \cup R) = \{\{b\}\} \neq \{\{a,b\}\} = \mathcal{AS}(\mathsf{forget}_1(P \cup R,\{c\}))$, and therefore $\mathsf{forget}_1(P,\{c\}) \cup R \not\equiv_u \mathsf{forget}_1(P \cup R,\{c\})$. □

Theorem 14
F_S satisfies **(SE)**, **(W)**, **(PP)**, **(SC)**, **(RC)**, **(NC)**, **(PI)**, $(\mathbf{E}_{\mathcal{C}_H})$, $(\mathbf{E}_{\mathcal{C}_d})$, but not **(sC)**, **(wE)**, **(SI)**, **(CP)**, **(SP)**, **(wC)**, **(wSP)**, **(sSP)**, **(UP)**, **(UI)**, $(\mathbf{SI_u})$ and $(\mathbf{E}_{\mathcal{C}_n})$.

Proof
In the case of **(SE)**, Wong (2009) proved that F_S satisfies **(SE)** (Lemma 4.27), and it is implicit that it satisfies $(\mathbf{E}_{\mathcal{C}_d})$. Delgrande and Wang (2015) proved that F_{SE} satisfies **(SE)**, **(W)** and **(PP)** (Proposition 1). Therefore, by Thm. 3, the results also hold for F_S. The fact that F_S satisfies **(SC)**, **(RC)**, **(NC)** and **(PI)** was proved by Gonçalves et al. (2017). Finally, the fact that $(\mathbf{E}_{\mathcal{C}_H})$ holds follows from Theorem 8 (Wang et al. 2014), which states that F_{HT} is closed for the class of Horn programs, and Thm. 4, which implies that the operators from F_S are equivalent with those of F_{HT} for Horn programs.

For the negative results, in the case of **(sC)**, consider forgetting about a from $P = \{a \leftarrow not\, a\}$. The result should be strongly equivalent to $\emptyset$, i.e., the forgetting operation introduces a new answer set. Turning to **(wE)**, this property requires that the results of forgetting about p from $P = \{q \leftarrow not\, p, q \leftarrow not\, q\}$ and from $Q = \{q \leftarrow\}$ have the same answer-sets, while F_S requires that the results be strongly equivalent to $\mathsf{f}(P,p) = \{q \leftarrow not\, q\}$ and $\mathsf{f}(Q,p) = \{q \leftarrow\}$, which are obviously not equivalent.

Wong (2009) provided a counterexample showing that F_S does not satisfy **(SI)**. The fact that F_S does not satisfy **(CP)** and **(SP)** follows immediately from the fact that it does not satisfy **(sC)**, and items 3. and 4. of Thm. 1. The counterexample for **(wC)** is similar to other cases, just by considering forgetting about b from $P = \{a \leftarrow not\, b,\ b \leftarrow not\, a\}$. Since $\mathcal{AS}(P) = \{\{a\},\{b\}\}$, to satisfy **(wC)**, the result must have the two answer sets $\{a\}$ and $\emptyset$, which is not possible for disjunctive programs. Therefore, F_S does not satisfy **(wC)**. Gonçalves et al. (2020) showed that F_S does not satisfy **(wSP)** and **(sSP)**, and that it does not satisfy **(UP)** (Gonçalves et al. 2019).

To prove that F_S does not satisfy **(UI)**, consider forgetting about p from the program $P = \{a \leftarrow not\, p, b,\ p \leftarrow not\, a,\ \leftarrow p, b\}$. We have that $P \cup \{b \leftarrow\} \models_{\mathsf{HT}} a \leftarrow$, and since F_S satisfies **(PP)**, we have that $\mathsf{f}(P \cup \{b \leftarrow\}, p) \models_{\mathsf{HT}} a \leftarrow$. Since $P \not\models_{\mathsf{HT}} a \leftarrow b$, we have that $\mathsf{f}(P,p) \not\models_{\mathsf{HT}} a \leftarrow b$. Therefore $\mathsf{f}(P,p) \cup \{b \leftarrow\} \not\models_{\mathsf{HT}} a \leftarrow$.

For the case of $(\mathbf{SI_u})$, consider the same program as for **(UI)**, i.e., $P = \{a \leftarrow not\, p, b,\ p \leftarrow not\, a,\ \leftarrow p, b\}$. Again, we have that $P \cup \{b \leftarrow\} \models_{\mathsf{HT}} a \leftarrow$, and since F_S satisfies **(PP)**, we have that $\mathsf{f}(P \cup \{b \leftarrow\}, p) \models_{\mathsf{HT}} a \leftarrow$. This means that $\mathcal{HT}(\mathsf{f}(P \cup \{b \leftarrow\}, p\}) \subseteq \mathcal{HT}(\{a \leftarrow\}) = \{\langle a,a \rangle, \langle ab,ab \rangle, \langle a,ab \rangle\}$. Since, by using the same argument as before, it is also the case that $\mathsf{f}(P \cup \{b \leftarrow\}, p) \models_{\mathsf{HT}} b \leftarrow$, we can conclude that $\mathcal{HT}(\mathsf{f}(P \cup \{b \leftarrow\}, p)) \subseteq \mathcal{HT}(\{a \leftarrow\}) \cap \mathcal{HT}(\{b \leftarrow\}) = \{\langle ab,ab \rangle\}$. Since $\langle ab,ab \rangle \in \mathcal{HT}(P \cup \{b \leftarrow\})$ and F_S satisfies **(W)**, we have that $\langle ab,ab \rangle \in$

$\mathcal{HT}(\mathsf{f}(P\cup\{b\leftarrow\},p))$, and therefore $\mathcal{HT}(\mathsf{f}(P\cup\{b\leftarrow\},p))=\{\langle ab,ab\rangle\}$. We can then conclude that $\mathcal{AS}(\mathsf{f}(P\cup\{b\leftarrow\},p))=\{\{a,b\}\}$. On the other hand, we know that $\mathcal{HT}(\mathsf{f}(P,p)\cup\{b\leftarrow\})\subseteq\mathcal{HT}(\{b\leftarrow\})$. Since $\mathsf{f}(P,p)\cup\{b\leftarrow\}\not\models_{\mathsf{HT}}\{a\leftarrow\}$, we have that $\mathcal{HT}(\mathsf{f}(P,p)\cup\{b\leftarrow\})\not\subseteq\mathcal{HT}(\{a\leftarrow\})$. We therefore have two alternative possibilities: a) $\langle b,b\rangle\in\mathcal{HT}(\mathsf{f}(P,p)\cup\{b\leftarrow\})$ or b) $\langle b,ab\rangle\in\mathcal{HT}(\mathsf{f}(P,p)\cup\{b\leftarrow\})$. If a) is the case, then $\{b\}\in\mathcal{AS}(\mathsf{f}(P,p)\cup\{b\leftarrow\})$. If b) is the case we have that $\{a,b\}\notin\mathcal{AS}(\mathsf{f}(P,p)\cup\{b\leftarrow\})$. In both cases we can conclude that $\mathcal{AS}(\mathsf{f}(P,p)\cup\{b\leftarrow\})\neq\mathcal{AS}(\mathsf{f}(P\cup\{b\leftarrow\},p))=\{\{a,b\}\}$. Therefore, $\mathsf{f}(P,p)\cup\{b\leftarrow\}\not\equiv_u\mathsf{f}(P\cup\{b\leftarrow\},p)$, showing that F_{sem} does not satisfy $(\mathbf{SI_u})$.

Finally, for the negative result of $(\mathbf{E}_{\mathcal{C}_n})$, we prove for F_{SE}. The corresponding result for F_S follows immediately from Theorem 3. Let $P=\{h_1\leftarrow not\,a_1,\ h_2\leftarrow not\,a_2,\ h_3\leftarrow a_1,a_2\}$ be a normal program. Using Theorem 3 (Delgrande and Wang 2015), the result of forgetting about a_1 from P is equivalent to $P'=\{h_1\vee h_2\leftarrow not\,h_3,\ h_2\leftarrow not\,a_2\}$. Using the characterization result for a set of HT-models of normal programs (Eiter et al. 2004) it not difficult to see that P' is not strongly equivalent to any normal program: just note that both $\langle\{h_1\},\{h_1,h_2,a_2\}\rangle$ and $\langle\{h_2\},\{h_1,h_2,a_2\}\rangle$ are HT-models of P', but the so-called Here-intersection $\langle\{h_1\}\cap\{h_2\},\{h_1,h_2,a_2\}\rangle=\langle\emptyset,\{h_1,h_2,a_2\}\rangle$ is not. □

Theorem 15

F_W satisfies **(sC)**, **(wE)**, **(SE)**, **(PP)**, **(SI)**, **(SC)**, **(RC)**, **(NC)**, **(PI)**, **(sSP)**, **(UI)**, $(\mathbf{SI_u})$, $(\mathbf{E}_{\mathcal{C}_H})$, $(\mathbf{E}_{\mathcal{C}_n})$, $(\mathbf{E}_{\mathcal{C}_d})$, but not **(W)**, **(CP)**, **(SP)**, **(wC)**, **(wSP)** and **(UP)**.

