Artificial Intelligence 275 (2019) 575–601

Contents lists available at ScienceDirect

Artificial Intelligence

www.elsevier.com/locate/artint

Forgetting auxiliary atoms in forks

Felicidad Aguado [a], Pedro Cabalar [a,*], Jorge Fandinno [a,b,c], David Pearce [d], Gilberto Pérez [a], Concepción Vidal [a]

[a] CITIC Research Center, University of A Coruña, Spain
[b] Institut de Recherche en Informatique de Toulouse, Toulouse University, France
[c] Universität Potsdam, Germany
[d] Universidad Politécnica de Madrid, Spain

ARTICLE INFO

Article history:
Received 7 June 2018
Received in revised form 12 July 2019
Accepted 16 July 2019
Available online 19 July 2019

Keywords:
Answer set programming
Non-monotonic reasoning
Equilibrium logic
Denotational semantics
Forgetting
Strong equivalence

ABSTRACT

In this work we tackle the problem of checking strong equivalence of logic programs that may contain local auxiliary atoms, to be removed from their stable models and to be forbidden in any external context. We call this property *projective strong equivalence* (PSE). It has been recently proved that not any logic program containing auxiliary atoms can be reformulated, under PSE, as another logic program or formula without them – this is known as *strongly persistent forgetting*. In this paper, we introduce a conservative extension of *Equilibrium Logic* and its monotonic basis, the logic of *Here-and-There*, in which we deal with a new connective '|' we call *fork*. We provide a semantic characterisation of PSE for forks and use it to show that, in this extension, it is always possible to forget auxiliary atoms under strong persistence. We further define when the obtained fork is representable as a regular formula.

1. Introduction

Answer Set Programming (ASP [1]) has become an established problem-solving paradigm for Knowledge Representation and Reasoning (KRR). The reasons for this success derive from the practical point of view, with the availability of efficient solvers [2,3] and application domains [4], but also from its solid theoretical foundations, rooted in the *stable models* [5] semantics for normal logic programs that was later generalised to arbitrary propositional [6], first-order [7,8] and infinitary [9] formulas. An important breakthrough that supported these extensions of ASP has been its logical characterisation in terms of *Equilibrium Logic* [6] and its monotonic basis, the intermediate logic of *Here-and-There* (HT). Despite its expressiveness, a recent result [10] has shown that Equilibrium Logic has limitations in capturing the representational power of auxiliary atoms, which cannot always be forgotten. To illustrate this point, take the following problem.

Example 1. Two individuals, mother and father, both carrying alleles a and b, procreate an offspring. We want to generate all the possible ways in which the offspring may inherit its parents' genetic information.

* Corresponding author.
E-mail addresses: aguado@udc.es (F. Aguado), cabalar@udc.es (P. Cabalar), jorge.fandino@udc.es, jorge.fandinno@irit.fr, fandinno@uni-potsdam.de (J. Fandinno), david.pearce@upm.es (D. Pearce), gilberto.pvega@udc.es (G. Pérez), concepcion.vidalm@udc.es (C. Vidal).

https://doi.org/10.1016/j.artint.2019.07.005
0004-3702/

According to Mendelian laws, we should obtain three possible combinations that, ignoring their frequency, correspond to the sets of alleles $\{a\}$, $\{b\}$ and $\{a, b\}$. These are, in fact, the three classical models of disjunction $a \vee b$. To obtain these three solutions as stable models in ASP, the straightforward way would be to use the three rules:

$$a \vee \neg a \qquad b \vee \neg b \qquad \bot \leftarrow \neg a \wedge \neg b \tag{P_1}$$

We assume here some familiarity with ASP: disjunctions of the form $p \vee \neg p$ act as non-deterministic *choice rules* (allowing the arbitrary inclusion of atom p) and $\bot \leftarrow \neg a \wedge \neg b$ is a *constraint* forbidding models where $a \vee b$ does not hold. Moreover, when we include $p \vee \neg p$ for all atoms, as in the example, stable models just coincide with classical models. A drawback of this representation is that it does not differentiate the information coming from each parent, possibly becoming a problem of elaboration tolerance. For instance, if only the mother's information were available, one would expect to obtain the stable models $\{a\}$ and $\{b\}$ but not $\{a, b\}$, as there is no evidence of that combination without further information about the father. So, the mother alone would be better represented by a regular disjunction $a \vee b$. However, we cannot represent each parent as an independent disjunction like that, since $(a \vee b) \wedge (a \vee b)$ just amounts to $(a \vee b)$ and the combination $\{a, b\}$ is not obtained. A simple way to represent these two disjunctions separately is using auxiliary atoms to keep track of alleles from the mother $(ma \vee mb)$ and the father $(fa \vee fb)$. This leads to program P_2:

$$ma \vee mb \qquad a \leftarrow ma \qquad b \leftarrow mb \tag{P_m}$$

$$fa \vee fb \qquad a \leftarrow fa \qquad b \leftarrow fb \tag{P_f}$$

consisting of the mother's contribution P_m and the father's contribution P_f. Four stable models are obtained from P_2, $\{ma, fa, a\}$, $\{mb, fb, b\}$, $\{ma, fb, a, b\}$ and $\{mb, fa, a, b\}$, but if we *project* them on the original vocabulary $V = \{a, b\}$ (i.e. we remove auxiliary atoms), they collapse to three $\{a\}$, $\{b\}$ and $\{a, b\}$ as expected. Note that, although auxiliary atoms in this example have a meaning in the real world (they represent the effective sources of each inherited allele) they were not part of the original alphabet $V = \{a, b\}$ of Example 1, which does not distinguish between the same effect $\{a, b\}$ but due to different sources $\{ma, fb, a, b\}$ and $\{mb, fa, a, b\}$.

As we have seen, P_1 and P_2 are "V-equivalent" in the sense that they yield the same stable models when projected to alphabet $V = \{a, b\}$. A natural question is whether this also holds in any context, that is, if $P_1 \cup Q$ and $P_2 \cup Q$ also yield the same V-projected stable models, for any context Q in the target alphabet V (since we want to keep auxiliary atoms local or hidden). This is obviously a kind of *strong equivalence* relation [11] – in fact it is one of the possible generalisations[1] of strong equivalence studied in [12]. In this paper, we will just call it *projective strong equivalence* (PSE) with respect to V, or V*-strong equivalence* for short. The PSE relation has also been used in the literature for comparing a program P and some transformation $tr(P)$ that either extends the vocabulary with new auxiliary atoms [13] (called there *strong faithfulness*) or reduces it for forgetting atoms as in [10] (called there *strong persistence*).

As we will see later, programs P_1 and P_2 are indeed V-strongly equivalent, so they express the same combined knowledge obtained from both parents. However, if we want to keep program P_m alone capturing the mother's contribution, *there is no possible* $\{a, b\}$*-strongly equivalent representation* in Equilibrium Logic (the same happens with P_f). In other words, we cannot forget atoms ma and mb in P_m and get a program preserving PSE. This impossibility follows from a recent result in [10] that shows that forgetting atoms under strong persistence is sometimes impossible. In practice, this means that auxiliary atoms in ASP are more than 'just' auxiliary, as they allow one to represent problems that cannot be captured without them. A natural idea is to consider an extension of ASP in which forgetting auxiliary atoms is always possible.

In this paper, we extend logic programs to include a new construct '$|$' we call *fork* and whose intuitive meaning is that the stable models of $P \mid P'$ correspond to the union of stable models from P and P' in any context[2] Q, that is $SM[(P \mid P') \wedge Q] = SM[P \wedge Q] \cup SM[P' \wedge Q]$. Using this construct, we can represent Example 1 as the conjunction of two forks $(a \mid b) \wedge (a \mid b)$, one per each parent. This conjunction of forks is not idempotent but will actually amount to $(a \mid b \mid (a \wedge b))$. We will show that forgetting is always possible in forks but some of them, such as $(a \mid b)$, cannot be represented in Equilibrium Logic.

The rest of the paper is organised as follows. The next section recalls basic definitions of HT and Equilibrium Logic. Then, we introduce an alternative characterisation of HT in terms of T*-supports*. In the next section, we extend the syntax with the fork connective and generalise the semantics to sets of T-supports (so-called T*-views*). After that, we characterise PSE for forks and relate this property to forgetting. Finally, we discuss related work and conclude the paper. Proofs of results are collected in Appendix A.

2. Preliminaries

We begin by recalling some basic definitions and results related to HT. Let At be a finite set of atoms called the *(propositional) signature*. A *(propositional) formula* φ is defined using the grammar:

[1] It corresponds to *relativised* strong equivalence (with respect to V) with projection (with respect to V).

[2] For simplicity, we understand programs as the conjunction of their rules.

$$\varphi ::= \bot \mid p \mid \varphi \wedge \varphi \mid \varphi \vee \varphi \mid \varphi \rightarrow \varphi$$

where p is an atom $p \in At$. We will use Greek letters φ, ψ, γ and their variants to stand for formulas. We define the derived operators $\neg\varphi \stackrel{\text{def}}{=} (\varphi \rightarrow \bot)$, $\top \stackrel{\text{def}}{=} \neg\bot$ and $\varphi \leftrightarrow \psi \stackrel{\text{def}}{=} (\varphi \rightarrow \psi) \wedge (\psi \rightarrow \varphi)$. Given a formula φ, by $At(\varphi) \subseteq At$ we denote the set of atoms occurring in φ. A *literal* is an atom p or its negation $\neg p$. A *program* is a set of implications of the form $\alpha \rightarrow \beta$ where α is a conjunction of literals and β a disjunction of literals. A *theory* is a set of formulas. For simplicity, we consider finite theories understood as the conjunction of their formulas. The extension to infinite theories is straightforward.

A *classical interpretation* T is a set of atoms $T \subseteq At$. We write $T \models \varphi$ to stand for the usual classical satisfaction of a formula φ. An HT-*interpretation* is a pair $\langle H, T\rangle$ (respectively called "here" and "there") of sets of atoms $H \subseteq T \subseteq At$; it is said to be *total* when $H = T$. The fact that an interpretation $\langle H, T\rangle$ *satisfies* a formula φ, written $\langle H, T\rangle \models \varphi$, is recursively defined as follows:

- $\langle H, T\rangle \not\models \bot$
- $\langle H, T\rangle \models p$ iff $p \in H$
- $\langle H, T\rangle \models \varphi \wedge \psi$ iff $\langle H, T\rangle \models \varphi$ and $\langle H, T\rangle \models \psi$
- $\langle H, T\rangle \models \varphi \vee \psi$ iff $\langle H, T\rangle \models \varphi$ or $\langle H, T\rangle \models \psi$
- $\langle H, T\rangle \models \varphi \rightarrow \psi$ iff both (i) $T \models \varphi \rightarrow \psi$ and (ii) $\langle H, T\rangle \not\models \varphi$ or $\langle H, T\rangle \models \psi$

By abuse of notation, we use '$\models$' both for classical and for HT-satisfaction: the ambiguity is removed by the form of the left interpretation (a single set T for classical and a pair $\langle H, T\rangle$ for HT). It is not difficult to see that, for total interpretations, $\langle T, T\rangle \models \varphi$ amounts to classical satisfaction $T \models \varphi$. A formula φ is an HT-*tautology* (or is HT-*valid*) iff it is satisfied by any HT-interpretation. HT is strictly weaker than classical logic. For instance, the *excluded middle* $\varphi \vee \neg\varphi$ is not valid in HT, but the *weak excluded middle*:

$$\neg\varphi \vee \neg\neg\varphi \tag{1}$$

is an HT-tautology. On the other hand, since HT is an intermediate logic, any intuitionistic tautology is also an HT tautology.

An HT-interpretation $\langle H, T\rangle$ is a *model* of a theory Γ if $\langle H, T\rangle \models \varphi$ for all $\varphi \in \Gamma$; otherwise, $\langle H, T\rangle$ is a *countermodel* of Γ. Two formulas φ and ψ are HT-*equivalent* if they have the same models: this is the same as requiring that $\varphi \leftrightarrow \psi$ is a tautology.

Proposition 1 *(Persistence). For any HT-interpretation $\langle H, T\rangle$ and any formula φ:*
$\langle H, T\rangle \models \varphi$ implies $\langle T, T\rangle \models \varphi$ (i.e. $T \models \varphi$ classically).

Rather than obtaining the models of a formula, we will sometimes be interested in the converse problem, namely, obtaining a formula from a set of HT-models – this problem was studied in [14]. Given an HT-interpretation $\langle H, T\rangle$ we define the formula (or *clause*) $C_{\langle H,T\rangle}$ as:

$$\Big(\bigwedge_{a \in H} a\Big) \wedge \Big(\bigwedge_{b \in At \setminus T} \neg b\Big) \wedge \Big(\bigwedge_{c \in T \setminus H} \neg\neg c\Big) \wedge \Big(\bigwedge_{d,e \in T \setminus H,\ d \neq e} (d \rightarrow e)\Big)$$

and, for any set of HT-interpretations S, we define the disjunction of all such formulas as:

$$\Phi_S \stackrel{\text{def}}{=} \bigvee_{\langle H,T\rangle \in S} C_{\langle H,T\rangle}$$

Example 2. For signature $At = \{a, b\}$, take the set of HT-interpretations:

$$S = \{\ \langle\{p\},\{p\}\rangle,\ \langle\{q\},\{q\}\rangle,\ \langle\{p\},\{p,q\}\rangle,\ \langle\{q\},\{p,q\}\rangle,\ \langle\{p,q\},\{p,q\}\rangle\}.$$

for signature $At = \{p, q\}$. Formula Φ_S is the disjunction of clauses:

$(H, T) \in S$	$C_{\langle H,T\rangle}$	
$\langle\{p\},\{p\}\rangle$	$p \wedge \neg q$	(2)
$\langle\{q\},\{q\}\rangle$	$q \wedge \neg p$	(3)
$\langle\{p\},\{p,q\}\rangle$	$p \wedge \neg\neg q$	(4)
$\langle\{q\},\{p,q\}\rangle$	$q \wedge \neg\neg p$	(5)
$\langle\{p,q\},\{p,q\}\rangle$	$p \wedge q$	(6)

In general, Φ_S has as many clauses as interpretations in S but, normally, the formula can be simplified.[3] In this case, if we apply distributivity on (2) $\vee$ (4) we get $p \wedge (\neg q \vee \neg\neg q)$. The last conjunct is an instance of weak excluded middle (1) and can removed leaving p. Using the same reasoning (3) $\vee$ (5) $\leftrightarrow q$. Finally, this leaves the disjunction $p \vee q \vee (p \wedge q)$ that is intuitionistically equivalent to $p \vee q$. In fact, it is easy to check that our starting set S precisely collects the HT-models of $p \vee q$. $\square$

It should be noticed, however, that not any arbitrary set S of interpretations may correspond to a set of models, since persistence (Proposition 1) must be preserved. We say that a set S of HT-interpretations is *total-closed* if $\langle H, T\rangle \in S$ implies $\langle T, T\rangle \in S$. It is easy to see that S in Example 2 is total-closed.

Theorem 1 *(Theorem 2 in [14]).* Each total-closed set S of interpretations is the set of HT-models of the disjunction of clauses Φ_S.

An alternative way to obtain the HT-models of a formula φ is using Ferraris' reduct [16], φ^T, defined as the result of replacing by $\bot$ those maximal subformulas of φ that are not (classically) satisfied by interpretation T. As an example, given $\varphi = (\neg a \to b)$ we have the reducts $\varphi^{\emptyset} = \bot$, $\varphi^{\{a\}} = (\bot \to \bot)$, $\varphi^{\{b\}} = (\neg\bot \to b)$ and $\varphi^{\{a,b\}} = (\bot \to b)$. The correspondence with HT-satisfaction is given by:

Proposition 2 *(Lemma 1, [16]). Given $H \subseteq T$: $\langle H, T\rangle \models \varphi$ iff $H \models \varphi^T$.*

A total interpretation $\langle T, T\rangle$ is an *equilibrium model* of a formula φ iff $\langle T, T\rangle \models \varphi$ and there is no $H \subset T$ such that $\langle H, T\rangle \models \varphi$. If so, we say that T is a *stable model* of φ. By Proposition 2, this means that T is a stable model of φ iff it is a minimal classical model of φ^T. We write $SM[\varphi]$ to stand for the set of stable models of φ. Moreover, we represent their projection onto some vocabulary V as $SM_V[\varphi] \stackrel{\text{def}}{=} \{\, T \cap V \mid T \in SM[\varphi]\,\}$.

Definition 1 *(Projective strong entailment/equivalence).* Let φ and ψ be formulas and $V \subseteq At$ some vocabulary (set of atoms). We say that φ *V-strongly entails* ψ, written $\varphi \mid\!\sim_V \psi$ if $SM_V[\varphi \wedge \gamma] \subseteq SM_V[\psi \wedge \gamma]$ for any formula γ such that $At(\gamma) \subseteq V$. We further say that φ and ψ are *V-strongly equivalent*, written $\varphi \cong_V \psi$, if both $\varphi \mid\!\sim_V \psi$ and $\psi \mid\!\sim_V \varphi$, that is, $SM_V[\varphi \wedge \gamma] = SM_V[\psi \wedge \gamma]$ for any formula γ such that $At(\gamma) \subseteq V$.

When the vocabulary $V \supseteq At(\varphi) \cup At(\psi)$ contains the original language of φ and ψ, the projection has no relevant effect and the previous definitions amount to regular (non-projective) *strong entailment* and *strong equivalence*. In this case, we simply drop the V subindex in the previous notations. The following results, respectively proved in [11] and [17], characterise non-projective strong equivalence and entailment in terms of HT:

Proposition 3 *(From [11] and [17]). Let φ, ψ be a pair of formulas. Then*

(i) *$\varphi \cong \psi$ iff φ and ψ are HT-equivalent,*
(ii) *$\varphi \mid\!\sim \psi$ iff both φ classically entails ψ and, for any H, if $\langle H, T\rangle \models \psi$ and $T \models \varphi$ then $\langle H, T\rangle \models \varphi$.*

In the case of projected strong entailment and equivalence, a semantic characterisation was provided in [12], although limited to the case of disjunctive logic programs. We will provide later a characterisation of strong entailment and equivalence for fork formulas that, for the particular case in which the fork operator does not occur, will also constitute an extension of [12] to arbitrary propositional formulas.