Proof

In the case of **(sC)**, the result follows immediately from Lemma 3. For **(wE)**, let $\mathsf{f}\in\mathsf{F}_W$, P_1,P_2 disjunctive programs over $\mathcal{A}$, and $V\subseteq\mathcal{A}$. Suppose $\mathcal{AS}(P_1)=\mathcal{AS}(P_2)$. We aim to prove that $\mathcal{AS}(\mathsf{f}(P_1,V))=\mathcal{AS}(\mathsf{f}(P_2,V))$. The result follows from Lemma 3 and the assumption $\mathcal{AS}(P_1)=\mathcal{AS}(P_2)$, since $\mathcal{AS}(\mathsf{f}(P_1,V))=\{X\in\mathcal{AS}(P_1):V\cap X=\emptyset\}=\{X\in\mathcal{AS}(P_2):V\cap X=\emptyset\}=\mathcal{AS}(\mathsf{f}(P_2,V))$.

Wong (2009) proved that F_W satisfies **(SE)** (Lemma 4.27), and it is implicit that it satisfies $(\mathbf{E}_{\mathcal{C}_d})$.

The fact that **(PP)** holds follows easily from the definition of F_W. Note that, by definition, $r\in\mathsf{f}_W(P,v)$ for every disjunctive rule r not containing v such that $P\models_{\mathsf{HT}}r$. Therefore, for every disjunctive rule r not containing v and such that $P\models_{\mathsf{HT}}r$ we have that $\mathsf{f}_W(P,v)\models_{\mathsf{HT}}r$.

For **(SI)**, let $\mathsf{f}\in\mathsf{F}_W$, P be a program over a signature $\mathcal{A}$, $V\subseteq\mathcal{A}$, and R a program over $\mathcal{A}\setminus V$. We aim to prove that $\mathsf{f}(P\cup R,V)\equiv_{\mathsf{HT}}\mathsf{f}(P,V)\cup R$. Consider the following sequence of equalities:

$$
\begin{aligned}
\mathcal{HT}_{\|V}(\mathsf{f}(P\cup R,V)) &= \mathcal{HT}((P\cup R)\cup\{\leftarrow v:v\in V\})\\
&= \mathcal{HT}((P\cup\{\leftarrow v:v\in V\})\cup R)\\
&= \mathcal{HT}((P\cup\{\leftarrow v:v\in V\}))\cap\mathcal{HT}(R)\\
&= \mathcal{HT}_{\|V}(\mathsf{f}(P,V))\cap\mathcal{HT}(R)\\
&= \mathcal{HT}_{\|V}(\mathsf{f}(P,V)\cup R)
\end{aligned}
$$

The first and fourth equalities follow from Lemma 2. The second and the third are well-known properties of HT-models. The fifth equality follows from the fact that R does not contain any $v\in V$ and Lemma 6. We then have that $\mathcal{HT}_{\|V}(\mathsf{f}(P\cup R,V))=\mathcal{HT}_{\|V}(\mathsf{f}(P,V)\cup R)$. Since both $\mathsf{f}(P\cup R,V)$ and $\mathsf{f}(P,V)\cup R$ do not contain any $v\in V$, the above equality immediately entails that $\mathcal{HT}(\mathsf{f}(P\cup R,V))=\mathcal{HT}(\mathsf{f}(P,V)\cup R)$.

The fact that F_W satisfies **(SC)**, **(RC)**, **(NC)** and **(PI)** was proved by Gonçalves et al. (2017).

Property **(sSP)** follows from Thm. 1 and the fact that F_W satisfies **(sC)** and **(SI)**.

In the case of **(UI)** the result follows from the already proved fact that F_W satisfies **(SI)**, and item 15 of Thm. 1.

The fact that F_W satisfies $(\mathbf{SI_u})$ follows from the already proved fact that F_W satisfies **(SI)**, and item 17 of Thm. 1.

For $(\mathbf{E}_{\mathcal{C}_H})$, let $\mathsf{f} \in \mathsf{F}_W$ and P a Horn program. Using Lemma 2 we have that $\mathcal{HT}_{\|V}(\mathsf{f}(P,V)) = \mathcal{HT}(P \cup \{\leftarrow v : v \in V\})$. Since $P \cup \{\leftarrow v : v \in V\}$ is a normal program, it can be easily shown that $\mathcal{HT}(\mathsf{f}(P,V)) = (\mathcal{HT}_{\|V}(\mathsf{f}(P,V)))_{\dagger V}$ satisfies all conditions characterizing a class of HT-models of a Horn program (Wang et al. 2014), taking into account that $\mathcal{HT}(P \cup \{\leftarrow v : v \in V\})$ satisfies these conditions.

For $(\mathbf{E}_{\mathcal{C}_n})$, let $\mathsf{f} \in \mathsf{F}_W$ and P a normal program. Using Lemma 1 we have that $\mathcal{HT}_{\|V}(\mathsf{f}(P,v)) = \mathcal{HT}(P \cup \{\leftarrow v\})$. Since $P \cup \{\leftarrow v\}$ is a normal program, it can be easily shown that $\mathcal{HT}(\mathsf{f}(P,v)) = (\mathcal{HT}_{\|V}(\mathsf{f}(P,v)))_{\dagger v}$ satisfies all conditions characterizing a class of HT-models of a normal program (Eiter et al. 2004), taking into account that $\mathcal{HT}(P \cup \{\leftarrow v\})$ satisfies these conditions.

For the negative results, Wang et al. (2012) provided counterexamples showing that F_W does not satisfy $(\mathbf{W})$. The negative results for **(CP)** and **(SP)** can be illustrated with forgetting about b from $P = \{a \leftarrow \mathit{not}\, b,\ b \leftarrow \mathit{not}\, a\}$. Since $\mathcal{AS}(P) = \{\{a\},\{b\}\}$, the result must have two answer sets $\{a\}$ and $\emptyset$, which is not possible for disjunctive programs obtained from operators in F_W.

The counterexample for **(wC)** is similar to other cases, just by considering forgetting about b from $P = \{a \leftarrow \mathit{not}\, b,\ b \leftarrow \mathit{not}\, a\}$. Since $\mathcal{AS}(P) = \{\{a\},\{b\}\}$, to satisfy **(wC)**, the result must have the two answer sets $\{a\}$ and $\emptyset$, which is not possible for disjunctive programs. Therefore, F_W does not satisfy **(wC)**, which also implies that it does not satisfy **(wSP)** by Thm. 1. □

Theorem 16

F_{HT} satisfies **(SE)**, **(W)**, **(PP)**, **(SI)**, **(SC)**, **(RC)**, **(NC)**, **(PI)**, **(UI)**, $(\mathbf{SI_u})$, $(\mathbf{E}_{\mathcal{C}_H})$, $(\mathbf{E}_{\mathcal{C}_e})$, but not **(sC)**, **(wE)**, **(CP)**, **(SP)**, **(wC)**, **(wSP)**, **(sSP)**, **(UP)**, $(\mathbf{E}_{\mathcal{C}_n})$ and $(\mathbf{E}_{\mathcal{C}_d})$.

Proof

Wang et al. (2012) proved that F_{HT} satisfies **(SE)** (Proposition 3), **(W)**, **(PP)** (Theorem 3), $(\mathbf{E}_{\mathcal{C}_H})$ (Theorem 2), and $(\mathbf{E}_{\mathcal{C}_e})$ (Theorem 1).