3. *T*-supports

As we saw before, for deciding whether some total interpretation $\langle T, T\rangle$ is an equilibrium model of a formula or not, we check its H-minimality among models of the form $\langle H, T\rangle$. It makes sense, therefore, to organise the HT-models grouping those H components that correspond to each fixed T. This idea was already explored in [18] and is extended in this section so it can actually be used as a complete alternative characterisation of HT semantics.

Definition 2 *(T-support).* Given a set T of atoms, a *T-support* $\mathcal{H}$ is a set of subsets of T, that is $\mathcal{H} \subseteq 2^T$, satisfying $T \in \mathcal{H}$ if $\mathcal{H} \neq \emptyset$. We write $\mathbf{H}_T$ to stand for the set of all possible T-supports.

[3] A systematic simplification method is explained in [15]: that method starts from countermodels instead, and obtains a minimal-size logic program as a result.

To increase readability of examples, we will just write a support as a sequence of interpretations between square brackets. For instance, possible supports for $T = \{a, b\}$ are $[\{a, b\}\ \{a\}]$, $[\{a, b\}\ \{b\}\ \emptyset]$ or the empty support $[\ ]$.

Intuitively, $\mathcal{H}$ will be used to capture the set of "here" components H that *support* the "there" world T as a model of a given formula φ, that is, the set of H's such that $\langle H, T\rangle \models \varphi$. When $\mathcal{H}$ is empty $[\ ]$, there is no support for T, so $\langle T, T\rangle \not\models \varphi$ and thus, T is not even a classical model. If $\mathcal{H}$ is not empty, this means we have at least some model $\langle H, T\rangle$ and, by Proposition 1, $\langle T, T\rangle$ must be a model too; this is why we require $T \in \mathcal{H}$ in the set. When not empty, the fewer models in $\mathcal{H}$, the more supported is T, since it is closer to being stable. Seeing "more supported" as an ordering relation, the "most supported" $\mathcal{H}$ (the top element) would precisely be $\mathcal{H} = [\ T\ ]$ corresponding to a stable model. This ordering relation is formally defined below.

Definition 3. Given a set $T \subseteq At$ of atoms and two T-supports $\mathcal{H}$ and $\mathcal{H}'$ we write $\mathcal{H} \preceq_T \mathcal{H}'$ iff either $\mathcal{H} = [\ ]$ or $\mathcal{H} \supseteq \mathcal{H}' \neq [\ ]$.

Proposition 4. *The relation $\preceq_T$ is a partial order on $\mathbf{H}_T$ with $[\ ]$ and $[\ T\ ]$ its bottom and top elements, respectively.* □

We usually write $\mathcal{H} \preceq \mathcal{H}'$ instead of $\mathcal{H} \preceq_T \mathcal{H}'$ when clear from the context. As an example, the classical interpretation $T = \{a, b\}$ is more supported in $\mathcal{H}_1 = [\{a, b\}\ \{a\}]$ than in $\mathcal{H}_2 = [\{a, b\}\ \{a\}\ \{b\}\ \emptyset]$, that is $\mathcal{H}_2 \preceq \mathcal{H}_1$, because $\mathcal{H}_2$ contains additional interpretations and is further from being stable.The next result collects several useful properties about T-supports and their ordering relation $\preceq$.

Proposition 5. *Given T-supports $\mathcal{H}$, $\mathcal{H}'$ and $\mathcal{H}''$ we have:*

(i) *$\mathcal{H} \cap \mathcal{H}' = [\ ]$ iff $\mathcal{H} = [\ ]$ or $\mathcal{H}' = [\ ]$,*
(ii) *if $\mathcal{H}' \preceq \mathcal{H}''$ then $\mathcal{H} \cap \mathcal{H}' \preceq \mathcal{H} \cap \mathcal{H}''$,*
(iii) *if $[\ ] \neq \mathcal{H}' \preceq \mathcal{H}''$ then $\mathcal{H} \cup \mathcal{H}' \preceq \mathcal{H} \cup \mathcal{H}''$,*
(iv) *if $\mathcal{H} \preceq \mathcal{H}' \cup \mathcal{H}''$ then $\mathcal{H} \preceq \mathcal{H}'$ or $\mathcal{H} \preceq \mathcal{H}''$.* □

Given a T-support $\mathcal{H}$, we define its complementary support $\overline{\mathcal{H}}$ as:

$$\overline{\mathcal{H}} \stackrel{\text{def}}{=} \begin{cases} [\ ] & \text{if } \mathcal{H} = 2^T \\ [\ T\ ] \cup \{H \subseteq T \mid H \notin \mathcal{H}\} & \text{otherwise} \end{cases}$$

We also define $\mathcal{H}_V$ as the projection of every set in $\mathcal{H}$ to the vocabulary V, i.e., $\mathcal{H}_V \stackrel{\text{def}}{=} \{H \cap V \mid H \in \mathcal{H}\}$. The relation between T-supports and formulas is given by the following definition.

Definition 4 *(T-denotation).* Let $T \subseteq At$. The *T-denotation* of a formula φ, written $[\![\, \varphi \,]\!]^T$, is a T-support recursively defined as follows:

$$[\![\, \bot \,]\!]^T \stackrel{\text{def}}{=} [\ ] \qquad\qquad [\![\, \varphi \wedge \psi \,]\!]^T \stackrel{\text{def}}{=} [\![\, \varphi \,]\!]^T \cap [\![\, \psi \,]\!]^T$$

$$[\![\, p \,]\!]^T \stackrel{\text{def}}{=} \{\, H \subseteq T \mid p \in H \,\} \qquad\qquad [\![\, \varphi \vee \psi \,]\!]^T \stackrel{\text{def}}{=} [\![\, \varphi \,]\!]^T \cup [\![\, \psi \,]\!]^T$$

$$[\![\, \varphi \to \psi \,]\!]^T \stackrel{\text{def}}{=} \begin{cases} [\ ] & \text{if } [\![\, \varphi \,]\!]^T \neq [\ ] \text{ and } [\![\, \psi \,]\!]^T = [\ ] \\ \overline{[\![\, \varphi \,]\!]^T} \cup [\![\, \psi \,]\!]^T & \text{otherwise} \end{cases}$$

The following proposition follows by structural induction and shows that T-denotations can be used as an alternative semantics for the logic HT.

Proposition 6. *For any interpretation $\langle H, T\rangle$ and formula φ: $\langle H, T\rangle \models \varphi$ iff $H \in [\![\, \varphi \,]\!]^T$.*

Corollary 1. *For any set T of atoms and propositional formulas φ, ψ, the following conditions hold:*

(i) *$T \models \varphi$ iff $[\![\, \varphi \,]\!]^T \neq [\ ]$ iff $T \in [\![\, \varphi \,]\!]^T$.*
(ii) *T is a stable model of φ iff $[\![\, \varphi \,]\!]^T = [\ T\ ]$.*
(iii) *Given $H \subseteq T$: $H \in [\![\, \varphi \,]\!]^T$ iff $H \models \varphi^T$.*

The last item asserts that the denotation $[\![\, \varphi \,]\!]^T$ can also be seen as a semantic counterpart of Ferraris' reduct φ^T.

Suppose that, for each T, we are given some arbitrary support $\sigma(T)$. The following result asserts that any such arbitrary assignment of supports σ corresponds to (the denotation of) some formula.

T	$[\![(7)]\!]^T$	$[\![(8)]\!]^T$
$\emptyset$	$[\,]$	$[\,]$
$\{p\}$	$[\,\{p\}\,]$	$[\,\{p\}\,]$
$\{q\}$	$[\,\{q\}\,]$	$[\,\{q\}\,]$
$\{p,q\}$	$[\{p,q\}\ \{p\}\ \{q\}\ \emptyset\,]$	$[\,\{p,q\}\ \{p\}\ \{q\}\,]$

Fig. 1. T-denotations for (7) and (8).

Proposition 7. *Let σ be a function that assigns an arbitrary support $\sigma(T) \in \mathbf{H}_T$ for each $T \subseteq At$. Then, $[\![\Phi_S]\!]^T = \sigma(T)$ for all T where $S \stackrel{\text{def}}{=} \{\langle H, T\rangle \mid H \in \sigma(T)\}$.*

Proof. We observe first that S is total-closed: if $\langle H, T\rangle \in S$ then $H \in \sigma(T)$ and, by construction of support, $T \in \sigma(T)$, and so $\langle T, T\rangle \in S$. Then,

$$\begin{array}{llll} H \in [\![\Phi_S]\!]^T & \Leftrightarrow & \langle H, T\rangle \models \Phi_S & \text{by Proposition 6} \\ & \Leftrightarrow & \langle H, T\rangle \in S & \text{by Theorem 1 and } S \text{ total-closed} \\ & \Leftrightarrow & H \in \sigma(T) & \text{by definition of } S \quad \Box \end{array}$$

Example 3 *(Implementing a choice).* To implement a choice rule for atom p (in modern ASP syntax, written $0\ \{p\}\ 1$ or just $\{p\}$) a knowledge engineer uses an auxiliary atom q. As a first option, she considers the use of rules:

$$(\neg p \to q) \wedge (\neg q \to p) \tag{7}$$

However, having a disjunctive ASP solver, another possibility could be:

$$p \vee q \tag{8}$$

Is there any substantial difference? $\Box$

The T-denotations of both options are shown in Fig. 1. According to Corollary 1, stable models correspond to T's such that $[\![\varphi]\!]^T = [\ T\ ]$, that is, we get the two stable models $\{p\}$ and $\{q\}$ in both columns. In fact, all rows coincide except for $T = \{p,q\}$ where $\emptyset \in [\![(7)]\!]^{\{p,q\}}$ but $\emptyset \notin [\![(8)]\!]^{\{p,q\}}$. Proposition 6 tells us that a difference in T-denotations means that the formulas are not HT equivalent, and so, they are not strongly equivalent [11]. In fact, the counterexample of strong equivalence is well-known in the literature: adding $(p \to q) \wedge (q \to p)$ to (7) yields no stable model, while the same addition to (8) produces stable model $\{p,q\}$.

Even though denotations do not coincide, we can still observe that $[\![(7)]\!]^T \preceq [\![(8)]\!]^T$ holds in all rows, that is, the first formula is always less supported (further from being stable) than the second one. In fact, this has an interesting consequence, as stated by the next result:

Proposition 8. *For any two propositional formulas φ, ψ the following hold:*

(i) *$\varphi \mathrel{|\!\!\sim} \psi$ iff $[\![\varphi]\!]^T \preceq [\![\psi]\!]^T$ for every set $T \subseteq At$ of atoms,*
(ii) *$\varphi \cong \psi$ iff $[\![\varphi]\!]^T = [\![\psi]\!]^T$ for every set $T \subseteq At$ of atoms.*

While (ii) is an immediate consequence of Proposition 6, item (i) states that φ strongly entails ψ iff the former is always less supported than the latter. Note how Proposition 8 is much more readable than Proposition 3, especially regarding strong entailment and its relation to strong equivalence.

In our example, Proposition 8 implies that if we replace (7) by (8) in any program, we will get the same or perhaps more stable models, i.e., $(7) \mathrel{|\!\!\sim} (8)$. Moreover, it is not difficult to check[4] that this happens with $\neg p \to q$ and $\neg p \to q$ individually, that is, both $(\neg p \to q) \mathrel{|\!\!\sim} (8)$ and $(\neg q \to p) \mathrel{|\!\!\sim} (8)$.

The previous analysis does not seem to solve our query in Example 3 in a satisfactory way. After all, an additional important premise we are not using is that auxiliary atom q is not supposed to appear anywhere else in the rest of the program. So, if our context never contains q, there is no way to form a positive loop between p and q. According to [19], disjunction (8) would always be *head-cycle-free* and could be "shifted" into (7) to obtain the same set of stable models. This points out that we should have some way to prove $(8) \cong_V (7)$ for any alphabet V not containing q. As we have seen, regular strong equivalence does not suffice for this purpose. We will be back to the example later on.

[4] The only difference we get when removing $\neg q \to p$ from (7) in Fig. 1 is that the row for $T = \{p\}$ gets an additional interpretation $\emptyset$. The same happens for $T = \{q\}$ when removing $\neg p \to q$ instead.

4. Forks and T-views

A *fork* is defined using the grammar:

$$F \ ::= \ \bot \ \big| \ p \ \big| \ (F \mid F) \ \big| \ F \wedge F \ \big| \ \varphi \vee \varphi \ \big| \ \varphi \rightarrow F$$

where φ is a propositional formula and $p \in At$ is an atom. Notice that the fork operator '|' cannot occur in the scope of disjunction or negation, since $\neg F$ stands for $F \rightarrow \bot$ and implications do not allow forks in the antecedent. This definition suffices for the purposes of the current paper. However, the extension of the semantics to arbitrarily nested connectives offers multiple alternatives and it is unclear yet which properties should be satisfied. For this reason, we leave the arbitrary nesting of forks for future work.

As we will see, a fork F will always be reducible to the form $(\varphi_1|\ldots|\varphi_n)$ where each φ_i is a formula. Thus, a natural way to define its semantics is keeping a set of supports $\Delta = \{\mathcal{H}_1, \ldots, \mathcal{H}_n\}$ for each classical interpretation T. However, this intuition needs to be refined: if we allowed a pair of supports in Δ such that $\mathcal{H}_i \preceq \mathcal{H}_j$, then $\mathcal{H}_i$ would be useless since, according to Proposition 8 ii), if it yields a stable model, the latter is always produced by $\mathcal{H}_j$ too. For this reason, we will collect sets of supports that are $\preceq$-closed, so their *maximal* elements become the representative ones. Formally, given a T-support $\mathcal{H}$ we define the set of (non-empty) $\preceq$-smaller supports $\downarrow\mathcal{H} = \{\mathcal{H}' \mid \mathcal{H}' \preceq \mathcal{H}\} \setminus \{\,[\ ]\,\}$. This is usually called the *ideal* of $\mathcal{H}$. Note that, the empty support $[\ ]$ is not included in the ideal. As a result, $\downarrow[\ ] = \emptyset$. We extend this notation to any set of supports Δ so that:

$$^{\downarrow}\Delta \stackrel{\text{def}}{=} \bigcup_{\mathcal{H}\in\Delta} \downarrow\mathcal{H} = \{\, \mathcal{H}' \mid \mathcal{H}' \preceq \mathcal{H}, \mathcal{H} \in \Delta \,\} \setminus \{\,[\,]\,\}$$

Definition 5 *(T-view).* A *T-view* is a set of T-supports $\Delta \subseteq \mathbf{H}_T$ that is $\preceq$-closed, i.e., $^{\downarrow}\Delta = \Delta$.

If Δ is a T-view and the $\preceq$-greatest T-support $[\ T\ ]$ is included in Δ, then Δ is precisely $\downarrow[\ T\ ]$. We proceed next to define the semantics of forks in terms of T-views. To emphasise the duality between conjunction and disjunction, we will be interested in dealing with a weaker version of the membership relation, $\hat{\in}$, defined as follows. Given a T-view Δ, we write $\mathcal{H} \,\hat{\in}\, \Delta$ iff $\mathcal{H} \in \Delta$ or both $\mathcal{H} = [\ ]$ and $\Delta = \emptyset$. We are now ready to extend the concept of T-denotation to forks.

Definition 6 *(T-denotation of a fork).* Let At be a propositional signature and $T \subseteq At$ a set of atoms. The *T-denotation* of a fork F, written $\langle\!\langle F \rangle\!\rangle^T$, is a T-view recursively defined as follows:

$$\begin{aligned}
\langle\!\langle \bot \rangle\!\rangle^T &\stackrel{\text{def}}{=} \emptyset \\
\langle\!\langle p \rangle\!\rangle^T &\stackrel{\text{def}}{=} \downarrow[\![\, p \,]\!]^T \quad \text{for any atom } p \\
\langle\!\langle F \wedge G \rangle\!\rangle^T &\stackrel{\text{def}}{=} {}^{\downarrow}\{\, \mathcal{H} \cap \mathcal{H}' \mid \mathcal{H} \in \langle\!\langle F \rangle\!\rangle^T \text{ and } \mathcal{H}' \in \langle\!\langle G \rangle\!\rangle^T \,\} \\
\langle\!\langle \varphi \vee \psi \rangle\!\rangle^T &\stackrel{\text{def}}{=} {}^{\downarrow}\{\, \mathcal{H} \cup \mathcal{H}' \mid \mathcal{H} \,\hat{\in}\, \langle\!\langle \varphi \rangle\!\rangle^T \text{ and } \mathcal{H}' \,\hat{\in}\, \langle\!\langle \psi \rangle\!\rangle^T \,\} \\
\langle\!\langle \varphi \rightarrow F \rangle\!\rangle^T &\stackrel{\text{def}}{=} \begin{cases} \{\, 2^T \,\} & \text{if } [\![\, \varphi \,]\!]^T = [\,] \\ {}^{\downarrow}\{\, \overline{[\![\, \varphi \,]\!]^T} \cup \mathcal{H} \mid \mathcal{H} \in \langle\!\langle F \rangle\!\rangle^T \,\} & \text{otherwise} \end{cases} \\
\langle\!\langle F \mid G \rangle\!\rangle^T &\stackrel{\text{def}}{=} \langle\!\langle F \rangle\!\rangle^T \cup \langle\!\langle G \rangle\!\rangle^T
\end{aligned}$$

We will see later that the fork operator '|' is commutative, associative and idempotent and, also that conjunction and implication distribute over '|'. As for the rest of operators, note that the definitions above also cover propositional formulas. The following result shows that this new T-denotation of a propositional formula φ as a T-view, $\langle\!\langle \varphi \rangle\!\rangle^T$, is precisely the ideal of its T-denotation as a T-support $[\![\, \varphi \,]\!]^T$.

Proposition 9. *Let φ be a propositional formula and $T \subseteq At$ be a set of atoms. Then, $\langle\!\langle \varphi \rangle\!\rangle^T = \downarrow[\![\, \varphi \,]\!]^T$.*

Corollary 2. *Given a propositional formula φ, a set $T \subseteq At$ of atoms is a stable model of φ iff $\langle\!\langle \varphi \rangle\!\rangle^T = \downarrow[\ T\ ]$.*

Corollary 2 provides a natural way to define stable models of forks.

Definition 7. Given a fork F, we say that $T \subseteq At$ is a *stable model* of F iff $\langle\!\langle F \rangle\!\rangle^T = \downarrow[\ T\ ]$ or, equivalently, $[\ T\ ] \in \langle\!\langle F \rangle\!\rangle^T$. $SM[F]$ denotes the set of stable models of F.

The intuition behind a fork is that we can collect its stable models independently:

Proposition 10. *Given forks F and G: $SM[F \mid G] = SM[F] \cup SM[G]$.*

Once $SM[F]$ is defined, we can immediately extend the definition of V-strong entailment and equivalence to forks in the obvious way, i.e., using forks instead of propositional formulas. To be precise:

Definition 8 *(Projective strong entailment/equivalence of forks).* Let F and G be two forks and $V \subseteq At$ some vocabulary (set of atoms). We say that F *V-strongly entails* G, written $F \mathrel{|\!\!\sim}_V G$ if $SM_V[F \wedge L] \subseteq SM_V[G \wedge L]$ for any fork L such that $At(L) \subseteq V$. We further say that F and G are *V-strongly equivalent*, written $F \cong_V G$, if both $F \mathrel{|\!\!\sim}_V G$ and $G \mathrel{|\!\!\sim}_V F$, that is, $SM_V[F \wedge L] = SM_V[G \wedge L]$ for any fork L such that $At(L) \subseteq V$.

As before, when $V \supseteq At(F) \cup At(G)$ we talk about non-projective strong entailment/equivalence and just drop the V subindex. We postpone the effect of projecting onto some vocabulary V to the next section and focus on the regular, non-projected versions $\mathrel{|\!\!\sim}$ and $\cong$. As in Proposition 8, $\mathrel{|\!\!\sim}$ and $\cong$ have a simple characterisation in terms of denotations:

Proposition 11. *For any pair of forks F, G the following hold:*

(i) *$F \mathrel{|\!\!\sim} G$ iff $\langle\!\langle F \rangle\!\rangle^T \subseteq \langle\!\langle G \rangle\!\rangle^T$ for every set $T \subseteq At$,*
(ii) *$F \cong G$ iff $\langle\!\langle F \rangle\!\rangle^T = \langle\!\langle G \rangle\!\rangle^T$ for every set $T \subseteq At$.*

This helps us to derive the following interesting properties:

Proposition 12. *Let F, G, L be arbitrary forks and φ a formula. Then:*

$$(F \mid G) \mid L \cong F \mid (G \mid L) \tag{9}$$
$$F \mid G \cong G \mid F \tag{10}$$
$$(F \mid G) \cong G \qquad \text{if } F \mathrel{|\!\!\sim} G \tag{11}$$
$$(F \mid G) \wedge L \cong (F \wedge L) \mid (G \wedge L) \tag{12}$$
$$\varphi \to (F \mid G) \cong (\varphi \to F) \mid (\varphi \to G) \tag{13}$$
$$\varphi \to F \wedge G \cong (\varphi \to F) \wedge (\varphi \to G) \tag{14}$$
$$\varphi \to (\psi \to F) \cong \varphi \wedge \psi \to F \tag{15}$$
$$\top \to F \cong F \tag{16}$$
$$\neg\varphi \mid \neg\neg\varphi \cong \top \tag{17}$$

As we can see, (9) and (10) respectively guarantee that the fork operator '|' is associative and commutative. Property (11) is a kind of "subsumption": when we have $F \mathrel{|\!\!\sim} G$, then fork F is subsumed by G in $F \mid G$ and so, F can be removed. As an example of subsumption, take the fork $(\neg p \to q) \mid (p \vee q)$. As we saw before, $(\neg p \to q) \mathrel{|\!\!\sim} (p \vee q)$ because $[\![\neg p \to q]\!]^T \preceq [\![p \vee q]\!]^T$ for all T. Then, the ideal $\downarrow\![\![\neg p \to q]\!]^T$ is included in $\downarrow\![\![p \vee q]\!]^T$ which (by Proposition 9) is the same as saying $\langle\!\langle \neg p \to q \rangle\!\rangle^T \subseteq \langle\!\langle p \vee q \rangle\!\rangle^T$. But then, $\langle\!\langle (\neg p \to q) \mid (p \vee q) \rangle\!\rangle^T = \langle\!\langle \neg p \to q \rangle\!\rangle^T \cup \langle\!\langle p \vee q \rangle\!\rangle^T = \langle\!\langle p \vee q \rangle\!\rangle^T$. In other words, $(\neg p \to q) \mid (p \vee q) \cong (p \vee q)$.

Properties (12) and (13) directly imply that we can reformulate any fork in a normal form where '|' is applied on a sequence of formulas:

Corollary 3. *For any fork F, there is a strongly equivalent fork $G \cong F$ of the form $G = (\varphi_1 \mid \ldots \mid \varphi_n)$ where each φ_i is a propositional formula.* □

Note that, in general, the reduction of F into normal form may lead to a final result G exponentially larger than F, due to the application of distributivity (12). Another important consequence of conjunction-distributivity (12) is that, although $\wedge$ is idempotent on propositional formulas, it ceases to be so when connecting forks. In other words, $(F \mid F) \not\cong F$ in the general case. As an illustration:

Example 4 *(Counterexample of idempotence).* Take, for instance, the formalisation of Example 1 using the expression $(a \mid b) \wedge (a \mid b)$. If we apply distributivity and reduce to a normal form:

$$\begin{array}{rcll} (a \mid b) \wedge (a \mid b) & \cong & (a \wedge (a \mid b)) \ \mid \ (b \wedge (a \mid b)) & \text{distributivity (12)} \\ & \cong & (a \wedge a) \mid (a \wedge b) \mid (b \wedge a) \mid (b \wedge b) & \text{distributivity (12) and assoc. of } \mid \\ & \cong & a \mid (a \wedge b) \mid (b \wedge a) \mid b & \wedge\text{-idempotence} \\ & \cong & a \mid (a \wedge b) \mid b & \text{commut. of } \wedge \text{ and `}\mid\text{'-idempotence} \end{array}$$

but then, from the last expression, we must get stable model $\{a, b\}$ from $a \wedge b$ (Proposition 10) whereas $\{a, b\}$ is not among the stable models of $(a \mid b)$. □

5. Projective strong equivalence/entailment

In this section, we provide a semantic characterisation of projective strong entailment $\mathrel{|\!\!\sim}_V$ and equivalence $\cong_V$ for some vocabulary $V \subseteq At$. We say that a T-support $\mathcal{H}$ is V*-unfeasible*[5] iff there is some $H \subset T$ in $\mathcal{H}$ satisfying $H \cap V = T \cap V$; we call it V*-feasible* otherwise. The reason for the name "unfeasible" is that, if we take a formula φ with denotation $[\![\varphi]\!]^T = \mathcal{H}$, then T will never become stable if we are only allowed to use vocabulary V when adding a context γ. To do so, we would need $[\![\varphi \wedge \gamma]\!]^T = [\ T\ ]$ but $H \subset T$ should also belong to the support since H and T are indistinguishable for any γ over V.

Definition 9. Let $V \subseteq At$ be a vocabulary and $T \subseteq V$ be a set of atoms. Then, the V-T*-denotation* of a fork F is a T-view defined as follows:

$$\langle\!\langle\, F \,\rangle\!\rangle^T_V \stackrel{\text{def}}{=} {\downarrow}\{\, \mathcal{H}_V \;\big|\; \mathcal{H} \in \langle\!\langle\, F \,\rangle\!\rangle^{T'} \text{ s.t. } T' \cap V = T \text{ and } \mathcal{H} \text{ is } V\text{-feasible} \,\}$$

In other words, we collect all the feasible supports $\mathcal{H}$ that belong to any T'-denotation $\langle\!\langle\, F \,\rangle\!\rangle^{T'}$ such that T' coincides with T for atoms in V, and then we project the supports taking $\mathcal{H}_V$. In doing so, we can just consider maximal $\mathcal{H}$'s in $\langle\!\langle\, F \,\rangle\!\rangle^{T'}$. As might be expected, projecting the T-denotation of a fork F on a superset $V \supseteq At(F)$ of its atoms produces no effect:

Proposition 13. *For any vocabulary $V \subseteq At$, fork F with $At(F) \subseteq V$ and set $T \subseteq V$ of atoms, $\langle\!\langle\, F \,\rangle\!\rangle^T_V = \langle\!\langle\, F \,\rangle\!\rangle^T$.*

More interestingly, the V-T-denotation of F can be used to precisely characterise its projected stable models.

Proposition 14. *For any vocabulary $V \subseteq At$, fork F and set $T \subseteq V$ of atoms, it holds that $T \in SM_V[F]$ iff $\langle\!\langle\, F \,\rangle\!\rangle^T_V = {\downarrow}[\ T\ ]$.*

We naturally extend the definitions of projective strong entailment/equivalence of Definition 1 for propositional formulas to the case of forks. As a main result, we also extend Proposition 11 to the projective case.

Theorem 2. For any vocabulary $V \subseteq At$, forks F, G, the following hold:

(i) $F \mathrel{|\!\!\sim}_V G$ iff $\langle\!\langle\, F \,\rangle\!\rangle^T_V \subseteq \langle\!\langle\, G \,\rangle\!\rangle^T_V$ for every set $T \subseteq V$ of atoms, and
(ii) $F \cong_V G$ iff $\langle\!\langle\, F \,\rangle\!\rangle^T_V = \langle\!\langle\, G \,\rangle\!\rangle^T_V$ for every set $T \subseteq V$ of atoms.

Moreover, the following result shows that we can just use a formula as a context instead of an arbitrary fork.

Proposition 15. *For any vocabulary $V \subseteq At$, forks F, G, the following hold:*

(i) *$F \mathrel{|\!\!\sim}_V G$ iff $SM_V[F \wedge \gamma] \subseteq SM_V[G \wedge \gamma]$ for any formula γ with $At(\gamma) \subseteq V$,*
(ii) *$F \cong_V G$ iff $SM_V[F \wedge \gamma] = SM_V[G \wedge \gamma]$ for any formula γ with $At(\gamma) \subseteq V$.*

As an immediate consequence we can extend the characterisation of PSE from disjunctive logic programs in [12] to arbitrary propositional formulas.

[5] This notion is analogous to condition *ii*) in the definition of V-SE-models that characterises relativised strong equivalence [20].

Corollary 4. *For any vocabulary $V \subseteq At$, formulas φ, ψ, the following hold:*

(i) $\varphi \mathrel{|\!\sim}_V \psi$ *iff* $\langle\!\langle \varphi \rangle\!\rangle^T_V \subseteq \langle\!\langle \psi \rangle\!\rangle^T_V$ *for every set* $T \subseteq V$ *of atoms, and*
(ii) $\varphi \cong_V \psi$ *iff* $\langle\!\langle \varphi \rangle\!\rangle^T_V = \langle\!\langle \psi \rangle\!\rangle^T_V$ *for every set* $T \subseteq V$ *of atoms.*

The next result guarantees that, for studying $F \mathrel{|\!\sim}_V G$ (and so, $F \cong_V G$ too), atoms not occurring in those forks become irrelevant.

Theorem 3 *(Free atom invariance).* Let F and G be two forks and let At be a signature such that $At \supset At(F) \cup At(G)$ and $a \in At \setminus (At(F) \cup At(G))$, for some atom a. For any $V \subseteq At$ we have: $F \mathrel{|\!\sim}_V G$ for signature At iff $F \mathrel{|\!\sim}_{V'} G$ for signature $At' = At \setminus \{a\}$ and $V' = V \setminus \{a\}$.

As a corollary, we can analyse $F \mathrel{|\!\sim}_V G$ with $At \supseteq At(F) \cup At(G)$ and projections $V \subseteq At$ by just exclusively focusing on signature $At' = At(F) \cup At(G)$ and projecting on $V' = V \cap At'$.

Corollary 5 *(Signature independence). Let F and G be two forks for a signature At that may include additional atoms, that is, $At(F) \cup At(G) \subseteq At$. Given $V \subseteq At$, then $F \mathrel{|\!\sim}_V G$ for signature At is equivalent to $F \mathrel{|\!\sim}_{V'} G$ for $V' = V \cap At'$ under any signature At' with $At(F) \cup At(G) \subseteq At' \subseteq At$.*

Example 5 *(Example 3 continued).* Back to Example 3, we can now check the PSE of eqreff:pqloop $= (\neg p \to q) \wedge (\neg q \to p)$ and $(8) = (p \vee q)$. To do so, we have to test if $\langle\!\langle (7) \rangle\!\rangle^T_V = \langle\!\langle (8) \rangle\!\rangle^T_V$ for every $T \subseteq V$ and any vocabulary V not containing q. According to the corollary above (signature independence), we can just restrict the study to $At = \{p, q\}$ and $V = \{p\}$. Take $\langle\!\langle (7) \rangle\!\rangle^T_V$ first. Definition 9 starts from (non-projective) denotations $\langle\!\langle (7) \rangle\!\rangle^{T'}$ for T' in the original signature $At = \{p, q\}$. Since (7) is a propositional formula (it contains no forks), by Proposition 9 we have $\langle\!\langle (7) \rangle\!\rangle^{T'} = {\downarrow}[\![(7)]\!]^{T'}$, that is, the maximal supports are just the T-denotations we already obtained in Fig. 1. Now, for vocabulary $V = \{p\}$ we may only have $T = \{p\}$ and $T = \emptyset$. For the first case, the potential candidates T' such that $T' \cap V = \{p\}$ are the rows $T' = \{p\}$ and $T' = \{p, q\}$ from Fig. 1. However, the support we have for latter is V-unfeasible because it contains $H = \{p\}$ such that $H \subset \{p, q\} = T'$ but they coincide in the truth of $V = \{p\}$. Thus, for $T = \{p\}$ we only have the feasible (maximum) support $[\{p\}]$ from $T' = \{p\}$ which yields $\langle\!\langle (7) \rangle\!\rangle^{\{p\}}_V = {\downarrow}[\ \{p\}\ ]$. For $T = \emptyset$ the only non-empty case is $T' = \{q\}$ and, after removing atom q, we obtain $[\emptyset]$ as maximum support, i.e., $\langle\!\langle (7) \rangle\!\rangle^{\emptyset}_V = {\downarrow}[\ \emptyset\ ]$. It is easy to see that (8) yields the same result: the only difference we had was in the last row of Fig. 1, for $T' = \{p, q\}$, but this support is V-unfeasible again, as it also contains $H = \{p\}$.

To sum up $\langle\!\langle (7) \rangle\!\rangle^{\{p\}}_V = \langle\!\langle (8) \rangle\!\rangle^{\{p\}}_V = {\downarrow}[\ \{p\}\ ]$ and $\langle\!\langle (7) \rangle\!\rangle^{\emptyset}_V = \langle\!\langle (8) \rangle\!\rangle^{\emptyset}_V = {\downarrow}[\ \emptyset\ ]$ so both formulas satisfy PSE, $(7) \cong_V (8)$. In other words, they have the same behaviour in any context, assuming that $q \notin V$ is an auxiliary atom. It is not difficult to see that, as the auxiliary atom is wiped out in the process, its name can be always changed by a fresh atom. In other words, to put an example, we also have $(7) \cong_V (p \vee q')$ for any alphabet V such that $\{q, q'\} \cap V = \emptyset$. □

One important remark regarding projective strong equivalence $\varphi \cong_V \psi$ is that replacements of φ by ψ are not guaranteed to be safe at *any* subformula level, as happened with regular strong equivalence $\varphi \cong \psi$. Indeed, since the latter corresponds to HT-equivalence [11] and HT satisfies the law of substitution of equivalents, we know that, for instance, $\varphi \cong \psi$ also implies $\varphi \vee \gamma \cong \psi \vee \gamma$ or $\varphi \to \gamma \cong \psi \to \gamma$ to put a pair of examples. Unfortunately, this property is lost when we consider $\cong_V$ instead, which is only safe for replacements in conjunctions or replacements of complete formulas inside a theory. As a counterexample, take $\varphi = q$ and $\psi = \neg q$ and assume we study their projective strong equivalence $q \cong_V \neg q$ with respect to a vocabulary V that does not contain q. Thanks to signature independence (Corollary 5) we can just focus on denotations with respect to signature $At' = \{q\}$, i.e., the only atom occurring in these formulas:

T'	$[\![q]\!]^{T'}$	$[\![\neg q]\!]^{T'}$
$\emptyset$	$[\ ]$	$[\ \emptyset\ ]$
$\{q\}$	$[\ \{q\}\ ]$	$[\ ]$

Note that all supports are empty or singletons, so they are feasible. Now, for the projection on V, due to Corollary 5 again, we can take $V' = V \cap At' = \emptyset$ since $q \notin V$. The only possible classical interpretation for an empty vocabulary $V' = \emptyset$ is $T = \emptyset$ (all atoms false). To compute $\langle\!\langle q \rangle\!\rangle^{\emptyset}_{\emptyset}$, the only possibility is using $T' = \{q\}$ since its support $[\ \{q\}\ ]$ is non-empty and feasible. After removing atom q we get $\langle\!\langle q \rangle\!\rangle^{\emptyset}_{\emptyset} = {\downarrow}[\ \emptyset\ ]$. Similarly, for $\langle\!\langle \neg q \rangle\!\rangle^{\emptyset}_{\emptyset}$ the only possibility is $T' = \emptyset$ and we get $\langle\!\langle \neg q \rangle\!\rangle^{\emptyset}_{\emptyset} = {\downarrow}[\ \emptyset\ ]$ which, as we can see, coincides with $\langle\!\langle \neg q \rangle\!\rangle$, and so $q \cong_V \neg q$. Moreover, in an empty signature, the denotation we have obtained ${\downarrow}[\ \emptyset\ ]$ corresponds to a tautology $\top$. This result is not surprising since atom q is going to be ignored in the "rest of the theory" and formulas $\varphi = q$ and $\psi = \neg q$ only affect to the truth of q but not to the rest of the signature. Thus $q \wedge \gamma \cong_V \neg q \wedge \gamma$ for any formula γ in an alphabet V not containing q. Moreover, since q and $\neg q$ behave as $\top$ with respect to the rest of the theory, we even have the stronger condition $q \wedge \gamma \cong_V \neg q \wedge \gamma \cong_V \gamma$.