For **(SI)**, let $\mathsf{f} \in \mathsf{F}_{\mathsf{HT}}$, P be a program over a signature $\mathcal{A}$, $V \subseteq \mathcal{A}$, and R a program over $\mathcal{A} \setminus V$. We aim to prove that $\mathsf{f}(P,V) \cup R \equiv_{\mathsf{HT}} \mathsf{f}(P \cup R, V)$, which is the same as $\mathcal{HT}(\mathsf{f}(P,V) \cup R) = \mathcal{HT}(\mathsf{f}(P \cup R, V))$. Consider the following sequence of equalities.

$$\begin{aligned}
\mathcal{HT}(\mathsf{f}(P,V) \cup R) &= \mathcal{HT}(\mathsf{f}(P,V)) \cap \mathcal{HT}(R) \\
&= (\mathcal{HT}_{\|V}(P))_{\dagger V} \cap (\mathcal{HT}_{\|V}(R))_{\dagger V} \\
&= (\mathcal{HT}_{\|V}(P) \cap \mathcal{HT}_{\|V}(R))_{\dagger V} \\
&= (\mathcal{HT}_{\|V}(P \cup R))_{\dagger V} \\
&= \mathcal{HT}(\mathsf{f}(P \cup R, V))
\end{aligned}$$

The first equality follows from a well-known property of HT-models. The second equality follows from the definition of HT-forgetting. The third equality follows from Lemma 8, while the fourth equality follows from Lemma 6. Finally, the last equality also follows from the definition of HT-forgetting.

The fact that F_{HT} satisfies **(SC)**, **(RC)**, **(NC)** and **(PI)** was proved by Gonçalves et al. (2017).

In the case of **(UI)** the result follows from the already proved fact that F_{HT} satisfies **(SI)**, and item 15 of Thm. 1.

The fact that F_{HT} satisfies $(\mathbf{SI_u})$ follows from the already proved fact that F_{HT} satisfies **(SI)**, and item 17 of Thm. 1.

For the negative results, in the case of **(sC)**, consider forgetting about a from $P = \{a \leftarrow not\, a\}$. The result should be strongly equivalent to $\emptyset$, i.e., the forgetting operation introduces a new answer-set. Turning to **(wE)**, this property requires that the results of forgetting about p from $P = \{q \leftarrow not\, p, q \leftarrow not\, q\}$ and from $Q = \{q \leftarrow\}$ have the same answer sets, while F_{HT} requires that the results be strongly equivalent to $\mathsf{f}(P,p) = \{q \leftarrow not\, q\}$ and $\mathsf{f}(Q,p) = \{q \leftarrow\}$, which are obviously not equivalent.

The fact that F_{HT} does not satisfy **(CP)** and **(SP)** follows immediately from the fact that it does not satisfy **(sC)**, and items 3. and 4. of Thm. 1.

For **(wC)**, consider forgetting about b from $P = \{a \leftarrow not\, b,\ b \leftarrow not\, a\}$. Since $\mathcal{AS}(P) = \{\{a\},\{b\}\}$, to satisfy **(wC)**, the result must have the two answer sets $\{a\}$ and $\emptyset$, but the result of $\mathsf{f}(P,b)$ for $\mathsf{f} \in \mathsf{F}_{\mathsf{HT}}$ is equivalent to the empty program. Therefore, the answer set $\{a\}$ is not preserved, and thus F_{HT} does not satisfy **(wC)**.

The fact that F_{HT} does not satisfy **(wSP)** and **(sSP)** follows immediately from the fact that it does not satisfy **(sC)** and **(wC)**, and items 8. and 9. of Thm. 1.

Gonçalves et al. (2019) showed that F_{HT} does not satisfy **(UP)**. Wang (2012) provided counterexamples showing that F_{HT} does not satisfy $(\mathbf{E}_{\mathcal{C}_n})$ and $(\mathbf{E}_{\mathcal{C}_d})$. □

Theorem 17
F_{SM} satisfies **(sC)**, **(wE)**, **(SE)**, **(PP)**, **(PI)**, **(CP)**, **(wC)**, $(\mathbf{E}_{\mathcal{C}_H})$, $(\mathbf{E}_{\mathcal{C}_e})$, but not **(W)**, **(SI)**, **(SC)**, **(RC)**, **(NC)**, **(SP)**, **(wSP)**, **(sSP)**, **(UP)**, **(UI)**, $(\mathbf{SI_u})$, $(\mathbf{E}_{\mathcal{C}_n})$ and $(\mathbf{E}_{\mathcal{C}_d})$.

Proof
By definition, F_{SM} satisfies **(CP)**. From this and Thm. 1 it follows easily that F_{SM} satisfies **(sC)**, **(wC)** and **(wE)**. Wang et al. (2013) have shown that F_{SM} satisfies **(SE)**, **(PP)**, $(\mathbf{E}_{\mathcal{C}_e})$ (Proposition 1), and $(\mathbf{E}_{\mathcal{C}_H})$ (Theorem 3), and the fact that F_{SM} satisfies **(PI)** was proved by Gonçalves et al. (2017).

For the negative results, Wang et al. (2013) pointed out that F_{SM} does not satisfy **(W)**. In the case of **(SI)**, consider forgetting about b from $P = \{a \leftarrow not\, b,\ b \leftarrow not\, c\}$. In this case we have that $\mathsf{f}(P,b) \equiv_{\mathsf{HT}} \emptyset$ for $\mathsf{f} \in \mathsf{F}_{\mathsf{SM}}$, so adding $c \leftarrow$ results precisely in a program containing this fact. If we add $c \leftarrow$ before forgetting, then the HT-models of the result of forgetting, ignoring all occurrences of b, correspond precisely to $\langle\{c\},\{c\}\rangle, \langle\{c\},\{a,c\}\rangle$, and $\langle\{a,c\},\{a,c\}\rangle$. To preserve the answer sets, only the last of these three can be considered. Hence, $a \leftarrow$ and $c \leftarrow$ (or strongly equivalent rules) occur in the result of forgetting for any $\mathsf{f} \in \mathsf{F}_{\mathsf{SM}}$, and **(SI)** does not hold. Since this counterexample only adds a fact before and after forgetting, it also shows that **(UI)** does not hold.

The fact that F_{SM} does not satisfy **(SC)**, **(RC)**, **(NC)** was proved by Gonçalves et al. (2017). Knorr and Alferes (2014) provided a counterexample showing that F_{SM} does not satisfy **(SP)**.

Gonçalves et al. (2020) showed that F_{SM} does not satisfy **(wSP)** and **(sSP)**, and that it does not satisfy **(UP)** (2019). For the case of $(\mathbf{SI_u})$, consider a program $P = \{a \leftarrow not\, b, b \leftarrow not\, c\}$. We can easily check that for any $\mathsf{f} \in \mathsf{F}_{\mathsf{SM}}$, we have that $\mathsf{f}(P,b) \equiv_{\mathsf{HT}} \{\}$. We can then conclude that $\mathcal{AS}(\mathsf{f}(P,b) \cup \{c \leftarrow\}) = \{\{c\}\}$. On the other hand, since F_{SM} satisfies **(CP)**, we have that $\mathcal{AS}(\mathsf{f}(P \cup \{c \leftarrow\}, b)) = \mathcal{AS}(P \cup \{c \leftarrow\})_{\|\{b\}} = \{\{a,c\}\}$. But then, $\mathcal{AS}(\mathsf{f}(P,b) \cup \{c \leftarrow\}) =$

$\{\{c\}\} \neq \{\{a,c\}\} = \mathcal{AS}(\mathsf{f}(P \cup \{c \leftarrow\}, b))$, and we can conclude that $\mathsf{f}(P,b) \cup \{c \leftarrow\} \not\equiv_u \mathsf{f}(P \cup \{c \leftarrow\}, b)$, showing that F_{SM} does not satisfy $(\mathbf{SI_u})$.

Wang et al. (2013) provided counterexamples showing that F_{SM} does not satisfy $(\mathbf{E}_{\mathcal{C}_n})$ and $(\mathbf{E}_{\mathcal{C}_d})$. □

Theorem 18
F_{Sas} satisfies **(sC)**, **(wE)**, **(SE)**, **(W)**, **(PP)**, **(SI)**, **(SC)**, **(RC)**, **(NC)**, **(PI)**, **(CP)**, **(SP)**, **(wC)**, **(wSP)**, **(sSP)**, **(UP)**, **(UI)**, $(\mathbf{SI_u})$, and $(\mathbf{E}_{\mathcal{C}_H})$.

Proof
By definition of the class, **(SP)** is satisfied. From this fact it easily follows from the results in Thm. 1 that F_{Sas} satisfies **(sC)**, **(wE)**, **(SE)**, **(PP)**, **(SI)**, **(CP)**, **(SP)**, **(wC)**, **(wSP)**, **(sSP)**, **(UP)**, **(UI)**, $(\mathbf{SI_u})$.

For **(W)**, **(SC)**, **(RC)**, **(NC)** and **(PI)**, first note that Thm. 1 (Gonçalves et al. 2020) states that there is no operator over a class of programs that contains normal programs and that satisfies **(SP)**. Therefore, every operator in F_{Sas} is necessarily defined over $\mathcal{C}_H$. Thm. 2 states that when restricted to Horn programs the result of operators in F_{Sas} is strongly equivalent to the result of operators in the class F_S. Since Thm. 14 states that F_S satisfies **(W)**, **(SC)**, **(RC)**, **(NC)**, **(PI)** and $(\mathbf{E}_{\mathcal{C}_H})$, so does F_{Sas}. □

Theorem 19
F_{SE} satisfies **(SE)**, **(W)**, **(PP)**, **(SC)**, **(RC)**, **(NC)**, **(PI)**, $(\mathbf{E}_{\mathcal{C}_H})$, $(\mathbf{E}_{\mathcal{C}_d})$, but not **(sC)**, **(wE)**, **(SI)**, **(CP)**, **(SP)**, **(wC)**, **(wSP)**, **(sSP)**, **(UP)**, **(UI)**, $(\mathbf{SI_u})$ and $(\mathbf{E}_{\mathcal{C}_n})$.

Proof
The results follow from Thm. 3 and Thm. 14. □

Theorem 20
F_{SP} satisfies **(SE)**, **(PP)**, **(SI)**, **(wC)**, **(wSP)**, **(UI)**, $(\mathbf{SI_u})$, $(\mathbf{E}_{\mathcal{C}_H})$, $(\mathbf{E}_{\mathcal{C}_e})$, but not **(sC)**, **(wE)**, **(W)**, **(SC)**, **(RC)**, **(NC)**, **(PI)**, **(CP)**, **(SP)**, **(sSP)**, **(UP)**, $(\mathbf{E}_{\mathcal{C}_n})$ and $(\mathbf{E}_{\mathcal{C}_d})$.

Proof
Gonçalves et al. (2020) showed that F_{SP} satisfies **(SE)**, **(PP)**, **(SI)**, **(wC)**, **(wSP)**, $(\mathbf{E}_{\mathcal{C}_H})$ and $(\mathbf{E}_{\mathcal{C}_e})$. Property **(UI)** follows from item 15 of Thm. 1 and the fact that F_{SP} satisfies **(SI)**.

The fact that F_{SP} satisfies $(\mathbf{SI_u})$ follows from the already proved fact that F_{SP} satisfies **(SI)**, and item 17 of Thm. 1.

The negative results for **(sC)**, **(wE)**, **(W)**, **(CP)**, **(SP)**, **(sSP)**, $(\mathbf{E}_{\mathcal{C}_n})$ and $(\mathbf{E}_{\mathcal{C}_d})$ were proven by Gonçalves et al. (2020).

For **(SC)**, consider the program $P = \{a \leftarrow \mathit{not}\, p,\ p \leftarrow \mathit{not}\, a\}$. We have that $P \models_{\mathsf{HT}} a \leftarrow \mathit{not}\, p$, but $\mathsf{f}(P,\{p\}) \not\models_{\mathsf{HT}} \mathsf{f}(\{a \leftarrow \mathit{not}\, p\}, p)$, since $\mathsf{f}(P,\{p\})$ is strongly equivalent to $\{a \leftarrow \mathit{not}\,\mathit{not}\, a\}$ and $\mathsf{f}(\{a \leftarrow \mathit{not}\, p\}, \{p\})$ is strongly equivalent to $\{a \leftarrow\}$.

The fact that F_{SP} does not satisfy **(RC)** follows from item 14. of Thm. 1 and the fact that F_{SP} satisfies **(wC)** and **(UI)**.

For **(NC)** consider the program $P = \{a \leftarrow p; p \leftarrow \mathit{not}\,\mathit{not}\, p\}$. Then, for $\mathsf{f} \in \mathsf{F}_{\mathsf{SP}}$ we have that $\mathsf{f}(P,\{p\}) \equiv_{\mathsf{HT}} \{a \leftarrow \mathit{not}\,\mathit{not}\, a\}$. Therefore, $\mathsf{f}(P,\{p\}) \models_{\mathsf{HT}} a \leftarrow \mathit{not}\,\mathit{not}\, a$, but it is not the case that $P \models_{\mathsf{HT}} a \leftarrow \mathit{not}\,\mathit{not}\, a, \mathit{not}\, p$.

The fact that F_{SP} does not satisfy **(PI)** follows from the result given in (Gonçalves et al. 2017) about the impossibility of iteration while satisfying **(SP)**. The counterexample for **(UP)** was presented by Gonçalves et al. (2019). □

Theorem 21
F_{R} satisfies **(sC)**, **(SE)**, **(PP)**, **(SI)**, **(RC)**, **(NC)**, **(sSP)**, **(UI)**, $(\mathbf{SI_u})$, $(\mathbf{E}_{\mathcal{C}_H})$, $(\mathbf{E}_{\mathcal{C}_e})$, but not **(wE)**, **(W)**, **(SC)**, **(PI)**, **(CP)**, **(SP)**, **(wC)**, **(wSP)**, **(UP)**, $(\mathbf{E}_{\mathcal{C}_n})$ and $(\mathbf{E}_{\mathcal{C}_d})$.

Proof
Gonçalves et al. (2020) showed that F_{R} satisfies **(sC)**, **(SE)**, **(PP)**, **(SI)**, **(sSP)**, $(\mathbf{E}_{\mathcal{C}_H})$ and $(\mathbf{E}_{\mathcal{C}_e})$.

To show that F_{R} satisfies **(NC)**, let $r = A \leftarrow B, not\, C, not\, not\, D$ be a rule such that $\mathsf{f}(P,V) \models_{\mathsf{HT}} r$. Then, $\mathcal{HT}(\mathsf{f}(P,V)) \subseteq \mathcal{HT}(r)$. Let $r' = A \leftarrow B, not\, C, not\, V, not\, not\, D$. Then it is clear that $\mathcal{HT}(r) \subseteq \mathcal{HT}(r')$, and as a consequence we can conclude that $\mathcal{HT}(\mathsf{f}(P,V)) \subseteq \mathcal{HT}(r')$. Consider two cases: first let $\langle X,Y\rangle \in \mathcal{HT}(P)$ such that $Y \cap V \neq \emptyset$. In this case, $\langle X,Y\rangle \in \mathcal{HT}(r')$ since $Y \models r'$ and $\{r'\}^Y = \emptyset$. Now take $\langle X,Y\rangle \in \mathcal{HT}(P)$ such that $Y \cap V = \emptyset$. By definition of F_{R}, we have that $\langle X,Y\rangle \in \mathcal{HT}(\mathsf{f}(P,V))$, and therefore $\langle X,Y\rangle \in \mathcal{HT}(r')$. In both cases, $\langle X,Y\rangle \in \mathcal{HT}(r')$, and therefore $P \models_{\mathsf{HT}} r'$.

To prove **(RC)**, let $r = A \leftarrow B, not\, C, not\, not\, D$ be a rule such that $\mathsf{f}(P,V) \models_{\mathsf{HT}} r$. Let $r' = A \leftarrow B, not\, C, not\, V, not\, not\, D$. We can easily prove that if $\langle X,Y\rangle \in \mathcal{HT}(r')$ and $Y \cap V = \emptyset$ then $\langle X,Y\rangle \in \mathcal{HT}(r)$. Now, since F_{R} satisfies **(NC)**, we can conclude that $P \models_{\mathsf{HT}} r'$. We aim to prove that $\mathsf{f}(\{r'\},V) \models_{\mathsf{HT}} r$. For that, let $\langle X,Y\rangle \in \mathcal{HT}(\mathsf{f}(\{r'\},V))$. Then, there exists $A \in Rel^Y_{\langle\{r'\},V\rangle}$ such that $\langle X \cup X', Y \cup A\rangle \in \mathcal{HT}(r')$, where $X' \subseteq A$. We now prove that $A = \emptyset$. Suppose not. Then, since $\{r'\}^{Y\cup A} = \{\}$, we have that $\langle Y, Y \cup A\rangle$ is also a model of r', which contradicts the fact that $A \in Rel^Y_{\langle\{r'\},V\rangle}$. Since A must be empty we can conclude that $\langle X,Y\rangle \in \mathcal{HT}(r')$. Since $Y \cap V = \emptyset$, the above consideration entails that $\langle X,Y\rangle \in \mathcal{HT}(r)$, which completes the proof.