Now, if we try to extrapolate this result to occurrences of q or $\neg q$ as subformulas in other expressions, PSE is not a guarantee any more. To see why, take now the formulas $q \vee p$ and $\neg q \vee p$. If we just consider projective equivalence, we can immediately see that $q \vee p$ has stable model $\{p\}$ while $\neg q \vee p$ has the unique stable model $\emptyset$ and they do not coincide for atom p (which is not auxiliary). So, they cannot satisfy PSE since they do not even satisfy projective equivalence. As expected from Corollary 4, it is not difficult to check that their V-T-denotations differ. Note that $q \vee p$ is equivalent to (8) and we had already obtained $\langle\!\langle (8) \rangle\!\rangle_V^{\{p\}} = \downarrow[\ \{p\}\ ]$ and $\langle\!\langle (8) \rangle\!\rangle_V^{\emptyset} = \downarrow[\ \emptyset\ ]$ Without entering into further details, it can be checked that the (non-empty) V-T-denotations for $\neg q \vee p$ correspond to $\langle\!\langle \neg q \vee p \rangle\!\rangle_V^{\{p\}} = \downarrow[\ \{p\}\ \emptyset\ ]$ and $\langle\!\langle \neg q \vee p \rangle\!\rangle_V^{\emptyset} = \downarrow[\ \emptyset\ ]$.

6. Forgetting

As we mentioned in the introduction, PSE is closely related to forgetting. Let At be a signature and $V \subseteq At$. Given an expression φ over At, a *forgetting operator* is a partial function $\mathtt{f}(\varphi, V) = \psi$ that assigns a new expression ψ over[6] V. Operator $\mathtt{f}$ is said to be *strongly persistent* iff $\varphi \cong_V \mathtt{f}(\varphi, V)$ for every formula φ and set $V \subseteq At$ of atoms for which it is defined. Now, imagine we wish to apply forgetting on a fork F over At keeping atoms $V \subseteq At$. In light of Corollary 4, we can start by obtaining the projected denotations $\langle\!\langle F \rangle\!\rangle_V^T$ for all $T \subseteq V$. This corresponds to a set of T-views that can be precisely captured by another fork over V, as stated by the following result.

Proposition 16. *Let V be a vocabulary $V \subseteq At$, and σ a mapping that assigns some arbitrary T-view $\sigma(T)$ to each $T \subseteq V$. Let us define the set of sets of HT-interpretations:*

$$\Sigma \stackrel{\text{def}}{=} \{\ \{\langle H, T\rangle \mid H \in \mathcal{H}\}\ \mid\ T \subseteq V,\ \mathcal{H} \text{ maximal } T\text{-view in } \sigma(T)\}$$

and enumerate them as $\Sigma = \{S_1, \ldots, S_n\}$. Then, fork $G_\sigma \stackrel{\text{def}}{=} (\Phi_{S_1} \mid \cdots \mid \Phi_{S_n})$ captures σ, that is, $\langle\!\langle G_\sigma \rangle\!\rangle^T = \sigma(T)$ for all $T \subseteq V$. □

Example 6. Suppose we have the mapping σ:

T	maximal supports in $\sigma(T)$
$\{a\}$	$[\ \{a\}\ ]$
$\{b\}$	$[\ \{b\}\ ]$
$\{a,b\}$	$[\ \{a,b\}\ \{b\}\ ]$
	$[\ \{a,b\}\ \{a\}\ ]$

Applying Proposition 16 the set Σ will contain four sets of interpretations, corresponding to the four maximal supports in $\sigma(T)$. They yield the formulas:

i	S_i	Φ_{S_i}
1	$\{\langle\{a\},\{a\}\rangle\}$	$a \wedge \neg b$
2	$\{\langle\{b\},\{b\}\rangle\}$	$b \wedge \neg a$
3	$\{\langle\{b\},\{a,b\}\rangle, \langle\{a,b\},\{a,b\}\rangle\}$	$(b \wedge \neg\neg a) \vee (a \wedge b)$
4	$\{\langle\{a\},\{a,b\}\rangle, \langle\{a,b\},\{a,b\}\rangle\}$	$(a \wedge \neg\neg b) \vee (a \wedge b)$

In HT, the formula $(a \wedge b)$ is stronger than the formulas $(b \wedge \neg\neg a)$ and $(a \wedge \neg\neg b)$. Thus, we can remove $(a \wedge b)$ in the last two disjunctions. This leaves us the fork $G_\sigma = (a \wedge \neg b) \mid (b \wedge \neg a) \mid (a \wedge \neg\neg b) \mid (b \wedge \neg\neg a)$. Applying distributivity, we get $(a \wedge (\neg b \mid \neg\neg b)) \mid (b \wedge (\neg a \mid \neg\neg a))$. Finally, given (17) this amounts to the final fork $G_\sigma \cong (a \mid b)$. □

Definition 10 *(Forgetting operator).* We define the operator $\mathtt{fk}(F, V) \stackrel{\text{def}}{=} G_\sigma$ where G_σ is obtained as in Proposition 16 above taking the mapping $\sigma(T) = \langle\!\langle F \rangle\!\rangle_V^T$. □

Theorem 4. Operator $\mathtt{fk}$ is a total, strongly persistent forgetting operator over forks. In other words, for every fork F and set $V \subseteq At$ of atoms, $At(\mathtt{fk}(F, V)) \subseteq V$ and $F \cong_V \mathtt{fk}(F, V)$. □

Recall from [10] that such a total operator has been shown not to exist for HT. However, since every propositional formula is also a fork, forgetting in HT is now possible if we allow the target language to be extended with the fork '|' operator – "we can always forget as a fork." Furthermore, the following relation between the T-denotation of forks and of formulas sheds light to the reason why it is not possible to forget inside HT.

[6] Note that we are defining the forgetting operator with respect to the projected signature instead of the forgotten atoms $At \setminus V$, which is the usual definition in the literature.

T'	maximal supports in $\langle\!\langle P_m \rangle\!\rangle^{T'}$
$\{ma, a\}$	$[\ \{ma, a\}\ ]$
$\{mb, b\}$	$[\ \{mb, b\}\ ]$
$\{ma, a, b\}$	$[\ \{ma, a, b\}\ \{ma, a\}\ ]$
$\{mb, a, b\}$	$[\ \{mb, a, b\}\ \{mb, b\}\ ]$
$\{ma, mb, a, b\}$	$[\ \{ma, mb, a, b\}$ $\{mb, a, b\}\ \{mb, b\}$ $\{ma, a, b\}\ \{ma, a\}\ ]$

T	maximal supports in $\langle\!\langle P_m \rangle\!\rangle^T_V$
$\{a\}$	$[\ \{a\}\ ]$
$\{b\}$	$[\ \{b\}\ ]$
$\{a, b\}$	$[\ \{a, b\}\ \{b\}\ ]$ $[\ \{a, b\}\ \{a\}\ ]$

Fig. 2. Forgetting ma and mb in P_m.

Proposition 17. *Given sets $T \subseteq V \subseteq At$ of atoms, then:*

(i) *any formula φ with $At(\varphi) \subseteq V$ satisfies $\langle\!\langle \varphi \rangle\!\rangle^T_V = \downarrow [\![\varphi]\!]^T$ and, thus, $\langle\!\langle \varphi \rangle\!\rangle^T_V$ has a $\preceq$-maximum element;*
(ii) *for every T-view Δ with a $\preceq$-maximum element, there is a propositional formula φ with $At(\varphi) \subseteq V$ that satisfies $\langle\!\langle \varphi \rangle\!\rangle^T_V = \Delta$ and $\langle\!\langle \varphi \rangle\!\rangle^{T'}_V = \emptyset$ for every $T' \subseteq V$ with $T' \neq T$.*

That is, there is a one-to-one correspondence between each assignment of T-views with some $\preceq$-maximum (i.e. *unique* maximal) element and a formula modulo strong equivalence in the vocabulary V. On the other hand, there are T-views that have more than one $\preceq$-maximal element and, thus, they *cannot be represented* as formulas. That is, there are more theories modulo $\cong_V$ than theories over V modulo $\cong$.

Example 7 *(Examples 3 and 5 continued).* To illustrate Proposition 17, let us return to our Example 3. To forget the auxiliary atom q in (7)=$(\neg p \rightarrow q) \wedge (\neg q \rightarrow p)$ we can use its $\{p\}$-T-denotation we had already computed. As we saw, both $\langle\!\langle (7) \rangle\!\rangle^{\{p\}}_V = \downarrow[\ \{p\}\ ]$ and $\langle\!\langle (7) \rangle\!\rangle^{\emptyset}_V = \downarrow[\ \emptyset\ ]$ have a unique, maximum support in their respective views. This means that the obtained projection can be represented as a formula φ such that $[\![\varphi]\!]^{\{p\}} = [\{p\}]$ and $[\![\varphi]\!]^{\emptyset} = [\emptyset]$ or, if preferred, whose HT models are $\langle \{p\}, \{p\} \rangle$ and $\langle \emptyset, \emptyset \rangle$. It is easy to see that such a formula is precisely $p \vee \neg p$ (or, equivalently, $\neg\neg p \rightarrow p$) which is also a well-known representation of a choice that does not require auxiliary atoms. To conclude Example 3, we remind that (8)=$(p \vee q)$ and notice that we have just proved in this way that (7) $\cong_V$ (8) $\cong_V$ $p \vee \neg p$ so the three representations are strongly equivalent modulo auxiliary atoms. □

Example 8. To see an example of forgetting that is not reducible to a propositional formula, consider program P_m from the introduction interpreted as the conjunction of its rules, and assume we want to forget ma and mb. Let us take all subsets $T' \subseteq \{a, b, ma, mb\}$. Since P_m is a formula, all its T'-views will have a $\preceq$-maximum element, $[\![P_m]\!]^{T'}$, shown in the left table of Fig. 2 where, for brevity, we only show cases of non-empty supports (i.e., when T' is a classical model).

Now, according to the definition of $\langle\!\langle P_m \rangle\!\rangle^T_V$, for each $T \subseteq V = \{a, b\}$ we must find those T' such that $T' \cap \{a, b\} = T$. For $T = \{a\}$ the only possibility is $T' = \{ma, a\}$ that, after removing ma, yields a maximum support $[\{a\}]$. The case for $T = \{b\}$ is completely symmetric, yielding maximum support $[\{b\}]$. But for $T = \{a, b\}$ we get three candidate interpretations, $T'_1 = \{ma, a, b\}$, $T'_2 = \{mb, a, b\}$ and $T'_3 = \{ma, mb, a, b\}$. A first observation is that the support for $T'_3 = \{ma, mb, a, b\}$ is $\{a, b\}$-unfeasible, since it contains $\{ma, a, b\}$ and $\{mb, a, b\}$ that coincide with T'_3 for atoms $\{a, b\}$. After removing ma, mb in the supports of the feasible candidates, T'_1 and T'_2, the respective results are $[\ \{a, b\}\ \{a\}\ ]$ and $[\ \{a, b\}\ \{b\}\ ]$ that are not $\preceq$-comparable. Therefore, they become the two $\preceq$-maximal supports in the T-view $\langle\!\langle P_m \rangle\!\rangle^{\{a,b\}}_V$. Proposition 17 tells us that this *is not representable* as a propositional formula, although Proposition 16 always guarantees a representation as a fork. In particular, the denotations we obtained coincide with the mapping σ from Example 6, where we used Proposition 16 to obtain the fork $G_\sigma \cong (a \mid b)$, that is the result we were expecting from the introduction. Analogously, forgetting fa, fb in P_f yields a second fork $(a \mid b)$. As for the whole program $P_2 = P_m \wedge P_f$, its (non-empty) V-T-denotations are:

T	maximal supports in $\langle\!\langle P_2 \rangle\!\rangle^T_V$
$\{a\}$	$[\ \{a\}\ ]$
$\{b\}$	$[\ \{b\}\ ]$
$\{a, b\}$	$[\ \{a, b\}\ ]$

that, as we can see, have one $\preceq$-maximum T-support per each T-view. This implies that P_2 *is representable* as a propositional formula over V. In fact, the denotations $\langle\!\langle P_2 \rangle\!\rangle^T_V$ coincide with $\langle\!\langle P_1 \rangle\!\rangle^T$ for every T, so P_1 is the result of forgetting ma and mb in P_2. Moreover, if we represent Example 1 using a fork per each parent, $(a \mid b) \wedge (a \mid b)$ we get again the same denotations, that is, $P_1 \cong (a \mid b) \wedge (a \mid b) \cong (a \mid b \mid (a \wedge b))$ (as we saw in Example 2). □

Our running example has illustrated that constructions like $(a \mid b)$ cannot be represented as propositional formulas or logic programs, showing the richer expressiveness of forks *under the assumption of a fixed vocabulary*. However, if auxiliary atoms are allowed, we have seen that $(a \mid b)$ can be represented as program P_m using the hidden atoms ma and mb. In fact, the rules we used in P_m to encode $(a \mid b)$ can be extrapolated so that any fork in normal form can be represented as a formula with additional fresh atoms, as stated in the following result:

Proposition 18. *Let $F = (\varphi_1 \mid \cdots \mid \varphi_n)$ be a fork in normal form, $V = At(F)$ and let*

$$\gamma(F) = (a_1 \vee \cdots \vee a_n) \wedge (a_1 \to \varphi_1) \wedge \cdots \wedge (a_n \to \varphi_n)$$

be a formula with $a_i \notin V$ for all $1 \le i \le n$. Then, $F \cong_V \gamma(F)$.

Note that the encoding $\gamma(F)$ has linear size with respect to F and that $\gamma(F)$ is always a formula. Therefore, the complexity of brave and cautious reasoning for forks in normal form amounts to Σ_2^P and Π_2^P-complete, as happens for formulas in equilibrium logic [21]. If we jump to arbitrary forks, we can always transform them into normal form and apply Proposition 18 to conclude:

Corollary 6. *Any fork F can be rewritten as an $At(F)$-equivalent propositional formula φ that may contain auxiliary atoms.*

However, as we explained before, the straightforward reduction into normal form may cause an exponential blow up due to distributivity (13). A polynomial reduction to normal form that introduces auxiliary atoms is still under study.

7. Related work

In this section, we study relations to the two main topics in the paper: forgetting and (variants of) strong equivalence.

7.1. Relation to forgetting

The introduction of forks in logic programs has allowed us to define a forgetting operator $\mathtt{fk}(F, V)$ that is total and *modular*, that is: $\mathtt{fk}(\mathtt{fk}(F, V), V') \cong \mathtt{fk}(F, V')$ for any $V' \subseteq V$. This means that there is no difference between simultaneously forgetting a group of atoms[7] comparison purposes, in the rest of this section, we consider the restriction of $\mathtt{fk}(\varphi, V)$ to propositional formulas, that is, we consider the cases where $\mathtt{fk}(\varphi, V)$ is some formula ψ. Notice that, under this restriction, operator $\mathtt{fk}$ is not always defined: this is a consequence of the impossibility of arbitrary forgetting proved in [10]. As we saw, this is the case because some applications of $\mathtt{fk}$ on formulas produce a fork that is not representable as a formula. For instance, we saw in Example 8 that $\mathtt{fk}(P_m, \{a, b\}) = (a \mid b)$ and this fork has no representation as a formula. Still, when $\mathtt{fk}$ produces a formula, Theorem 4 guarantees that strong persistence is preserved.

The approach in [10] not only detected that strongly persistent operators are partial, but further established a condition that precisely captures when these operators are defined (i.e., when it is possible to forget). We prove next that there is a one-to-one correspondence between that condition and the cases in which $\mathtt{fk}(\varphi, V)$ can be represented as a formula (Proposition 17). We begin recalling some definitions from [10].

Definition 11 *(Forgetting instance, Definition 1 from [10]).* Let $\mathcal{C}$ be a class of programs over a signature At. A (forgetting) instance (over $\mathcal{C}$) is a pair $\langle P, U\rangle$ such that $P \in \mathcal{C}$ and $U \subseteq At$.

Definition 12 *(Strong Persistence for Forgetting Instance, Definition 2 from [10]).* A forgetting operator $\mathtt{f}$ over $\mathcal{C}$ satisfies $\mathbf{SP}_{\langle P,U\rangle}$ if $SM[\mathtt{f}(P, V) \cup R] = SM_V[P \cup R]$ for all programs R with $At(R) \subseteq V$, where $V = (At \setminus U)$. Also, $\mathtt{f}$ satisfies $\mathbf{SP}_U$ if $\mathtt{f}$ satisfies $\mathbf{SP}_{\langle P,U\rangle}$ for all $P \in \mathcal{C}$.

Clearly, a forgetting operator $\mathtt{f}$ over $\mathcal{C}$ satisfies $\mathbf{SP}_{\langle P,U\rangle}$ iff $\mathtt{f}(P, V) \cong_V P$.