Property **(UI)** follows from item 15 of Thm. 1 and the fact that F_{R} satisfies **(SI)**.

Property $(\mathbf{SI_u})$ follows from item 17 of Thm. 1 and the fact that F_{R} satisfies **(SI)**.

The negative results for **(wE)**, **(W)**, **(PI)**, **(CP)**, **(SP)**, **(wC)**, **(wSP)**, $(\mathbf{E}_{\mathcal{C}_n})$ and $(\mathbf{E}_{\mathcal{C}_d})$ are due to Gonçalves et al. (2020).

For **(SC)**, consider the program $P = \{a \leftarrow not\, p,\ p \leftarrow not\, a\}$. We have that $P \models_{\mathsf{HT}} a \leftarrow not\, p$, but $\mathsf{f}(P,\{p\}) \not\models_{\mathsf{HT}} \mathsf{f}(\{a \leftarrow not\, p\}, p)$, since $\mathcal{HT}(\mathsf{f}(P,\{p\}))$ contains $\langle\emptyset,\emptyset\rangle$, therefore $\mathsf{f}(P,\{p\}) \not\models_{\mathsf{HT}} a \leftarrow$, but $\mathsf{f}(\{a \leftarrow not\, p\},\{p\})$ is strongly equivalent to $\{a \leftarrow\}$.

The fact that F_{R} does not satisfy **(PI)** follows from the result about the impossibility of iteration while satisfying **(SP)** (Gonçalves et al. 2017).

The counterexample for **(UP)** was given by Gonçalves et al. (2019). □

Theorem 22
F_{M} satisfies **(sC)**, **(wE)**, **(SE)**, **(PP)**, **(CP)**, **(wC)**, **(sSP)**, $(\mathbf{E}_{\mathcal{C}_H})$, $(\mathbf{E}_{\mathcal{C}_e})$, but not **(W)**, **(SI)**, **(SC)**, **(RC)**, **(NC)**, **(PI)**, **(SP)**, **(wSP)**, **(UP)**, **(UI)**, $(\mathbf{SI_u})$, $(\mathbf{E}_{\mathcal{C}_n})$ and $(\mathbf{E}_{\mathcal{C}_d})$.

Proof
Gonçalves et al. (2020) showed that F_{M} satisfies **(sC)**, **(wE)**, **(SE)**, **(PP)**, **(CP)**, **(wC)**, **(sSP)**, $(\mathbf{E}_{\mathcal{C}_H})$, $(\mathbf{E}_{\mathcal{C}_e})$, as well as the negative results for **(W)**, **(SI)**, **(SP)**, **(wSP)**, $(\mathbf{E}_{\mathcal{C}_n})$ and $(\mathbf{E}_{\mathcal{C}_d})$.

In the case of **(SC)**, consider the program $P = \{a \leftarrow not\, p,\ p \leftarrow not\, a\}$. We have that $P \models_{\mathsf{HT}} a \leftarrow not\, p$, but $\mathsf{f}(P,\{p\}) \not\models_{\mathsf{HT}} \mathsf{f}(\{a \leftarrow not\, p\}, p)$, since $\mathsf{f}(P,\{p\})$ is strongly equivalent to $\{a \leftarrow not\, not\, a\}$ and $\mathsf{f}(\{a \leftarrow not\, p\},\{p\})$ is strongly equivalent to $\{a \leftarrow\}$.

In the case of **(RC)** consider forgetting about p from the program $P = \{a \leftarrow p,\ p \leftarrow \mathit{not\, not\, p}\}$. We have that $\mathsf{f}(P,\{p\})$ is strongly equivalent to $\{a \leftarrow \mathit{not\, not\, a}\}$. Nevertheless, there is no rule r' over a and p such that $P \models_{\mathsf{HT}} r'$ and $\mathsf{f}(\{r'\},\{p\}) \models_{\mathsf{HT}} a \leftarrow \mathit{not\, not\, a}$.

For **(NC)** consider the program $P = \{a \leftarrow p; p \leftarrow \mathit{not\, not\, p}\}$. Then, for $\mathsf{f} \in \mathsf{F_M}$ we have that $\mathsf{f}(P,\{p\}) \equiv_{\mathsf{HT}} \{a \leftarrow \mathit{not\, not\, a}\}$. Therefore, $\mathsf{f}(P,\{p\}) \models_{\mathsf{HT}} a \leftarrow \mathit{not\, not\, a}$, but it is not the case that $P \models_{\mathsf{HT}} a \leftarrow \mathit{not\, not\, a}, \mathit{not\, p}$.

The fact that $\mathsf{F_M}$ does not satisfy **(PI)** follows from the result about the impossibility of iteration while satisfying **(SP)** (Gonçalves et al. 2017). The counterexample for **(UP)** was presented by Gonçalves et al. (2019).

The negative result for **(UI)** follows from item 16. of Thm. 1 and the fact that $\mathsf{F_M}$ satisfies **(CP)** but not **(UP)**.

In the case of $(\mathbf{SI_u})$, consider a program P over $\mathcal{A} = \{a,b,p\}$ such that $\mathcal{HT}(P) = \{\langle ab,ab\rangle, \langle a,ab\rangle, \langle abp,abp\rangle, \langle b,abp\rangle, \langle a,a\rangle, \langle \emptyset,a\rangle, \langle ap,ap\rangle\}$. By definition, for $\mathsf{f} \in \mathsf{F_M}$, we have that forgetting p is such that $\mathcal{HT}(\mathsf{f}(P,p)) = \{\langle ab,ab\rangle, \langle b,ab\rangle, \langle a,ab\rangle, \langle a,a\rangle\}$. Consider R over $\mathcal{A}\backslash\{p\}$ such that $\mathcal{HT}(R) = \{\langle ab,ab\rangle, \langle b,ab\rangle\}$. Then, $\mathcal{HT}(P \cup R) = \{\langle ab,ab\rangle, \langle abp,abp\rangle, \langle b,abp\rangle\}$. Again by definition, for $\mathsf{f} \in \mathsf{F_M}$, we have that $\mathcal{HT}(\mathsf{f}(P \cup R,p)) = \{\langle ab,ab\rangle\}$. On the other hand, $\mathcal{HT}(\mathsf{f}(P,p) \cup R) = \{\langle ab,ab\rangle, \langle b,ab\rangle\}$. This then implies that $\mathcal{AS}(\mathsf{f}(P \cup R,p)) = \{\{a,b\}\} \neq \{\} = \mathcal{AS}(\mathsf{f}(P,p) \cup R)$, which means that $\mathsf{f}(P \cup R,p) \not\equiv_u \mathsf{f}(P,p) \cup R$, thus showing that $\mathsf{F_M}$ does not satisfy $(\mathbf{SI_u})$. □

Theorem 23
$\mathsf{F_{UP}}$ satisfies **(sC)**, **(wE)**, **(SE)**, **(PI)**, **(CP)**, **(wC)**, **(UP)**, **(UI)**, $(\mathbf{E}_{\mathcal{C}_H})$, $(\mathbf{E}_{\mathcal{C}_e})$, but not **(W)**, **(PP)**, **(SI)**, **(SC)**, **(RC)**, **(NC)**, **(SP)**, **(wSP)**, **(sSP)**, $(\mathbf{SI_u})$, $(\mathbf{E}_{\mathcal{C}_n})$ and $(\mathbf{E}_{\mathcal{C}_d})$.

Proof
Gonçalves et al. (2019) showed that $\mathsf{F_{UP}}$ satisfies **(sC)**, **(wE)**, **(SE)**, **(PI)**, **(CP)**, **(wC)**, **(UP)**, **(UI)** $(\mathbf{E}_{\mathcal{C}_H})$, $(\mathbf{E}_{\mathcal{C}_e})$, and that it does not satisfy **(W)**, **(PP)**, **(SI)**, **(SP)**, $(\mathbf{E}_{\mathcal{C}_n})$ and $(\mathbf{E}_{\mathcal{C}_d})$. Also, a weaker version of **(PI)** was proved, showing that it is possible to iterate the operators when applied in the context of modular logic programming. Gonçalves et al. (2021) proved that $\mathsf{F_{UP}}$ does not satisfy $(\mathbf{SI_u})$.

In the case of **(SC)**, consider the program $P = \{a \leftarrow \mathit{not\, p},\ p \leftarrow \mathit{not\, a}\}$. We have that $P \models_{\mathsf{HT}} a \leftarrow \mathit{not\, p}$, but $\mathsf{f}(P,\{p\}) \not\models_{\mathsf{HT}} \mathsf{f}(\{a \leftarrow \mathit{not\, p}\},p)$, since $\mathsf{f}(P,\{p\})$ is strongly equivalent to $\{a \leftarrow \mathit{not\, not\, a}\}$ and $\mathsf{f}(\{a \leftarrow \mathit{not\, p}\},\{p\})$ is strongly equivalent to $\{a \leftarrow\}$.

The negative result for **(RC)** follows from item 14. of Thm. 1 and the fact that $\mathsf{F_{UP}}$ satisfies **(wC)** and **(UI)**.

For **(NC)** consider the program $P = \{a \leftarrow p; p \leftarrow \mathit{not\, not\, p}\}$. Then, for $\mathsf{f} \in \mathsf{F_{UP}}$ we have that $\mathsf{f}(P,\{p\}) \equiv_{\mathsf{HT}} \{a \leftarrow \mathit{not\, not\, a}\}$. Therefore, $\mathsf{f}(P,\{p\}) \models_{\mathsf{HT}} a \leftarrow \mathit{not\, not\, a}$, but it is not the case that $P \models_{\mathsf{HT}} a \leftarrow \mathit{not\, not\, a}, \mathit{not\, p}$.

For **(wSP)**, consider P such that $\mathcal{HT}(P) = \{\langle ab,ab\rangle, \langle a,ab\rangle, \langle b,abp\rangle, \langle abp,abp\rangle, \langle a,abp\rangle\}$. Then, by definition $\mathcal{HT}(\mathsf{f}(P,\{p\})) = \{\langle ab,ab\rangle, \langle a,ab\rangle, \langle \emptyset,ab\rangle\}$. Consider a program R over $\{a,b\}$ such that $\mathcal{HT}(R) = \{\langle ab,ab\rangle, \langle \emptyset,ab\rangle\}$ (over $\{a,b\}$). Then, $\mathcal{HT}(P \cup R) = \mathcal{HT}(P) \cap \mathcal{HT}(R) = \{\langle abp,abp\rangle, \langle ab,ab\rangle\}$, thus $\{a,b\} \in \mathcal{AS}(P \cup R)_{\parallel V}$, but $\{a,b\} \notin \mathcal{AS}(\mathsf{f}(P,V) \cup R)$.

For **(sSP)**, consider P such that $\mathcal{HT}(P) = \{\langle ab,ab\rangle, \langle a,ab\rangle, \langle b,abp\rangle, \langle abp,abp\rangle\}$. Then, by definition $\mathcal{HT}(\mathsf{f}(P,\{p\})) = \{\langle ab,ab\rangle, \langle \emptyset,ab\rangle\}$. Consider a program R over $\{a,b\}$ s.t. $\mathcal{HT}(R) = \{\langle ab,ab\rangle, \langle a,ab\rangle, \langle b,ab\rangle\}$ (over $\{a,b\}$). Then $\mathsf{f}(P,\{p\}) \cup R$ has an answer set $\{a,b\}$, but this is not an answer set of $\mathsf{f}(P \cup R,\{p\})$. □

Lemma 1
Let P be a disjunctive program over a signature $\mathcal{A}$ and $v \in \mathcal{A}$. Then, for every $\mathsf{f} \in \mathsf{F}_W$ we have

$$\mathcal{HT}_{\|\{v\}}(\mathsf{f}(P,v)) = \mathcal{HT}(P \cup \{\leftarrow v\}).$$

Proof
Let P be a disjunctive program over a signature $\mathcal{A}$, and $v \in \mathcal{A}$. We prove the equality for f_W and the result extends to every $\mathsf{f} \in \mathsf{F}_W$ since these only differ from f_W up to strong equivalence. We prove both inclusions of the equality $\mathcal{HT}_{\|\{v\}}(\mathsf{f}_W(P,v)) = \mathcal{HT}(P \cup \{\leftarrow v\})$. In what follows let $A,B,C \subseteq \mathcal{A} \setminus \{v\}$.

We start with the left to right inclusion. Let $I = \langle X,Y \rangle$ such that $I \in \mathcal{HT}_{\|\{v\}}(\mathsf{f}_W(P,v))$. Since v does not occur in $\mathsf{f}(P,v)$ we have that $I \in \mathcal{HT}(\mathsf{f}_W(P,v))$. Since I does not contain v, then $I \models_{\mathsf{HT}} \leftarrow v$. We now show that $I \models_{\mathsf{HT}} r$, for every $r \in P$. If $r \in P$ such that it does not contain v, then clearly $r \in \mathsf{f}(P,v)$, and so $I \models_{\mathsf{HT}} r$. If $r \in P$ and it contains v, then we have the following cases:

- If r is a tautology, then clearly $I \models_{\mathsf{HT}} r$;
- If $v \in body(r)$, then, since I does not contain v, trivially $I \models_{\mathsf{HT}} r$;
- If $r = A, v \leftarrow B, not\, v, not\, C$ then the rule $r^* = A \leftarrow B, not\, C \in \mathsf{f}_W(P,v)$. Suppose $I \not\models_{\mathsf{HT}} r$. Then we have two cases:
  - — $Y \not\models r$, which is an absurd since $v \notin Y$ and $I \models_{\mathsf{HT}} r^*$;
  - — $Y \cap C = \emptyset$ and $X \not\models A, v \leftarrow B$. This is an absurd since $v \notin X$, ${r^*}^Y = A, v \leftarrow B$, and $X \models {r^*}^Y$.
- If $r = A, v \leftarrow B, not\, C$ then $r^* = A \leftarrow B, not\, C \in \mathsf{f}_W(P,v)$. Suppose $I \not\models_{\mathsf{HT}} r$. Then we have two cases:
  - — $Y \not\models r$, which is an absurd since $v \notin Y$ and $I \models_{\mathsf{HT}} r^*$;
  - — $Y \cap C = \emptyset$ and $X \not\models A, v \leftarrow B$. This is an absurd since $v \notin X$, ${r^*}^Y = A \leftarrow B$, and $X \models {r^*}^Y$.
- If $r = A \leftarrow B, not\, v, not\, C$ then $r^* = A \leftarrow B, not\, C \in \mathsf{f}_W(P,v)$. Suppose $I \not\models_{\mathsf{HT}} r$. Then we have two cases:
  - — $Y \not\models r$, which is an absurd since $v \notin Y$ and $I \models_{\mathsf{HT}} r^*$;
  - — $Y \cap C = \emptyset$ and $X \not\models A \leftarrow B$. This is an absurd since in this case ${r^*}^Y = A \leftarrow B$, and $X \models {r^*}^Y$.

We now prove the right to left inclusion. Let $I = \langle X,Y \rangle \in \mathcal{HT}(P \cup \{\leftarrow v\})$. Since $I \models_{\mathsf{HT}} \leftarrow v$ then clearly I does not contain v. We prove that for every $r \in \mathsf{f}_W(P,v)$ we have that $I \models_{\mathsf{HT}} r$. This immediately implies that $\mathcal{HT}(P \cup \{\leftarrow v\}) \subseteq \mathcal{HT}_{\|\{v\}}(\mathsf{f}(P,v))$. Let $r \in \mathsf{f}_W(P,v)$. We consider the following cases.

- $r \in Cn(P,v)$. In this case, since $I \in \mathcal{HT}(P \cup \{\leftarrow v\})$, we have that $I \in \mathcal{HT}(P)$, and therefore $I \models_{\mathsf{HT}} r$;
- If $r \notin Cn(P,v)$ then we have the following three possibilities:
  - — $r = A \leftarrow B, not\, C$ such that $r^* = A, v \leftarrow B, not\, v, not\, C \in Cn(P,v)$. Suppose $I \not\models_{\mathsf{HT}} r$. Then we have two cases:
    - – $Y \not\models r$, which is an absurd since $v \notin Y$ and $I \models_{\mathsf{HT}} r^*$;
    - – $Y \cap C = \emptyset$ and $X \not\models A \leftarrow B$. This is an absurd since $v \notin X$, ${r^*}^Y = A, v \leftarrow B$, and $X \models {r^*}^Y$.