Definition 13 *(Criterion Ω, Definition 3 from [10]).* An instance $\langle P, U\rangle$ satisfies criterion Ω if there exists $Y \subseteq V$ with $V = (At \setminus U)$ such that the set of sets:

$$\mathcal{R}^Y_{\langle P,U\rangle} = \{\, R^{Y,A}_{\langle P,U\rangle} \mid A \in Rel^Y_{\langle P,U\rangle} \,\}$$

is non-empty and has no least element, where

[7] Remember that, in this paper, $\mathtt{f}(F, V)$ means forgetting atoms $At \setminus V$. or forgetting them one by one in any order. Previous approaches in the literature were obviously focused on programs without the (here introduced) forks. Therefore, for.

$$R^{Y,A}_{\langle P,U\rangle} = \{\, X \cap V \mid \langle X, Y \cup A\rangle \models P \,\}$$
$$Rel^{Y}_{\langle P,U\rangle} = \{\, A \subseteq U \mid Y \cup A \in [\![P]\!]^{Y\cup A} \text{ and } \forall A' \subset A,\ Y \cup A' \notin [\![P]\!]^{Y\cup A} \,\}$$

Proposition 19. *Given $V = (At \setminus U)$, the following conditions are equivalent:*

- *$\langle P, U\rangle$ does not satisfy Ω*
- *for any $Y \subseteq V$, $\langle\!\langle P \rangle\!\rangle^Y_V$ has only one maximal support*

Proof. Notice that, for any $Y \subseteq V$ and $A \subseteq U$, we have $(Y \cup A) \cap V = Y$. Moreover, the set of supports $[\![P]\!]^S$ with $S \subseteq At$ such that $S \cap V = Y$ and $[\![P]\!]^S$ is V-feasible is the same as the set of supports $[\![P]\!]^{Y\cup A}$ with $A \subseteq U$ and such that for all $A' \subset A$, $Y \cup A' \notin [\![P]\!]^{Y\cup A}$. Besides, we have:

$$R^{Y,A}_{\langle P,U\rangle} = [\![P]\!]^{Y\cup A}_V.$$

Consequently, the set $\mathcal{R}^Y_{\langle P,U\rangle}$ represents the Y-view consisting of all Y-supports of the form $[\![P]\!]^S_V$ where $S \subseteq At$ is such that $S \cap V = Y$ and $[\![P]\!]^S$ is V-feasible, so the minimal elements (considering $\subseteq$) of $\mathcal{R}^Y_{\langle P,U\rangle}$ are just the maximal elements (considering $\preceq$) of $\langle\!\langle P \rangle\!\rangle^Y_V$. Therefore, we can conclude that $\langle P, U\rangle$ does not satisfy criterion Ω iff for any $Y \subseteq V$, the Y-view $\langle\!\langle P \rangle\!\rangle^Y_V$ has only one maximal support. $\square$

The proof of the previous proposition shows that the Ω criterion coincides with the condition in Proposition 17, that is, that each T-view in the denotation contains a unique maximal support. Consequently:

Theorem 5 *(Theorem 2 and Corollary 1 from [10]).* There is a forgetting operator $\mathtt{f}$ that satisfies $\mathbf{SP}_{\langle P,U\rangle}$ iff $\langle P, U\rangle$ does not satisfy Ω.

Proof. In view of Proposition 17 and Proposition 19, the fork $\mathtt{f}(P, V)$ can be represented as a formula iff $\langle P, U\rangle$ does not satisfy Ω. $\square$

Our main goal when defining the operator $\mathtt{fk}$ was to remove auxiliary atoms: this operation must preserve the program behaviour for the rest of atoms in any context. For this reason, the central property under consideration was strong persistence. Other forgetting operators in the literature were defined for different purposes. A recent and exhaustive classification of families of operators in terms of the properties they satisfy can be found in [22]. Our operator $\mathtt{fk}$ yields a propositional formula that is HT-equivalent to the result obtained by *Strong-AS forgetting* $\mathtt{f}_{sas}$ in [23]. Therefore, when both are defined, $\mathtt{fk}$ satisfies the same main properties as those for $\mathtt{f}_{sas}$ classified by [22]. The main difference is that $\mathtt{f}_{sas}$ is only defined for a strict syntactic subclass whereas $\mathtt{fk}(\varphi, V)$ is precisely defined (as a formula) when it is possible to forget atoms $At \setminus V$ in φ.

7.2. Relation to variants of strong equivalence

Regarding strong equivalence, the most relevant related work is surely [12], which contains a characterisation of different types of equivalence of disjunctive logic programs, including the case of (what we have called) PSE. In [12], PSE is characterised in terms of semantic structures called *certificates* which, as expected, have a strong relation to our denotations, as we will see next.

We start by reproducing the definitions of interpretation and model from [12].

Definition 14 *(V-SE-interpretation/model).* Let $V \subseteq At$ be a set of atoms. A *V-SE-interpretation* is a pair $\langle H, T\rangle$ of sets of atoms such that $T \subseteq At$ and either $H = T$ or $H \subset T \cap V$. A *V-SE-interpretation* $\langle H, T\rangle$ is a *V-SE-model* of a formula φ if it satisfies:

i) $T \models \varphi$,
ii) for all $H' \subset T$ with $H' \cap V = T \cap V$, $H' \not\models \varphi^T$, and
iii) if $H \subset T$, there exists $H' \subseteq T$ with $H' \cap V = H$ such that $H' \models \varphi^T$.

The set of all V-SE-models of a formula φ is denoted by $SE^V[\varphi]$. $\square$

where $\models$ above stands for classical implication and φ^T denotes Ferraris' reduct.[8] Since a V-SE-interpretation $\langle H, T\rangle$ satisfies $H \subseteq T$, it can be seen as a particular case of an HT-interpretation. Due to the strong relation between Ferraris' reduct

[8] The original definition in [12] uses the traditional Gelfond and Lifschitz' reduct [5] instead. Although both reducts are not generally equivalent, we can replace one by the other in the current context due to $T \models \varphi$, $H \subseteq T$ and Corollary 1 in [16].

and the logic of HT (Proposition 2), we can easily rephrase the conditions in Definition 14 in terms of HT-satisfaction as follows:

Proposition 20. *Given a set of atoms $V \subseteq At$, a V-SE-interpretation $\langle H, T\rangle$ is a V-SE-model of a formula φ iff*

i) $\langle T, T\rangle \models \varphi$,
ii) *for all $H' \subset T$ with $H' \cap V = T \cap V$, $\langle H', T\rangle \not\models \varphi$, and*
iii) *if $H \subset T$, there exists $H' \subseteq T$ with $H' \cap V = H$ s.t. $\langle H', T\rangle \models \varphi$.* □

And now, we can rephrase this again in terms of T-supports as follows:

Proposition 21. *Given a set of atoms $V \subseteq At$, a V-SE-interpretation $\langle H, T\rangle$ is a V-SE-model of a formula φ iff one of the following hold:*

i) *$H = T$ and $[\![\varphi]\!]^T \neq \emptyset$ ($T \models \varphi$),*
ii) *$H \subset T$, $\emptyset \neq [\![\varphi]\!]^T$ is V-feasible and $H \in [\![\varphi]\!]^T_V$.* □

Certificates from [12] were defined in the following way.

Definition 15 *(certificate, Definition 4 from [12]).* Let $T \subseteq V \subseteq At$ be sets of atoms and $\mathcal{S}$ be a set of V-SE-interpretations. A pair $\langle \mathcal{X}, T\rangle$ with $\mathcal{X} \subseteq 2^{At}$, is a *$V$-projection* of $\mathcal{S}$ iff there exists $T' \subseteq At$ such that

i) $\langle T', T'\rangle \in \mathcal{S}$,
ii) $T = T' \cap V$, and
iii) $\mathcal{X} = \{\, H \mid \langle H, T'\rangle \in \mathcal{S}, H \subset T' \,\}$.

If $V, W \subseteq At$ are two sets of atoms, a (V, W)-*certificate* of a formula φ is a $(V \cup W)$-projection of $SE^V[\varphi]$. □

The next result shows the correspondence between (V, V)-certificates and our V-feasible T-supports.

Proposition 22. *Given sets of atoms $T \subseteq V \subseteq At$ and a set of sets of atoms $\mathcal{X} \subseteq 2^{At}$, the pair $\langle \mathcal{X}, T\rangle$ is a (V, V)-certificate of a formula φ iff there exists some $T' \subseteq At$ such that $T = T' \cap V$, $\emptyset \neq [\![\varphi]\!]^{T'}$ is V-feasible and $[\![\varphi]\!]^{T'}_V \setminus \{ T \} = \mathcal{X}$.*

Definition 16. A (V, W)-certificate $\langle \mathcal{X}, T\rangle$ of a formula φ is said to be *minimal* iff, $\mathcal{Y} \subseteq \mathcal{X}$ implies $\mathcal{Y} = \mathcal{X}$ for any (V, W)-certificate $\langle \mathcal{Y}, T\rangle$.

Proposition 23. *Given sets of atoms $T \subseteq V \subseteq At$ and a set $\mathcal{X} \subseteq 2^{At}$, the pair $\langle \mathcal{X}, T\rangle$ is a minimal (V, V)-certificate of a formula φ iff there exists some $\mathcal{H}$ maximal in $\langle\!\langle \varphi \rangle\!\rangle^T_V$ such that $\mathcal{H} \setminus \{ T \} = \mathcal{X}$.*

Using this result, our Corollary 4 extends Theorem 4 in [12] (for disjunctive logic programs) so that:

Corollary 7. *V-T-denotations characterise projective strong entailment and equivalence of arbitrary propositional formulas, not only disjunctive logic programs. Moreover, this extended characterisation is also satisfied by certificates.* □

Regarding complexity, the connection with [12] plus the results from [13] can be used to prove the following result:

Proposition 24. *Given a set of atoms $V \subseteq At$ and two propositional formulas φ and ψ, deciding whether $\varphi \mathrel{|\!\!\sim_V} \psi$ is Π^P_4-complete.*

Proof. Hardness follows directly from the results in [12]. Regarding membership, we can use the translation in [13] that obtains, for any formula φ, a disjunctive logic program $\pi(\varphi)$ whose size is linear with respect to φ. This program is strongly faithful, that is, it satisfies $\varphi \cong_{At(\varphi)} \pi(\varphi)$. It is easy to check that the computation of $\pi(\varphi)$ in [13] runs in polynomial time. Hence, $\varphi \mathrel{|\!\!\sim_V} \psi$ can be reduced to $\pi(\varphi) \mathrel{|\!\!\sim_V} \pi(\psi)$ in polynomial time. □

Since Proposition 18 provides a polynomial translation from forks in normal form into formulas, we can extrapolate Proposition 24 as follows:

Theorem 6. Checking V-strong entailment and V-strong equivalence of two forks F and G in normal form is a Π^P_4-complete problem.

Although we conjecture a similar complexity result for arbitrary forks, this problem is still under study.[9] To sum up, we can see that certificates correspond to our non-empty T-supports which, in practice, are the ones relevant for checking projective strong entailment and equivalence. Our use of empty T-supports (and their extension to T-views as ideals) is related with our need of providing an algebraic semantics for forks, something that obviously was not among the goals of [12].

8. Conclusions

We have extended the syntax and semantics of Here-and-There (HT) to deal with a new type of construct '|' called *fork*. We have studied the property of *projective strong equivalence* (PSE) for forks: two forks satisfy PSE for a vocabulary V iff they yield the same stable models projected on V for any context over V. We also provided a semantic characterisation of PSE that allowed us to prove that it is always possible to forget (under strong persistence) an auxiliary atom in a fork, something recently proved to be false in standard HT [10].

For future work, we plan to extend these results to other characterisations of equivalence [24] and, in particular, study the case of Projective Uniform Equivalence, that is, PSE for vocabulary V where the context theories are sets of atoms from V. The generalisation of forks to allow arbitrary nesting of connectives is still ongoing work: several alternatives are under consideration, but the properties they should satisfy are unclear yet. Another natural extension of forks is to consider the addition of probabilities. In that way, for instance, our example about Mendelian laws could reflect the proportion of each possible combination, $1/4$ for $\{a\}$ and $\{b\}$ and $1/2$ for $\{a, b\}$. Doing so, we conjecture a strong formal connection to *CP-logic* [25], where the use of disjunction behaves as our fork connective. Note that, although forks do not deal with probabilities, they allow a more general syntax than CP-logic programs, which additionally require the well-founded model to be defined on all atoms. Similarly, we also plan a formal comparison with non-deterministic causal laws [26].

Acknowledgements

We are grateful to the anonymous reviewers of the Artificial Intelligence Journal, and previously, to the reviewers of the workshop ASPOCP'17, for their comments and suggestions that have helped improve the paper substantially. This work has been partially supported by MINECO (grant TIN2017-84453-P) and Xunta de Galicia (grants GPC ED431B 2019/03 and 2016-2019 ED431G/01 for CITIC center), Spain; by the Salvador de Madariaga programme, Spain; by the European Regional Development Fund (ERDF); and by the Centre International de Mathématiques et d'Informatique de Toulouse (CIMI) through contract ANR-11-LABEX-0040-CIMI within the programme ANR-11-IDEX-0002-02.

Declaration of Competing Interest

There are no competing interests.

Appendix A. Proofs of results

Proof of Proposition 4. Since $\mathcal{H} \subseteq \mathcal{H}$ for any $\mathcal{H}$ from $\mathbf{H}_T$, $\preceq$ is clearly reflexive. Take $\mathcal{H}, \mathcal{H}' \in \mathbf{H}_T$ such that $\mathcal{H} \preceq \mathcal{H}'$ and $\mathcal{H}' \preceq \mathcal{H}$. It could be that $\mathcal{H} = \mathcal{H}' = [\ ]$ or $\mathcal{H} \subseteq \mathcal{H}'$ and $\mathcal{H}' \subseteq \mathcal{H}$ which implies $\mathcal{H} = \mathcal{H}'$. Finally, to show that $\preceq$ is transitive, take $\mathcal{H} \preceq \mathcal{H}'$ and $\mathcal{H}' \preceq \mathcal{H}''$. If $\mathcal{H} = [\ ]$, then $\mathcal{H} \preceq \mathcal{H}''$. Otherwise, the three T-supports are non empty and the fact that $\mathcal{H}'' \subseteq \mathcal{H}'$ and $\mathcal{H}' \subseteq \mathcal{H}$ implies that $\mathcal{H} \preceq \mathcal{H}''$.

For the bottom and top elements, since any T-support $\mathcal{H}$ is either empty or $T \in \mathcal{H}$, it is clear that $[\ ] \preceq \mathcal{H}$ and $\mathcal{H} \preceq [\ T\ ]$ for any $\mathcal{H} \in \mathbf{H}_T$. □

Proof of Proposition 5. We prove each property independently as follows:

(i) The "if" direction is trivial. For the "only if" direction, note that $\mathcal{H} \cap \mathcal{H}' = [\ ]$ implies $T \notin \mathcal{H}$ or $T \notin \mathcal{H}'$ which, by definition of T-support, implies that $\mathcal{H} = [\ ]$ or $\mathcal{H}' = [\ ]$ hold.

(ii) If $\mathcal{H} \cap \mathcal{H}' = [\ ]$ the result is trivial. Hence, we assume without loss of generality that $\mathcal{H} \cap \mathcal{H}' \neq [\ ]$. Consequently, it holds that $\mathcal{H} \neq [\ ]$ and $\mathcal{H}' \neq [\ ]$. Since $\mathcal{H}' \preceq \mathcal{H}''$, then $\mathcal{H}'' \neq [\ ]$ and $\mathcal{H}' \supseteq \mathcal{H}''$. This implies that $\mathcal{H} \cap \mathcal{H}' \supseteq \mathcal{H} \cap \mathcal{H}''$ and $\mathcal{H} \cap \mathcal{H}'' \neq [\ ]$, so $\mathcal{H} \cap \mathcal{H}' \preceq \mathcal{H} \cap \mathcal{H}''$.

(iii) Since $\mathcal{H}' \neq [\ ]$, it follows that $\mathcal{H}' \preceq \mathcal{H}''$ implies that $\mathcal{H}' \supseteq \mathcal{H}''$, $\mathcal{H}'' \neq [\ ]$ and also $\mathcal{H} \cup \mathcal{H}' \supseteq \mathcal{H} \cup \mathcal{H}''$. Furthermore, $\mathcal{H}'' \neq [\ ]$ implies $\mathcal{H} \cup \mathcal{H}'' \neq [\ ]$ and, thus, it follows that $\mathcal{H} \cup \mathcal{H}' \preceq \mathcal{H} \cup \mathcal{H}''$.

(iv) If $\mathcal{H} = [\ ]$, the result holds by definition. Otherwise, $\mathcal{H} \preceq \mathcal{H}' \cup \mathcal{H}''$ implies $\mathcal{H} \supseteq \mathcal{H}' \cup \mathcal{H}''$ and $\mathcal{H}' \cup \mathcal{H}'' \neq [\ ]$. In its turn, this implies that $\mathcal{H} \supseteq \mathcal{H}'$ and $\mathcal{H} \supseteq \mathcal{H}''$ and that either $\mathcal{H}' \neq [\ ]$ or $\mathcal{H}'' \neq [\ ]$ hold. Hence, $\mathcal{H} \preceq \mathcal{H}'$ or $\mathcal{H} \preceq \mathcal{H}''$ hold. □

[9] As said before, we are considering a possible polynomial reduction into normal form that includes new auxiliary atoms.

Proof of Proposition 8. In view of Proposition 3 and Proposition 6, we know that $\varphi \mathrel{|\!\sim} \psi$ iff for any $T \subseteq At$, $T \models \varphi$ implies both $T \models \psi$ and also $H \in [\![\varphi]\!]^T$ for any $H \in [\![\psi]\!]^T$. Equivalently, for any $T \subseteq At$, it holds that $[\![\varphi]\!]^T = [\,]$ or $[\![\psi]\!]^T \neq [\,]$ and $[\![\psi]\!]^T \subseteq [\![\varphi]\!]^T$ which means that $[\![\varphi]\!]^T \preceq [\![\psi]\!]^T$. $\square$

Now, in order to prove Proposition 9, we introduce first a group of four lemmas on properties of sets of T-supports.