— $r = A \leftarrow B, not\,C$ and $r^* = A, v \leftarrow B, not\,C \in Cn(P,v)$. Suppose $I \not\models_{\mathsf{HT}} r$. Then we have two cases:
  – $Y \not\models r$, which is an absurd since $v \notin Y$ and $I \models_{\mathsf{HT}} r^*$;
  – $Y \cap C = \emptyset$ and $X \not\models A \leftarrow B$. This is an absurd since $v \notin X$, $r^{*Y} = A, v \leftarrow B$, and $X \models r^{*Y}$.

— $r = A \leftarrow B, not\,C$ and $r^* = A \leftarrow B, not\,v, not\,C \in Cn(P,v)$. Suppose $I \not\models_{\mathsf{HT}} r$. Then we have two cases:
  – $Y \not\models r$, which is an absurd since $v \notin Y$ and $I \models_{\mathsf{HT}} r^*$;
  – $Y \cap C = \emptyset$ and $X \not\models A \leftarrow B$. This is an absurd since $r^{*Y} = A \leftarrow B$, and $X \models r^{*Y}$.

□

Lemma 2
Let P be a disjunctive program over a signature $\mathcal{A}$ and $V \subseteq \mathcal{A}$. Then, for every $\mathsf{f} \in \mathsf{F}_W$ we have

$$\mathcal{HT}_{\|V}(\mathsf{f}(P,V)) = \mathcal{HT}(P \cup \{\leftarrow v : v \in V\}).$$

Proof
The result follows easily by induction on the number of elements of V, using the definition of $\mathsf{f}_W(P,V)$ and Lemma 3. □

Lemma 3
Let P be a disjunctive program over a signature $\mathcal{A}$ and $V \subseteq \mathcal{A}$. Then, for every $\mathsf{f} \in \mathsf{F}_W$

$$\mathcal{AS}(\mathsf{f}(P,V)) = \{X \in \mathcal{AS}(P) : V \cap X = \emptyset\}.$$

Proof
Let $\mathsf{f} \in \mathsf{F}_W$. We will prove both inclusions.

Let us start with the left to right inclusion. Let $X \in \mathcal{AS}(\mathsf{f}(P,V))$. Then $\langle X,X\rangle \in \mathcal{HT}(\mathsf{f}(P,V))$ and there is no $X' \subset X$ such that $\langle X',X\rangle \in \mathcal{HT}(\mathsf{f}(P,V))$. We can conclude that $V \cap X = \emptyset$, since otherwise $\langle X \setminus V, X\rangle \in \mathcal{HT}(\mathsf{f}(P,V))$, which contradicts $X \in \mathcal{AS}(\mathsf{f}(P,V))$.

Since $\langle X,X\rangle \in \mathcal{HT}_{\|V}(\mathsf{f}(P,V))$, by Lemma 2, we have that $\langle X,X\rangle \in \mathcal{HT}(P \cup \{\leftarrow v : v \in V\})$, and therefore, $\langle X,X\rangle \in \mathcal{HT}(P)$. We need to prove that there is no $X' \subset X$ such that $\langle X',X\rangle \in \mathcal{HT}(P)$. Suppose there is. Then $\langle X',X\rangle \in \mathcal{HT}(P \cup \{\leftarrow v : v \in V\}) = \mathcal{HT}_{\|V}(\mathsf{f}(P,V))$, thus contradicting the fact that $X \in \mathcal{AS}(\mathsf{f}(P,V))$.

Now let us prove the right to left inclusion. Let $X \in \mathcal{AS}(P)$ such that $V \cap X = \emptyset$. Then $\langle X,X\rangle \in \mathcal{HT}(P)$ and there is no $X' \subset X$ such that $\langle X',X\rangle \in \mathcal{HT}(P)$. Since $V \cap X = \emptyset$ we have $\langle X',X\rangle \in \mathcal{HT}(P \cup \{\leftarrow v : v \in V\}) = \mathcal{HT}_{\|V}(\mathsf{f}(P,V))$. We just need to prove that there is no $X' \subset X$ such that $\langle X',X\rangle \in \mathcal{HT}(P \cup \{\leftarrow v : v \in V\})$. Suppose there is. Then, $\langle X',X\rangle \in \mathcal{HT}(P)$, which contradicts $X \in \mathcal{AS}(P)$. □

Lemma 4
Let $\mathcal{A}$ be a signature and $V \subseteq \mathcal{A}$. Let $I = \langle X,Y\rangle$ an HT-interpretation over $\mathcal{A}$ that does not contain any $v \in V$ and I^* such that $I^* \sim_V I$. Then, for every formula φ over $\mathcal{A}$ not containing any $v \in V$, we have that $I \models_{\mathsf{HT}} \varphi$ iff $I^* \models_{\mathsf{HT}} \varphi$.

Proof
We prove the result by induction on the structure of the formula φ. For the base case suppose φ is a propositional atom $p \in \mathcal{A}$, thus $p \notin V$. Then, $I \models_{\mathsf{HT}} p$ iff $p \in X$ iff $p \in X'$ with $X' \sim_V X$ (since $p \notin V$) iff $I^* \models_{\mathsf{HT}} p$.

For the induction step we only consider the case $\varphi = \varphi_1 \supset \varphi_2$. The other cases are straightforward. We have that $I \models_{\mathsf{HT}} \varphi$ iff (i) $Y \models_{\mathsf{HT}} \varphi_1 \supset \varphi_2$ and (ii) $I \models_{\mathsf{HT}} \varphi_2$ whenever $I \models_{\mathsf{HT}} \varphi_1$. From a well known result from classical logic, we have that $Y \models_{\mathsf{HT}} \varphi_1 \supset \varphi_2$ iff $Y \cup V \models_{\mathsf{HT}} \varphi_1 \supset \varphi_2$ (since no $v \in V$ occurs in φ_1 nor in φ_2). The equivalence of condition (ii) with the correspondent one for I^* follows easily using induction hypothesis. □

Lemma 5
Let $\mathcal{A}$ be a signature and $V \subseteq \mathcal{A}$. For every formula φ over $\mathcal{A}$ not containing any $v \in V$ we have that $\mathcal{HT}(\varphi) = (\mathcal{HT}_{\|V}(\varphi))_{\dagger V}$

Proof
Let us prove both inclusions. For the left to right inclusion, assume that $I \in \mathcal{HT}(\varphi)$. Then, $I_{\|V} \in \mathcal{HT}_{\|V}(\varphi)$. Therefore, $I \in (\mathcal{HT}_{\|V}(\varphi))_{\dagger V}$.

For the reverse inclusion, let $I \in (\mathcal{HT}_{\|V}(\varphi))_{\dagger V}$. Then, $I_{\|V} \in \mathcal{HT}_{\|V}(\varphi)$. Then, there exists $I^* \in \mathcal{HT}(\varphi)$ such that $I^* \sim_V I_{\|\{v\}}$. Using Lemma 4 we have that $I_{\|V} \in \mathcal{HT}(\varphi)$. Since $I \sim_V I_{\|\{v\}}$, we can conclude, again using Lemma 4, that $I \in \mathcal{HT}(\varphi)$. □

Lemma 6
Let $\mathcal{A}$ be a signature, $v \in \mathcal{A}$, and P_1, P_2 programs over $\mathcal{A}$, such that P_2 does not contain v. Then,

$$(\mathcal{HT}_{\|V}(P_1 \cup P_2))_{\dagger V} = (\mathcal{HT}_{\|V}(P_1) \cap \mathcal{HT}_{\|V}(P_2))_{\dagger V}$$

Proof
We prove both inclusions. Let us start with the left to right inclusion. Let $I \in (\mathcal{HT}_{\|V}(P_1 \cup P_2))_{\dagger V}$. Then, $I_{\|V} \in \mathcal{HT}_{\|V}(P_1 \cup P_2)$. Therefore, there exists $I' \sim_V I_{\|V}$ such that $I' \in \mathcal{HT}(P_1 \cup P_2)$. From this we can conclude that $I' \in \mathcal{HT}(P_1)$ and $I' \in \mathcal{HT}(P_2)$, and thus $I_{\|V} \in \mathcal{HT}_{\|V}(P_1)$ and $I_{\|V} \in (P_2)$. Therefore, we have that $I_{\|V} \in \mathcal{HT}_{\|V}(P_1) \cap \mathcal{HT}_{\|V}(P_2)$, thus $I \in (\mathcal{HT}_{\|V}(P_1) \cap \mathcal{HT}_{\|V}(P_2))_{\dagger V}$.

For the reverse inclusion consider that $I \in (\mathcal{HT}_{\|V}(P_1) \cap \mathcal{HT}_{\|V}(P_2))_{\dagger V}$. Then, $I_{\|V} \in \mathcal{HT}_{\|V}(P_1) \cap \mathcal{HT}_{\|V}(P_2)$, and thus $I_{\|V} \in \mathcal{HT}_{\|V}(P_1)$ and $I_{\|V} \in \mathcal{HT}_{\|V}(P_2)$. Therefore, there exists $I^* \sim_V I_{\|V}$ such that $I^* \in \mathcal{HT}(P_1)$. Since $\mathcal{HT}(P_2) = (\mathcal{HT}_{\|V}(P_2))_{\dagger V}$ (by Lemma 5 and since no $v \in V$ occurs in P_2), $I_{\|V} \in \mathcal{HT}_{\|V}(P_2)$ and since $I^* \sim_V I_{\|V}$, we can conclude that $I^* \in \mathcal{HT}(P_2)$. Therefore, $I^* \in \mathcal{HT}(P_1 \cup P_2)$, and then $I_{\|V} \in \mathcal{HT}_{\|V}(P_1 \cup P_2)$. We can then conclude that $I \in (\mathcal{HT}_{\|V}(P_1 \cup P_2))_{\dagger V}$.
□

Lemma 7
Let $\mathcal{A}$ be a signature and $v \in \mathcal{A}$. Let B_1, B_2, C_1, C_2, $\{a\}$ sets of atoms over $\mathcal{A} \setminus \{v\}$. Consider the rules

$$r_1 = a \leftarrow B_1, v, not\, C_1$$
$$r_2 = v \leftarrow B_2, not\, C_2$$
$$r = a \leftarrow B_1, B_2, not\, C_1, not\, C_2$$

Then,

$$\mathcal{HT}(\{r_1\} \cup \{r_2\}) \subseteq \mathcal{HT}(\{r\})$$

Proof
Let $I = \langle X, Y \rangle \in \mathcal{HT}(\{r_1\} \cup \{r_2\})$. Then, we have

$i_1)$ $Y \models a \leftarrow B_1, v, not\, C_1$,
$ii_1)$ $X \models a \leftarrow B_1, v$ whenever $C_1 \cap Y = \emptyset$,
$i_2)$ $Y \models v \leftarrow B_2, not\, C_2$,
$ii_2)$ $X \models v \leftarrow B_2$ whenever $C_2 \cap Y = \emptyset$.

Our aim is to prove that $I \in \mathcal{HT}(\{r\})$, i.e.,

$i)$ $Y \models a \leftarrow B_1, B_2, not\, C_1, not\, C_2$ and
$ii)$ $X \models a \leftarrow B_1, B_2$ whenever $(C_1 \cup C_2) \cap Y = \emptyset$.

If $(C_1 \cup C_2) \cap Y \neq \emptyset$ then $i)$ and $ii)$ trivially hold. So, suppose that $(C_1 \cup C_2) \cap Y = \emptyset$. If $B_1 \cup B_2 \not\subseteq Y$, then again $i)$ and $ii)$ trivially hold. So assume that $B_1 \cup B_2 \subseteq Y$. Then from $i_1)$ and $i_2)$ we can conclude that $Y \models a$, and so $i)$ holds. If $B_1 \cup B_2 \not\subseteq X$ then immediately $ii)$ holds. So assume that $B_1 \cup B_2 \subseteq X$. Then $ii_1)$ and $ii_2)$ allow us to conclude that $X \models a$, and so $ii)$ holds. □

Lemma 8
Let $\mathcal{A}$ be a signature, $V \subseteq \mathcal{A}$, and P_1, P_2 programs over $\mathcal{A}$. Then, $(\mathcal{HT}_{\|V}(P_1))_{\dagger V} \cap (\mathcal{HT}_{\|V}(P_2))_{\dagger V}$ $= (\mathcal{HT}_{\|V}(P_1) \cap \mathcal{HT}_{\|V}(P_2))_{\dagger V}$.

Proof
Let I be an HT-interpretation over $\mathcal{A}$. Consider the following sequence of equivalent sentences:

$I \in (\mathcal{HT}_{\|V}(P_1))_{\dagger V} \cap (\mathcal{HT}_{\|V}(P_2))_{\dagger V}$ iff
$I \in (\mathcal{HT}_{\|V}(P_i))_{\dagger V}$, for each $i \in \{1,2\}$ iff
$I_{\|V} \in \mathcal{HT}_{\|V}(P_i)$ for each $i \in \{1,2\}$ iff
$I_{\|V} \in (\mathcal{HT}_{\|V}(P_1) \cap \mathcal{HT}_{\|V}(P_2))$ iff
$I \in (\mathcal{HT}_{\|V}(P_1) \cap \mathcal{HT}_{\|V}(P_2))_{\dagger V}$.

□

References

DELGRANDE, J. P. AND WANG, K. 2015. A syntax-independent approach to forgetting in disjuctive logic programs. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*, B. Bonet and S. Koenig, Eds. AAAI Press, 1482–1488.

EITER, T., FINK, M., TOMPITS, H., AND WOLTRAN, S. 2004. On eliminating disjunctions in stable logic programming. In *Principles of Knowledge Representation and Reasoning: Proceedings of the Ninth International Conference (KR2004)*, D. Dubois, C. A. Welty, and M. Williams, Eds. AAAI Press, 447–458.

EITER, T. AND WANG, K. 2008. Semantic forgetting in Answer Set Programming. *Artif. Intell. 172,* 14, 1644–1672.

GONÇALVES, R., JANHUNEN, T., KNORR, M., AND LEITE, J. 2021. On syntactic forgetting under uniform equivalence. In *Logics in Artificial Intelligence - 17th European Conference, JELIA 2021, Proceedings*, W. Faber, G. Friedrich, M. Gebser, and M. Morak, Eds. Lecture Notes in Computer Science, vol. 12678. Springer, 297–312.

GONÇALVES, R., JANHUNEN, T., KNORR, M., LEITE, J., AND WOLTRAN, S. 2019. Forgetting in Modular Answer Set Programming. In *The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019, The Thirty-First Innovative Applications of Artificial Intelligence Conference, IAAI 2019, The Ninth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2019*. AAAI Press, 2843–2850.

GONÇALVES, R., KNORR, M., AO LEITE, J., AND WOLTRAN, S. 2020. On the limits of forgetting in Answer Set Programming. *Artif. Intell. 286*, 103–307.

GONÇALVES, R., KNORR, M., AND LEITE, J. 2017. Iterative variable elimination in ASP. In *Progress in Artificial Intelligence - 18th EPIA Conference on Artificial Intelligence, EPIA 2017, Proceedings*, E. C. Oliveira, J. Gama, Z. A. Vale, and H. L. Cardoso, Eds. Lecture Notes in Computer Science, vol. 10423. Springer, 643–656.

KNORR, M. AND ALFERES, J. J. 2014. Preserving strong equivalence while forgetting. In *Logics in Artificial Intelligence - 14th European Conference, JELIA 2014. Proceedings*, E. Fermé and J. Leite, Eds. Lecture Notes in Computer Science, vol. 8761. Springer, 412–425.

WANG, Y., WANG, K., AND ZHANG, M. 2013. Forgetting for answer set programs revisited. In *IJCAI 2013, Proceedings of the 23rd International Joint Conference on Artificial Intelligence*, F. Rossi, Ed. IJCAI/AAAI, 1162–1168.

WANG, Y., ZHANG, Y., ZHOU, Y., AND ZHANG, M. 2012. Forgetting in logic programs under strong equivalence. In *Principles of Knowledge Representation and Reasoning: Proceedings of the Thirteenth International Conference, KR 2012*, G. Brewka, T. Eiter, and S. A. McIlraith, Eds. AAAI Press.

WANG, Y., ZHANG, Y., ZHOU, Y., AND ZHANG, M. 2014. Knowledge forgetting in Answer Set Programming. *J. Artif. Intell. Res. 50*, 31–70.

WONG, K.-S. 2008. Sound and complete inference rules for SE-consequence. *J. Artif. Intell. Res. 31*, 205–216.

WONG, K.-S. 2009. Forgetting in logic programs. Ph.D. thesis, The University of New South Wales.

ZHANG, Y. AND FOO, N. Y. 2006. Solving logic program conflict through strong and weak forgettings. *Artif. Intell. 170,* 8-9, 739–778.