Lemma 1. *Let Δ be a set of T-supports and $\mathcal{H}$ be a T-support. Then,*

$$^{\downarrow}\{\, \mathcal{H}' \cap \mathcal{H}'' \mid \mathcal{H}' \in \Delta \text{ and } \mathcal{H}'' \in {\downarrow}\mathcal{H} \,\} \;=\; ^{\downarrow}\{\, \mathcal{H}' \cap \mathcal{H} \mid \mathcal{H}' \in \Delta \,\}$$

Proof. First, it is clear that the $\supseteq$ direction holds because $\mathcal{H} \in {\downarrow}\mathcal{H}$. For the $\subseteq$ direction, notice that if $\mathcal{H}' \in \Delta$ and $\mathcal{H}'' \preceq \mathcal{H}$, then $\mathcal{H}' \cap \mathcal{H}'' \preceq \mathcal{H}' \cap \mathcal{H}$, by applying Proposition 5 (ii). $\square$

Lemma 2. *Let $\mathcal{H}$ and $\mathcal{H}'$ be two T-supports. Then,*

$$^{\downarrow}\{\, \mathcal{H}'' \cap \mathcal{H}' \mid \mathcal{H}'' \in {\downarrow}\mathcal{H} \,\} \;=\; ^{\downarrow}\{\mathcal{H}' \cap \mathcal{H}\}$$

Proof. First, it is clear that the $\supseteq$ direction holds because $\mathcal{H} \in {\downarrow}\mathcal{H}$. For the $\subseteq$ direction, notice that if $\mathcal{H}' \in \Delta$ and $\mathcal{H}'' \preceq \mathcal{H}$, then $\mathcal{H}' \cap \mathcal{H}'' \preceq \mathcal{H}' \cap \mathcal{H}$, by applying Proposition 5 (iii). $\square$

Lemma 3. *Let Δ be a set of T-supports and $\mathcal{H}$ a T-support. Then,*

$$^{\downarrow}\{\, \mathcal{H}' \cup \mathcal{H}'' \mid \mathcal{H}' \mathrel{\hat{\in}} \Delta \text{ and } \mathcal{H}'' \mathrel{\hat{\in}} {\downarrow}\mathcal{H} \,\} \;=\; ^{\downarrow}\{\, \mathcal{H}' \cup \mathcal{H} \mid \mathcal{H}' \mathrel{\hat{\in}} \Delta \,\}$$

Proof. First, it is clear that the $\supseteq$ direction holds because $\mathcal{H} \mathrel{\hat{\in}} {\downarrow}\mathcal{H}$. For the $\subseteq$ direction, we have two cases: if $\mathcal{H} = [\,]$, then ${\downarrow}\mathcal{H} = \emptyset$ and $\mathcal{H}'' = [\,]$. Hence, $\mathcal{H}' \cup \mathcal{H} = \mathcal{H}'$ and the statement holds. Otherwise, $\mathcal{H} \neq [\,]$ and, thus, $[\,] \neq \mathcal{H}'' \preceq \mathcal{H}$ implies $\mathcal{H}' \cup \mathcal{H}'' \preceq \mathcal{H}' \cup \mathcal{H}$, by applying Proposition 5 (iii). $\square$

Lemma 4. *Let $\mathcal{H}$ and $\mathcal{H}'$ be two T-supports. Then,*

$$^{\downarrow}\{\, \mathcal{H}'' \cup \mathcal{H}' \mid \mathcal{H}'' \mathrel{\hat{\in}} {\downarrow}\mathcal{H} \,\} \;=\; ^{\downarrow}\{\mathcal{H}' \cup \mathcal{H}\}$$

Proof. First, it is clear that the $\supseteq$ direction holds because $\mathcal{H} \mathrel{\hat{\in}} {\downarrow}\mathcal{H}$. For the $\subseteq$ direction, we have two cases: if $\mathcal{H} = [\,]$, then $\mathcal{H}'' = [\,]$ and the result is trivial. Otherwise, $\mathcal{H} \neq [\,]$ and, thus, $[\,] \neq \mathcal{H}'' \preceq \mathcal{H}$ implies $\mathcal{H}' \cup \mathcal{H}'' \preceq \mathcal{H}' \cup \mathcal{H}$, by applying Proposition 5 (iii). $\square$

Proof of Proposition 9. In case that $\varphi = \bot$ or $\varphi = p \in At$, the statement follows by definition. Otherwise, assume as induction hypothesis that the statement holds for every subformula of φ.

In case that $\varphi = \varphi_1 \wedge \varphi_2$, we can apply induction hypothesis and Lemma 1 and Lemma 2, to deduce that

$$\begin{aligned}
\langle\!\langle \varphi \rangle\!\rangle^T &= {}^{\downarrow}\{\, \mathcal{H} \cap \mathcal{H}' \mid \mathcal{H} \in \langle\!\langle \varphi_1 \rangle\!\rangle^T \text{ and } \mathcal{H}' \in \langle\!\langle \varphi_2 \rangle\!\rangle^T \,\} \\
&= {}^{\downarrow}\{\, \mathcal{H} \cap \mathcal{H}' \mid \mathcal{H} \in {\downarrow}[\![\varphi_1]\!]^T \text{ and } \mathcal{H}' \in {\downarrow}[\![\varphi_2]\!]^T \,\} \\
&= {}^{\downarrow}\{\, [\![\varphi_1]\!]^T \cap [\![\varphi_2]\!]^T \,\} \\
&= {}^{\downarrow}\{\, [\![\varphi_1 \wedge \varphi_2]\!]^T \,\} \\
&= {\downarrow}[\![\varphi_1 \wedge \varphi_2]\!]^T
\end{aligned}$$

Similarly, in case that $\varphi = \varphi_1 \vee \varphi_2$, we can apply induction hypothesis and Lemma 3 and Lemma 4, to deduce that

$$\begin{aligned}
\langle\!\langle \varphi \rangle\!\rangle^T &= {}^{\downarrow}\{\, \mathcal{H} \cup \mathcal{H}' \mid \mathcal{H} \mathrel{\hat{\in}} \langle\!\langle \varphi_1 \rangle\!\rangle^T \text{ and } \mathcal{H}' \mathrel{\hat{\in}} \langle\!\langle \varphi_2 \rangle\!\rangle^T \,\} \\
&= {}^{\downarrow}\{\, \mathcal{H} \cup \mathcal{H}' \mid \mathcal{H} \mathrel{\hat{\in}} {\downarrow}[\![\varphi_1]\!]^T \text{ and } \mathcal{H}' \mathrel{\hat{\in}} {\downarrow}[\![\varphi_2]\!]^T \,\} \\
&= {}^{\downarrow}\{\, [\![\varphi_1]\!]^T \cup [\![\varphi_2]\!]^T \,\} \\
&= {}^{\downarrow}\{\, [\![\varphi_1 \vee \varphi_2]\!]^T \,\} \\
&= {\downarrow}[\![\varphi_1 \vee \varphi_2]\!]^T
\end{aligned}$$

In case that $\varphi = \varphi_1 \to \varphi_2$. Take T such that $[\![\varphi_1]\!]^T = [\,]$, then $T \models \varphi_1 \to \varphi_2$ and:

$$\begin{aligned}
\downarrow[\![\varphi_1 \to \varphi_2]\!]^T &= \downarrow(\overline{[\![\varphi_1]\!]^T} \cup [\![\varphi_2]\!]^T) \\
&= \downarrow(2^T \cup [\![\varphi_2]\!]^T) \\
&= \{2^T\} \\
&= \langle\!\langle \varphi_1 \to \varphi_2 \rangle\!\rangle^T
\end{aligned}$$

Now, suppose that $[\![\varphi_1]\!]^T \neq [\,]$ and $[\![\varphi_2]\!]^T = [\,]$. By induction hypothesis, this implies that $\langle\!\langle \varphi_2 \rangle\!\rangle^T = \emptyset$. Then, since $T \models \varphi_1$, we have that

$$\begin{aligned}
\langle\!\langle \varphi_1 \to \varphi_2 \rangle\!\rangle^T &= {}^{\downarrow}\{\, \overline{[\![\varphi_1]\!]^T} \cup \mathcal{H} \mid \mathcal{H} \in \langle\!\langle \varphi_2 \rangle\!\rangle^T \,\} = \emptyset \\
&= \emptyset
\end{aligned}$$

On the other hand, we also have $\downarrow[\![\varphi_1 \to \varphi_2]\!]^T = \downarrow[\,] = \emptyset$, so the result holds. Finally take T such that $T \models \varphi_1$ and $T \models \varphi_2$. This implies $T \models \varphi_1 \to \varphi_2$ and that any T-support $\mathcal{H}$ verifies that $\mathcal{H} \in \langle\!\langle \varphi_2 \rangle\!\rangle^T = \downarrow[\![\varphi_2]\!]^T$ iff $[\,] \neq \mathcal{H} \preceq [\![\varphi_2]\!]^T$. By applying Proposition 5 (iii), $[\,] \neq \mathcal{H} \preceq [\![\varphi_2]\!]^T$ implies:

$$\overline{[\![\varphi_1]\!]^T} \cup \mathcal{H} \preceq \overline{[\![\varphi_1]\!]^T} \cup [\![\varphi_2]\!]^T = [\![\varphi_1 \to \varphi_2]\!]^T$$

This proves that $\langle\!\langle \varphi_1 \to \varphi_2 \rangle\!\rangle^T = \downarrow[\![\varphi_1 \to \varphi_2]\!]^T$. □

Corollary 8. *Given a set of atoms $T \subseteq At$, a fork F and a propositional formula φ, we have:*

$$\langle\!\langle F \wedge \varphi \rangle\!\rangle^T = {}^{\downarrow}\{\, \mathcal{H} \cap [\![\varphi]\!]^T \mid \mathcal{H} \in \langle\!\langle F \rangle\!\rangle^T \,\}$$

Proof. It is a consequence of Lemma 1 and Proposition 9. □

Proof of Proposition 10. It follows directly from Definition 6 and Definition 7. □

The following result can be proved by structural induction.

Lemma 5. *Let $V, S \subseteq At$ be sets of atoms and take φ a propositional formula such that $At(\varphi) \subseteq V$. Then, for any $H \subseteq S$, it holds that:*

$$\langle H, S\rangle \models \varphi \text{ is equivalent to } \langle H \cap V, S \cap V\rangle \models \varphi$$

Definition 17. Let $T, V \subseteq At$ be two sets of atoms. We say that a T-support $\mathcal{H}$ is V*-respectful*, if for any $H, H' \subseteq T$ such that $H \cap V = H' \cap V$, we have that $H \in \mathcal{H}$ iff $H' \in \mathcal{H}$.

Corollary 9. *Let $V, S \subseteq At$ be sets of atoms and take φ a propositional formula such that $At(\varphi) \subseteq V$. Then $[\![\varphi]\!]^S$ is V*-respectful.

Proof. This result is a consequence of Lemma 5 and Proposition 6. □

Lemma 6. *Let $V, S \subseteq At$ be sets of atoms and take L a fork such that $At(L) \subseteq V$. Then any $\mathcal{H}$ maximal in $\langle\!\langle L \rangle\!\rangle^S$ is V*-respectful.

Proof. If $L = \varphi$ with φ a propositional formula and $\mathcal{H}$ is maximal in $\langle\!\langle \varphi \rangle\!\rangle^S = \downarrow[\![\varphi]\!]^S$ then $\mathcal{H} = [\![\varphi]\!]^S$ and the result is just Corollary 9.
If $L = L_1 \wedge L_2$ and $\mathcal{H}$ is maximal in $\langle\!\langle L \rangle\!\rangle^S$, then we can suppose that $\mathcal{H} = \mathcal{H}_1 \cap \mathcal{H}_2$ with $\mathcal{H}_i$ maximal in $\langle\!\langle L_i \rangle\!\rangle^S$. Since, by induction each $\mathcal{H}_i$ is V-respectful, we know that $\mathcal{H}$ is also V-respectful.
Let $L = \varphi \to L_1$ with φ a propositional formula and L_1 a fork such that $At(\varphi), At(L_1) \subseteq V$. If $\mathcal{H}$ is maximal in $\langle\!\langle L \rangle\!\rangle^S$, we can assume, without loss of generality, that $\mathcal{H} = \overline{[\![\varphi]\!]^S} \cup \mathcal{H}'$ with $\mathcal{H}'$ maximal in $\langle\!\langle L_1 \rangle\!\rangle^S$. Suppose that $H, H' \subseteq S$ satisfy that $H \cap V = H' \cap V$ and $H \in \mathcal{H}$. Then $H \in \overline{[\![\varphi]\!]^S}$ or $H \in \mathcal{H}'$. By applying Corollary 9 and induction, we deduce that $H' \in \overline{[\![\varphi]\!]^S} \cup \mathcal{H}' = \mathcal{H}$.
Finally suppose that $L = (L_1 \mid L_2)$ and $\mathcal{H}$ is maximal in $\langle\!\langle L \rangle\!\rangle^S = \langle\!\langle L_1 \rangle\!\rangle^S \cup \langle\!\langle L_2 \rangle\!\rangle^S$. Again, applying induction, we can deduce that $\mathcal{H}$ is V-respectful. □

Lemma 7. *For any set of atoms $T \subseteq V \subseteq At$ and any T-support $\mathcal{H}$, there is a propositional formula φ with $At(\varphi) \subseteq V$ such that $[\![\varphi]\!]^T = \mathcal{H}$ and $[\![\varphi]\!]^Y = [\,]$ for every set of atoms $Y \subseteq V$ with $Y \neq T$.*

Proof. If $\mathcal{H}$ is a T-support, we can define an assignment $\sigma_{\mathcal{H}}$ by $\sigma_{\mathcal{H}}(T) = \mathcal{H}$ and $\sigma_{\mathcal{H}}(Y) = [\,]$ for any $Y \subseteq V$ with $Y \neq T$. Taking into account Proposition 7 with signature V, we know that the propositional formula Φ_S with $S = \{\, \langle H, T\rangle \mid H \in \mathcal{H} \,\}$ satisfies the required properties. □

Lemma 8. *For any set of atoms $T \subseteq V \subseteq At$ and any non-empty T-support $\mathcal{H} \neq [\,]$, there is a propositional formula φ that satisfies $At(\varphi) \subseteq V$ and:*

i) $\mathcal{H} \cap [\![\varphi]\!]^T = [\,T\,]$,
ii) $\mathcal{H}' \cap [\![\varphi]\!]^T \neq [\,T\,]$ *for any T-support $\mathcal{H}'$ such that $\mathcal{H} \not\preceq \mathcal{H}'$.*

Proof. Since $\overline{\mathcal{H}}$ is a T-support, from Lemma 7, there is a propositional formula φ such that $[\![\varphi]\!]^T = \overline{\mathcal{H}}$ and $At(\varphi) \subseteq V$. Furthermore, it follows that $\mathcal{H} \cap [\![\varphi]\!]^T = \mathcal{H} \cap \overline{\mathcal{H}} = [\,T\,]$. On the other hand, suppose that there is some T-support $\mathcal{H}'$ s.t. $\mathcal{H} \not\preceq \mathcal{H}'$ and

$$\mathcal{H}' \cap [\![\varphi]\!]^T = \mathcal{H}' \cap \overline{\mathcal{H}} = [\,T\,]$$

Without loss of generality, we can assume that $\mathcal{H}' \neq [\,]$. Then, $\mathcal{H} \not\preceq \mathcal{H}'$ and $\mathcal{H} \neq [\,]$ implies $\mathcal{H}' \not\subseteq \mathcal{H}$ and, thus, there is $H \in \mathcal{H}' \backslash \mathcal{H}$. Since $H \notin \mathcal{H}$ and $T \in \mathcal{H}$, it follows that $H \neq T$ and $H \in \overline{\mathcal{H}}$. This implies $H \in \mathcal{H}' \cap \overline{\mathcal{H}}$ and, thus, it follows that $H \in \mathcal{H}' \cap [\![\varphi]\!]^T$ which contradicts the assumption that $\mathcal{H}' \cap [\![\varphi]\!]^T = [\,T\,]$. □

Proof of Proposition 11. Suppose that $F \mathrel{|\!\!\sim} G$ and take $\mathcal{H} \in \langle\!\langle F \rangle\!\rangle^T$. If we apply Lemma 8, we know we can find φ such that $\mathcal{H} \cap [\![\varphi]\!]^T = [\,T\,]$ and $\mathcal{H}' \cap [\![\varphi]\!]^T \neq [\,T\,]$ for any $\mathcal{H}'$ such that $\mathcal{H} \not\preceq \mathcal{H}'$. Taking into account Corollary 8, it is clear that $[\,T\,] \in \langle\!\langle F \wedge \varphi \rangle\!\rangle^T$, so $T \in SM[F \wedge \varphi] \subseteq SM[G \wedge \varphi]$ which implies that $[\,T\,] \in \langle\!\langle G \wedge \varphi \rangle\!\rangle^T$. This means by applying Corollary 8 that there exists $\mathcal{H}' \in \langle\!\langle G \rangle\!\rangle^T$ such that $[\,T\,] = \mathcal{H}' \cap [\![\varphi]\!]^T$. Finally, we necessarily have that $\mathcal{H} \preceq \mathcal{H}'$ and $\mathcal{H} \in \langle\!\langle G \rangle\!\rangle^T$.
For the other direction, take L a fork with $T \in SM[F \wedge L]$. Since $[\,T\,] \in \langle\!\langle F \wedge L \rangle\!\rangle^T$, we know, by definition that $[\,T\,] = \mathcal{H} \cap \mathcal{H}'$ with $\mathcal{H} \in \langle\!\langle F \rangle\!\rangle^T$ and $\mathcal{H}' \in \langle\!\langle L \rangle\!\rangle^T$. By hypothesis, $\mathcal{H} \in \langle\!\langle G \rangle\!\rangle^T$, so we have that $[\,T\,] \in \langle\!\langle G \wedge L \rangle\!\rangle^T$ or equivalently $T \in SM[G \wedge L]$. □

Proof of Proposition 12. The proof of (9), (10) and (16) is straightforward. Taking into account Proposition 11 and the fact that $\langle\!\langle F \mid G \rangle\!\rangle^T = \langle\!\langle F \rangle\!\rangle^T \cup \langle\!\langle G \rangle\!\rangle^T$, we obtain (11).
For the proof of (12), notice that:

$$\begin{aligned}
\langle\!\langle (F \mid G) \wedge L \rangle\!\rangle^T &= {}^{\downarrow}\{\, \mathcal{H} \cap \mathcal{H}' \mid \mathcal{H} \in \langle\!\langle F \mid G \rangle\!\rangle^T \text{ and } \mathcal{H}' \in \langle\!\langle L \rangle\!\rangle^T \,\} \\
&= {}^{\downarrow}\{\, \mathcal{H} \cap \mathcal{H}' \mid \mathcal{H} \in \langle\!\langle F \rangle\!\rangle^T \cup \langle\!\langle G \rangle\!\rangle^T \text{ and } \mathcal{H}' \in \langle\!\langle L \rangle\!\rangle^T \,\} \\
&= {}^{\downarrow}(\{\, \mathcal{H} \cap \mathcal{H}' \mid \mathcal{H} \in \langle\!\langle F \rangle\!\rangle^T \text{ and } \mathcal{H}' \in \langle\!\langle L \rangle\!\rangle^T \,\} \\
&\quad \cup \ \{\, \mathcal{H} \cap \mathcal{H}' \mid \mathcal{H} \in \langle\!\langle G \rangle\!\rangle^T \text{ and } \mathcal{H}' \in \langle\!\langle L \rangle\!\rangle^T \,\}) \\
&= {}^{\downarrow}\{\, \mathcal{H} \cap \mathcal{H}' \mid \mathcal{H} \in \langle\!\langle F \rangle\!\rangle^T \text{ and } \mathcal{H}' \in \langle\!\langle L \rangle\!\rangle^T \,\} \\
&\quad \cup \ {}^{\downarrow}\{\, \mathcal{H} \cap \mathcal{H}' \mid \mathcal{H} \in \langle\!\langle G \rangle\!\rangle^T \text{ and } \mathcal{H}' \in \langle\!\langle L \rangle\!\rangle^T \,\} \\
&= \langle\!\langle F \wedge L \rangle\!\rangle^T \cup \langle\!\langle G \wedge L \rangle\!\rangle^T \\
&= \langle\!\langle (F \wedge L) \mid (G \wedge L) \rangle\!\rangle^T
\end{aligned}$$

In order to prove (13), assume first that $T \not\models \varphi$. Then, it follows that $\langle\!\langle \varphi \to (F \mid G) \rangle\!\rangle^T = \langle\!\langle \varphi \to F \rangle\!\rangle^T = \langle\!\langle \varphi \to G \rangle\!\rangle^T = \{2^T\}$. Since $\langle\!\langle (\varphi \to F) \mid (\varphi \to G) \rangle\!\rangle^T = \langle\!\langle \varphi \to F \rangle\!\rangle^T \cup \langle\!\langle \varphi \to G \rangle\!\rangle^T$, we finally conclude

$$\langle\!\langle \varphi \to (F \mid G) \rangle\!\rangle^T = 2^T = \langle\!\langle (\varphi \to F) \mid (\varphi \to G) \rangle\!\rangle^T$$

Otherwise,

$$\begin{aligned}
\langle\!\langle \varphi \to (F \mid G) \rangle\!\rangle^T &= {}^{\downarrow}\{\, \overline{[\![\varphi]\!]^T} \cup \mathcal{H} \mid \mathcal{H} \in \langle\!\langle F \rangle\!\rangle^T \cup \langle\!\langle G \rangle\!\rangle^T \,\} \\
&= {}^{\downarrow}(\{\, \overline{[\![\varphi]\!]^T} \cup \mathcal{H} \mid \mathcal{H} \in \langle\!\langle F \rangle\!\rangle^T \,\} \\
&\quad \cup \{\, \overline{[\![\varphi]\!]^T} \cup \mathcal{H} \mid \mathcal{H} \in \langle\!\langle G \rangle\!\rangle^T \,\}) \\
&= {}^{\downarrow}\{\, \overline{[\![\varphi]\!]^T} \cup \mathcal{H} \mid \mathcal{H} \in \langle\!\langle F \rangle\!\rangle^T \,\} \\
&\quad \cup {}^{\downarrow}\{\, \overline{[\![\varphi]\!]^T} \cup \mathcal{H} \mid \mathcal{H} \in \langle\!\langle G \rangle\!\rangle^T \,\} \\
&= \langle\!\langle (\varphi \to F) \mid (\varphi \to G) \rangle\!\rangle^T
\end{aligned}$$

Now we are going to prove (14). First suppose that $T \not\models \varphi$. In this case,

$$\langle\!\langle \varphi \to (F \wedge G) \rangle\!\rangle^T = \langle\!\langle \varphi \to F \rangle\!\rangle^T = \langle\!\langle \varphi \to G \rangle\!\rangle^T = \{2^T\}$$

so $\langle\!\langle \varphi \to (F \wedge G) \rangle\!\rangle^T = \langle\!\langle (\varphi \to F) \mid (\varphi \to G) \rangle\!\rangle^T = \{2^T\}$. Otherwise, it follows that:

$$\begin{aligned}
\langle\!\langle\, \varphi \rightarrow (F \wedge G) \,\rangle\!\rangle^T &= {}^{\downarrow}\{\, \overline{[\![\, \varphi \,]\!]^T} \cup \mathcal{H} \,\big|\, \mathcal{H} \in \langle\!\langle\, F \wedge G \,\rangle\!\rangle^T \,\} \\
&= {}^{\downarrow}(\{\, \overline{[\![\, \varphi \,]\!]^T} \cup (\mathcal{H}_1 \cap \mathcal{H}_2) \,\big|\, \mathcal{H}_1 \in \langle\!\langle\, F \,\rangle\!\rangle^T ,\ \mathcal{H}_2 \in \langle\!\langle\, G \,\rangle\!\rangle^T \,\} \\
&= {}^{\downarrow}\{\, (\overline{[\![\, \varphi \,]\!]^T} \cup \mathcal{H}_1) \cap (\overline{[\![\, \varphi \,]\!]^T} \cup \mathcal{H}_2) \,\big|\, \mathcal{H}_1 \in \langle\!\langle\, F \,\rangle\!\rangle^T ,\ \mathcal{H}_2 \in \langle\!\langle\, G \,\rangle\!\rangle^T \,\} \\
&= \langle\!\langle\, (\varphi \rightarrow F) \wedge (\varphi \rightarrow G) \,\rangle\!\rangle^T
\end{aligned}$$

Take φ and ψ two formulas and F a fork. In order to show (15), let us start supposing that $T \not\models \varphi$ or $T \not\models \psi$. Then, it holds that:

$$\langle\!\langle\, \varphi \rightarrow (\psi \rightarrow F) \,\rangle\!\rangle^T = \langle\!\langle\, \varphi \wedge \psi \rightarrow F \,\rangle\!\rangle^T = \{2^T\}$$

This implies that we can take T such that $T \models \varphi \wedge \psi$ and:

$$\begin{aligned}
\langle\!\langle\, \varphi \rightarrow (\psi \rightarrow F) \,\rangle\!\rangle^T &= {}^{\downarrow}\{\, \overline{[\![\, \varphi \,]\!]^T} \cup \mathcal{H} \,\big|\, \mathcal{H} \in \langle\!\langle\, \psi \rightarrow F \,\rangle\!\rangle^T \,\} \\
&= {}^{\downarrow}\{\, \overline{[\![\, \varphi \,]\!]^T} \cup (\overline{[\![\, \psi \,]\!]^T} \cup \mathcal{H}') \,\big|\, \mathcal{H}' \in \langle\!\langle\, F \,\rangle\!\rangle^T \,\} \\
&= {}^{\downarrow}\{\, \overline{[\![\, \varphi \wedge \psi \,]\!]^T} \cup \mathcal{H}' \,\big|\, \mathcal{H}' \in \langle\!\langle\, F \,\rangle\!\rangle^T \,\} \\
&= \langle\!\langle\, \varphi \wedge \psi \rightarrow F \,\rangle\!\rangle^T
\end{aligned}$$

Finally, to check (17), notice that $\langle\!\langle\, \neg\varphi \mid \neg\neg\varphi \,\rangle\!\rangle^T = \langle\!\langle\, \neg\varphi \,\rangle\!\rangle^T \cup \langle\!\langle\, \neg\neg\varphi \,\rangle\!\rangle^T$ for any $T \subseteq At$. Besides, if $T \models \varphi$, then $\langle\!\langle\, \neg\varphi \,\rangle\!\rangle^T = \emptyset$ and $\langle\!\langle\, \neg\neg\varphi \,\rangle\!\rangle^T = \{2^T\}$ and, if $T \not\models \varphi$, then $\langle\!\langle\, \neg\varphi \,\rangle\!\rangle^T = \{2^T\}$ and $\langle\!\langle\, \neg\neg\varphi \,\rangle\!\rangle^T = \emptyset$. $\square$

Lemma 9. *Given set of atoms $T \subseteq V \subseteq At$ and a T-support $\mathcal{H}$, then $\mathcal{H}$ is V-feasible and $\mathcal{H} = \mathcal{H}_V$.*

Proof. Notice that, for any $H \subseteq T \subseteq V$, then $H = H \cap V$. $\square$

Lemma 10. *Given sets of atoms $V, T \subseteq At$, a pair of T-supports $\mathcal{H}, \mathcal{H}'$ such that $[\ ] \neq \mathcal{H} \preceq \mathcal{H}'$ and $\mathcal{H}$ is V-feasible, then $\mathcal{H}'$ is V-feasible too.*

Proof. First, note that $[\ ] \neq \mathcal{H} \preceq \mathcal{H}'$ implies $\mathcal{H} \supseteq \mathcal{H}'$. Suppose, for the sake of contradiction, that $\mathcal{H}$ is V-unfeasible. Hence, there is $H \in \mathcal{H}'$ such that $H \subset T$ and $H \cap V = T \cap V$. Since $\mathcal{H} \supseteq \mathcal{H}'$, this implies that $H \in \mathcal{H}$ which is a contradiction with the assumption that $\mathcal{H}$ is V-feasible. $\square$

Lemma 11. *Let $T, V \subseteq At$ be two sets of atoms and $\mathcal{H}, \mathcal{H}'$ be a pair of T-supports such that $\mathcal{H} \preceq \mathcal{H}'$. Then, $\mathcal{H}_V \preceq \mathcal{H}'_V$.*

Proof. If $\mathcal{H} = [\ ]$, then $\mathcal{H}_V = [\ ] \preceq \mathcal{H}'_V$. Hence, we assume without loss of generality that $\mathcal{H} \neq [\ ]$ and, thus, $\mathcal{H} \preceq \mathcal{H}'$ implies $\mathcal{H} \supseteq \mathcal{H}'$ and $\mathcal{H}' \neq [\ ]$. Then, by definition, it follows that

$$\mathcal{H}_V \;=\; \{\, H \cap V \,\big|\, H \in \mathcal{H} \,\} \;\supseteq\; \{\, H \cap V \,\big|\, H \in \mathcal{H}' \,\} \;=\; \mathcal{H}'_V$$

Furthermore, $\mathcal{H}' \neq [\ ]$ implies $\mathcal{H}'_V \neq [\ ]$ and, thus, $\mathcal{H}_V \preceq \mathcal{H}'_V$ holds. $\square$

Lemma 12. *Given sets of atoms $V, T \subseteq At$, a pair of T-supports $\mathcal{H}, \mathcal{H}'$ such that $\mathcal{H} \cap \mathcal{H}'$ is V-feasible, $\mathcal{H}' \neq [\ ]$ and $\mathcal{H}'$ is V-respectful, then $\mathcal{H}$ is V-feasible.*

Proof. Suppose, for the sake of contradiction, that $\mathcal{H}$ is V-unfeasible. Hence, there is $H \in \mathcal{H}$ such that $H \subset T$ and $H \cap V = T \cap V$. Furthermore, since $\mathcal{H}' \neq [\ ]$, it follows that $T \in \mathcal{H}'$ and, since $\mathcal{H}'$ is V-respectful, the above implies that $H \in \mathcal{H}'$. Then, $H \in \mathcal{H} \cap \mathcal{H}'$ which is a contradiction with the assumption that $\mathcal{H} \cap \mathcal{H}'$ is V-feasible. $\square$

Lemma 13. *Let $T, V \subseteq At$ be two sets of atoms and $\mathcal{H}, \mathcal{H}'$ be a pair of T-supports. Then:*

i) $(\mathcal{H} \cap \mathcal{H}')_V \subseteq \mathcal{H}_V \cap \mathcal{H}'_V$,
ii) *In addition, if $\mathcal{H}$ is V-*respectful*, then*

$$(\mathcal{H} \cap \mathcal{H}')_V = \mathcal{H}_V \cap \mathcal{H}'_V.$$

Proof. By definition, it follows that

$$\begin{aligned}
(\mathcal{H} \cap \mathcal{H}')_V &= \{\, H' \,\big|\, \exists H \text{ s.t. } H' = H \cap V \text{ and } H \in \mathcal{H} \cap \mathcal{H}' \,\} \\
&= \{\, H' \,\big|\, \exists H \text{ s.t. } H' = H \cap V \text{ and } H \in \mathcal{H} \text{ and } H \in \mathcal{H}' \,\} \\
&\subseteq \mathcal{H}_V \cap \mathcal{H}'_V
\end{aligned}$$

Notice that, in general, the equality does not hold. Take $At = T = \{p, q\}$, $V = \{q\}$, $\mathcal{H} = [\emptyset, T]$ and $\mathcal{H}' = [\{p\}, T]$. Then $(\mathcal{H} \cap \mathcal{H}')_V = [T]_V = [\{q\}]$ and $\mathcal{H}_V \cap \mathcal{H}'_V = [\emptyset, \{q\}] \cap [\emptyset, \{q\}] = [\emptyset, \{q\}]$. Now, suppose that $\mathcal{H}$ is V-respectful. If $H \in \mathcal{H}_V \cap \mathcal{H}'_V$, then there exist $H_1 \in \mathcal{H}$ and $H_2 \in \mathcal{H}'$ such that $H = H_1 \cap V = H_2 \cap V$. But then, $H_2 \in \mathcal{H}$ and finally $H \in (\mathcal{H} \cap \mathcal{H}')_V$. □

Lemma 14. *Let $T, V \subseteq At$ be two sets of atoms and $\mathcal{H}, \mathcal{H}'$ be a pair of T-supports. Then, $(\mathcal{H} \cup \mathcal{H}')_V = \mathcal{H}_V \cup \mathcal{H}'_V$.*

Proof. By definition, it follows that

$$\begin{aligned}
(\mathcal{H} \cup \mathcal{H}')_V &= \{\, H' \mid \exists H \text{ s.t. } H' = H \cap V \text{ and } H \in \mathcal{H} \cup \mathcal{H}' \,\} \\
&= \{\, H' \mid \exists H \text{ s.t. } H' = H \cap V \text{ and either } H \in \mathcal{H} \text{ or } H \in \mathcal{H}') \,\} \\
&= \{\, H' \mid \exists H \text{ s.t. } H' = H \cap V \text{ and either } H' \in \mathcal{H}_V \text{ or } H' \in \mathcal{H}'_V) \,\} \\
&= \mathcal{H}_V \cup \mathcal{H}'_V \qquad \square
\end{aligned}$$

Lemma 15. *Let $S, V \subseteq At$ be two sets of atoms and $\mathcal{H}$ be some S-support. Then, we have $(\overline{\mathcal{H}})_V \supseteq \overline{\mathcal{H}_V}$. Moreover, when $\mathcal{H}$ is V-respectful, then $(\overline{\mathcal{H}})_V = \overline{\mathcal{H}_V}$.*

Proof. Let $T \stackrel{\text{def}}{=} S \cap V$. In case that $\mathcal{H} = 2^S$, it follows that

$$(\overline{\mathcal{H}})_V \;=\; [\,]_V \;=\; [\,] \;=\; \overline{2^T} \;=\; \overline{\mathcal{H}_V}$$

Otherwise,

$$\begin{aligned}
(\overline{\mathcal{H}})_V &= ([\,S\,] \cup \{H \subseteq S \mid H \notin \mathcal{H}\})_V \\
&= ([\,S \cap V\,] \cup \{H \cap V \mid H \subseteq S \text{ and } H \notin \mathcal{H}\}) \\
&= [\,T\,] \cup \{H \cap V \mid H \subseteq S \text{ and } H \notin \mathcal{H}\}
\end{aligned}$$

Furthermore, we have that $H \in \mathcal{H}$ implies $H \cap V \in \mathcal{H}_V$ or equivalently that $H \cap V \notin \mathcal{H}_V$ implies $H \notin \mathcal{H}$. Hence, we get

$$\begin{aligned}
\{H \cap V \mid H \subseteq S \text{ and } H \notin \mathcal{H}\} &\supseteq \{H \cap V \mid H \subseteq S \text{ and } H \cap V \notin \mathcal{H}_V\} \\
&\supseteq \{H' = H \cap V \subseteq T \mid H \subseteq S \text{ and } H' \notin \mathcal{H}_V\} \\
&= \overline{\mathcal{H}_V}
\end{aligned}$$

Note that, equality does not hold: Let $S = At = \{a, b\}$, $V = \{a\}$ and $\mathcal{H} = [\{a, b\}, \{b\}, \emptyset]$. Then, $\overline{\mathcal{H}} = [\{a, b\}, \{a\}]$ and $\overline{\mathcal{H}}_V = [\{a\}]$. On the other hand, we have $\mathcal{H}_V = [\{a\}, \emptyset]$ and $\overline{\mathcal{H}_V} = [\,]$. Clearly, $[\{a\}] = [\,]$. Furthermore, we also have $[\{a\}] \not\subseteq [\,]$. When $\mathcal{H}$ is V-respectful, and $H = H' \cap V \in (\overline{\mathcal{H}})_V$ with $H' \in \overline{\mathcal{H}}$, we have that $H' \cap V \in \overline{\mathcal{H}_V}$. Otherwise $H' \cap V = H'' \cap V$ with $H'' \in \mathcal{H}$ which would imply that $H' \in \mathcal{H}$. This completes the proof. □

The following group of auxiliary results will be used to prove properties about projective strong entailment and equivalence.

Lemma 16. *Given sets of atoms $T \subseteq V \subseteq At$ and a fork F, we have $\langle\!\langle F \rangle\!\rangle^T \subseteq \langle\!\langle F \rangle\!\rangle^T_V$.*

Proof. Since $T = T \cap V$ and applying Lemma 9, it follows that any $\mathcal{H} \in \langle\!\langle F \rangle\!\rangle^T$ is V-feasible and $\mathcal{H} = \mathcal{H}_V$. So $\mathcal{H} \in \langle\!\langle F \rangle\!\rangle^T_V$. □

Corollary 10. *Given sets of atoms $T \subseteq V \subseteq At$ and $S \subseteq At$ with $T = S \cap V$ and given a propositional formula φ such that $At(\varphi) \subseteq V$, it holds:*

1. $[\![\varphi]\!]^T = ([\![\varphi]\!]^S)_V$,
2. $\overline{[\![\varphi]\!]^T} = \overline{([\![\varphi]\!]^S)_V} = (\overline{[\![\varphi]\!]^S})_V$.

Proof. It is a consequence of Lemma 5, Lemma 15 and Corollary 9. □

Proof of Proposition 13. First of all, if we take into account Lemma 16, we only have to prove that $\langle\!\langle F \rangle\!\rangle^T_V \subseteq \langle\!\langle F \rangle\!\rangle^T$. We proceed by structural induction.

- $F=\varphi$ is a propositional formula. Take $\mathcal{H} \in \langle\!\langle \varphi \rangle\!\rangle^T_V$, then there exist $S \subseteq At$ and $\mathcal{H}' \in \langle\!\langle \varphi \rangle\!\rangle^S = \downarrow [\![\varphi]\!]^S$ with $S \cap V = T$ and $\mathcal{H} \preceq \mathcal{H}'_V$. This implies $\mathcal{H} \preceq \mathcal{H}'_V \preceq ([\![\varphi]\!]^S)_V = [\![\varphi]\!]^T$ by applying Lemma 11 and Corollary 10. Finally $\mathcal{H} \in \downarrow [\![\varphi]\!]^T = \langle\!\langle \varphi \rangle\!\rangle^T$ by Proposition 9.
- $F = F_1 \wedge F_2$
  Take $\mathcal{H} \in \langle\!\langle F \rangle\!\rangle^T_V$, then there exist $S \subseteq At$ and $\mathcal{H}' \in \langle\!\langle F \rangle\!\rangle^S$ with $S \cap V = T$ and $\mathcal{H} \preceq \mathcal{H}'_V$. By definition, we can find $\mathcal{H}_i$ maximal in $\langle\!\langle F_i \rangle\!\rangle^S$ for $i=1,2$ such that $\mathcal{H}' \preceq \mathcal{H}_1 \cap \mathcal{H}_2$. So, we have that $\mathcal{H} \preceq \mathcal{H}'_V \preceq (\mathcal{H}_1)_V \cap (\mathcal{H}_2)_V$ by applying Lemma 11, Lemma 13 and Lemma 6. Now, we can apply induction hypothesis to show that $(\mathcal{H}_i)_V \in \langle\!\langle F_i \rangle\!\rangle^T_V = \langle\!\langle F_i \rangle\!\rangle^T$ for $i=1,2$ and finally $\mathcal{H} \in \langle\!\langle F_1 \wedge F_2 \rangle\!\rangle^T = \langle\!\langle F \rangle\!\rangle^T$.
- $F = \varphi \to G$
  Take $\mathcal{H} \in \langle\!\langle F \rangle\!\rangle^T_V$, then there exist $S \subseteq At$ and $\mathcal{H}' \in \langle\!\langle F \rangle\!\rangle^S$ with $S \cap V = T$ and $\mathcal{H} \preceq \mathcal{H}'_V$. We can assume that $S \models \varphi$ (so $T \models \varphi$) and $\langle\!\langle G \rangle\!\rangle^S \neq \emptyset$. Otherwise, $\langle\!\langle F \rangle\!\rangle^S = \emptyset$.
  Let $\mathcal{H}'' \in \langle\!\langle G \rangle\!\rangle^S$ such that $\mathcal{H}' \preceq \overline{[\![\varphi]\!]^S} \cup \mathcal{H}''$. Then by applying Lemma 11, Lemma 14 and Corollary 10, it follows that $\mathcal{H} \preceq \mathcal{H}'_V \preceq \overline{[\![\varphi]\!]^T} \cup \mathcal{H}''_V$ with $\mathcal{H}''_V \in \langle\!\langle G \rangle\!\rangle^T_V \subseteq \langle\!\langle G \rangle\!\rangle^T$ and finally $\mathcal{H} \in \langle\!\langle \varphi \to G \rangle\!\rangle^T = \langle\!\langle F \rangle\!\rangle^T$.
- $F = (G \mid L)$
  In this case, if $\mathcal{H} \in \langle\!\langle F \rangle\!\rangle^T_V$, then there exist $S \subseteq At$ and $\mathcal{H}' \in \langle\!\langle G \rangle\!\rangle^S \cup \langle\!\langle L \rangle\!\rangle^S$ with $S \cap V = T$ and $\mathcal{H} \preceq \mathcal{H}'_V$. Then, by induction hypothesis, we have that $\mathcal{H}'_V \in \langle\!\langle G \rangle\!\rangle^T \cup \langle\!\langle L \rangle\!\rangle^T$ and $\mathcal{H} \in \langle\!\langle F \rangle\!\rangle^T$. □

Proof of Proposition 14. First of all, if $T \in SM_V[F]$, there exists $S \subseteq At$ such that $T = S \cap V$ and $[\ S\ ] \in \langle\!\langle F \rangle\!\rangle^S$. This implies $[\ T\ ] = [\ S\ ]_V \in \langle\!\langle F \rangle\!\rangle^T_V$. Conversely, if $[\ T\ ] \in \langle\!\langle F \rangle\!\rangle^T_V$, we know that $[\ T\ ] \preceq \mathcal{H}_V$ for some $\mathcal{H} \in \langle\!\langle F \rangle\!\rangle^{T'}$ that is V-feasible with $T = T' \cap V$. So, $\mathcal{H}_V = [\ T\ ]$ and $\mathcal{H} = [\ T'\ ] \in \langle\!\langle F \rangle\!\rangle^{T'}$ which means that $T' \in SM(F)$ and finally $T \in SM_V[F]$. □

Lemma 17. *Let $V, S \subseteq At$ be sets of atoms and take F a fork such that $At(F) \subseteq V$. Then, for any $\preceq$-maximal S-support $\mathcal{H} \in \langle\!\langle F \rangle\!\rangle^S$, it holds that $\mathcal{H}_V \in \langle\!\langle F \rangle\!\rangle^T$ with $T = S \cap V$.*

Proof. In case that F is a propositional formula, we have that $\langle\!\langle F \rangle\!\rangle^S = \downarrow [\![F]\!]^S$ and, thus, we have that $\mathcal{H} = [\![F]\!]^S$. Hence, $\mathcal{H}_V = ([\![F]\!]^S)_V = [\![F]\!]^T$ by applying Corollary 10. Otherwise, we assume as induction hypothesis that the lemma statement holds for every subfork of F.

In case that $F = F_1 \wedge F_2$. Then, $[\ ] \neq \mathcal{H} = \mathcal{H}_1 \cap \mathcal{H}_2$ with $\mathcal{H}_1 \in \langle\!\langle F_1 \rangle\!\rangle^S$ and $\mathcal{H}_2 \in \langle\!\langle F_2 \rangle\!\rangle^S$. Furthermore, let us assume without loss of generality that each $\mathcal{H}_i$ is $\preceq$-maximal in $\langle\!\langle F_i \rangle\!\rangle^S$. Notice that each $\mathcal{H}_i$ is V-respectful in view of Lemma 6. Then, $\mathcal{H}_V = (\mathcal{H}_1 \cap \mathcal{H}_2)_V = (\mathcal{H}_1)_V \cap (\mathcal{H}_2)_V$ by Lemma 13. By induction hypothesis, we have that $(\mathcal{H}_i)_V \in \langle\!\langle F_i \rangle\!\rangle^T$ and $\mathcal{H}_V \in \langle\!\langle F \rangle\!\rangle^T$.

In case that $F = \varphi \to G$. First of all, notice that $S \models \varphi$ iff $T \models \varphi$ by Lemma 5. In case that $S \not\models \varphi$, $\mathcal{H} = 2^S$ and $\mathcal{H}_V = 2^T \in \langle\!\langle F \rangle\!\rangle^T$. Otherwise $\langle\!\langle F \rangle\!\rangle^S = {\downarrow}\{\ \overline{[\![\varphi]\!]^S} \cup \mathcal{H}' \mid \mathcal{H}' \in \langle\!\langle G \rangle\!\rangle^S\ \}$. That is, $\mathcal{H} = \overline{[\![\varphi]\!]^S} \cup \mathcal{H}'$ for some $\mathcal{H}' \in \langle\!\langle G \rangle\!\rangle^S$. Assume that $\mathcal{H}'$ is $\preceq$-maximal in $\langle\!\langle G \rangle\!\rangle^S$, hence by Lemma 14 and Corollary 10.

$$
\begin{aligned}
\mathcal{H}_V &= (\overline{[\![\varphi]\!]^S} \cup \mathcal{H}')_V \\
&= (\overline{[\![\varphi]\!]^S})_V \cup \mathcal{H}'_V \\
&= \overline{[\![\varphi]\!]^T} \cup \mathcal{H}'_V
\end{aligned}
$$

Since, by induction hypothesis, we have that $\mathcal{H}'_V \in \langle\!\langle G \rangle\!\rangle^T$, we conclude that $\mathcal{H}_V \in \langle\!\langle F \rangle\!\rangle^T$.
In case that $F = (F_1 \mid F_2)$. Then, $\mathcal{H} \in \langle\!\langle F_1 \rangle\!\rangle^S$ or $\mathcal{H} \in \langle\!\langle F_2 \rangle\!\rangle^S$ and, thus, the result follows directly by induction hypothesis. □

Lemma 18. *Let $V, S, S' \subseteq At$ be sets of atoms such that $S \cap V = S' \cap V$ and take F a fork such that $At(F) \subseteq V$. Then, for any $\preceq$-maximal S-support $\mathcal{H} \in \langle\!\langle F \rangle\!\rangle^S$, there exists $\mathcal{H}' \in \langle\!\langle F \rangle\!\rangle^{S'}$ such that $\mathcal{H}_V \preceq \mathcal{H}'_V$.*

Proof. In case that F is a propositional formula, we have that $\langle\!\langle F \rangle\!\rangle^S = \downarrow [\![F]\!]^S$ and, thus, $\mathcal{H} = [\![F]\!]^S$. Hence, $\mathcal{H}_V = ([\![F]\!]^S)_V = [\![F]\!]^T = ([\![F]\!]^{S'})_V$ by applying Corollary 10. Otherwise, we assume as induction hypothesis that the lemma statement holds for every subfork of F.
In case that $F = F_1 \wedge F_2$. Then, $[\ ] \neq \mathcal{H} = \mathcal{H}_1 \cap \mathcal{H}_2$ with $\mathcal{H}_1 \in \langle\!\langle F_1 \rangle\!\rangle^S$ and $\mathcal{H}_2 \in \langle\!\langle F_2 \rangle\!\rangle^S$. Furthermore, let us assume without loss of generality that each $\mathcal{H}_i$ is $\preceq$-maximal in $\langle\!\langle F_i \rangle\!\rangle^S$. Notice that each $\mathcal{H}_i$ is V-respectful in view of Lemma 6. Then, $\mathcal{H}_V = (\mathcal{H}_1 \cap \mathcal{H}_2)_V = (\mathcal{H}_1)_V \cap (\mathcal{H}_2)_V$ by Lemma 13. By induction hypothesis, we have that $(\mathcal{H}_i)_V \preceq (\mathcal{H}'_i)_V$ with $\mathcal{H}'_i \in \langle\!\langle F_i \rangle\!\rangle^{S'}$ and $\mathcal{H}_V \preceq \mathcal{H}'_V$ with $\mathcal{H}' = (\mathcal{H}'_1 \cap \mathcal{H}'_2) \in \langle\!\langle F \rangle\!\rangle^{S'}$.

In case that $F = \varphi \to G$. First of all, notice that $S \models \varphi$ iff $T \models \varphi$ iff $S' \models \varphi$. In case that $S \not\models \varphi$, $\mathcal{H} = 2^S$ and $\mathcal{H}_V = 2^T = (2^{S'})_V$ with $2^{S'} \in \langle\!\langle F \rangle\!\rangle^{S'}$. Otherwise $\langle\!\langle F \rangle\!\rangle^S = {\downarrow}\{\ \overline{[\![\varphi]\!]^S} \cup \mathcal{H}' \mid \mathcal{H}' \in \langle\!\langle G \rangle\!\rangle^S\ \}$. That is, $\mathcal{H} = \overline{[\![\varphi]\!]^S} \cup \mathcal{H}'$ for some $\mathcal{H}' \in \langle\!\langle G \rangle\!\rangle^S$.

Let also assume that $\mathcal{H}'$ is $\preceq$-maximal in $\langle\!\langle G \rangle\!\rangle^S$. Hence, we know that there exists $\mathcal{H}'' \in \langle\!\langle G \rangle\!\rangle^{S'}$ such that $\mathcal{H}'_V \preceq \mathcal{H}''_V$ and it follows that:

$$\begin{aligned}
\mathcal{H}_V &= (\overline{[\![\varphi]\!]^S} \cup \mathcal{H}')_V \\
&= (\overline{[\![\varphi]\!]^S})_V \cup \mathcal{H}'_V \\
&= (\overline{[\![\varphi]\!]^{S'}})_V \cup \mathcal{H}'_V \\
&\preceq (\overline{[\![\varphi]\!]^{S'}})_V \cup \mathcal{H}''_V \\
&= (\overline{[\![\varphi]\!]^{S'}} \cup \mathcal{H}'')_V
\end{aligned}$$

and $\overline{[\![\varphi]\!]^{S'}} \cup \mathcal{H}'' \in \langle\!\langle F \rangle\!\rangle^{S'}$.
In case that $F = (F_1 \mid F_2)$. Then, $\mathcal{H} \in \langle\!\langle F_1 \rangle\!\rangle^S$ or $\mathcal{H} \in \langle\!\langle F_2 \rangle\!\rangle^S$ and, thus, the result follows directly by induction hypothesis. □

Lemma 19. *Given sets of atoms $T \subseteq V \subseteq At$ and forks F, L such that $At(L) \subseteq V$, we have:*

$$\langle\!\langle F \wedge L \rangle\!\rangle^T_V = {}^{\downarrow}\{\, \mathcal{H} \cap \mathcal{H}' \;\big|\; \mathcal{H} \in \langle\!\langle F \rangle\!\rangle^T_V \text{ and } \mathcal{H}' \in \langle\!\langle L \rangle\!\rangle^T \,\}$$

Proof. "$\subseteq$" $\mathcal{H} \in \langle\!\langle F \wedge L \rangle\!\rangle^T_V$

implies there are $S \subseteq At$ and $\mathcal{H}' \in \langle\!\langle F \wedge L \rangle\!\rangle^S$ s.t. $T = S \cap V$, $\mathcal{H} \preceq \mathcal{H}'_V$ and $\mathcal{H}'$ is V-feasible,
implies there are $S \subseteq At$ and $\mathcal{H}', \mathcal{H}_1, \mathcal{H}_2$ s.t. $T = S \cap V$, $\mathcal{H}_1 \in \langle\!\langle F \rangle\!\rangle^S$, $\mathcal{H}_2$ maximal in $\langle\!\langle L \rangle\!\rangle^S$, $\mathcal{H} \preceq \mathcal{H}'_V$, $\mathcal{H}' \preceq \mathcal{H}_1 \cap \mathcal{H}_2$ and $\mathcal{H}'$ is V-feasible,
implies there are $S \subseteq At$ and $\mathcal{H}', \mathcal{H}_1, \mathcal{H}_2$ s.t. $T = S \cap V$, $\mathcal{H}_1 \in \langle\!\langle F \rangle\!\rangle^S$, $\mathcal{H}_2$ maximal in $\langle\!\langle L \rangle\!\rangle^S$, $\mathcal{H} \preceq \mathcal{H}'_V$, $\mathcal{H}' \preceq \mathcal{H}_1 \cap \mathcal{H}_2$, $\mathcal{H}_1 \cap \mathcal{H}_2$ is V-feasible (Lemma 10) and $\mathcal{H}_2$ is V-respectful (Lemma 6),
implies there are $S \subseteq At$ and $\mathcal{H}', \mathcal{H}_1, \mathcal{H}_2$ s.t. $T = S \cap V$, $\mathcal{H}_1 \in \langle\!\langle F \rangle\!\rangle^S$, $\mathcal{H}_2$ maximal in $\langle\!\langle L \rangle\!\rangle^S$, $\mathcal{H} \preceq \mathcal{H}'_V$, $\mathcal{H}' \preceq \mathcal{H}_1 \cap \mathcal{H}_2$, $\mathcal{H}_1$ is V-feasible (Lemma 12) and $\mathcal{H}_2$ is V-respectful,
implies there are $S \subseteq At$ and $\mathcal{H}', \mathcal{H}_1, \mathcal{H}_2$ s.t. $T = S \cap V$, $\mathcal{H}_1 \in \langle\!\langle F \rangle\!\rangle^S$, $\mathcal{H}_2$ maximal in $\langle\!\langle L \rangle\!\rangle^S$, $\mathcal{H} \preceq \mathcal{H}'_V \preceq (\mathcal{H}_1 \cap \mathcal{H}_2)_V$ (Lemma 11), $(\mathcal{H}_1)_V \in \langle\!\langle F \rangle\!\rangle^T_V$ and $\mathcal{H}_2$ is V-respectful,
implies there are $S \subseteq At$ and $\mathcal{H}_1, \mathcal{H}_2$ s.t. $T = S \cap V$, $\mathcal{H}_1 \in \langle\!\langle F \rangle\!\rangle^S$, $\mathcal{H}_2$ maximal in $\langle\!\langle L \rangle\!\rangle^S$, $\mathcal{H} \preceq (\mathcal{H}_1 \cap \mathcal{H}_2)_V \preceq (\mathcal{H}_1)_V \cap (\mathcal{H}_2)_V$ (Lemma 13) and $(\mathcal{H}_1)_V \in \langle\!\langle F \rangle\!\rangle^T_V$,
implies there are $S \subseteq At$ and $\mathcal{H}_1, \mathcal{H}_2$ s.t. $T = S \cap V$, $\mathcal{H}_2 \in \langle\!\langle L \rangle\!\rangle^S$, $\mathcal{H} \preceq (\mathcal{H}_1)_V \cap (\mathcal{H}_2)_V$ and $(\mathcal{H}_1)_V \in \langle\!\langle F \rangle\!\rangle^T_V$,
implies there are $S \subseteq At$ and $\mathcal{H}_1, \mathcal{H}_2$ s.t. $T = S \cap V$, $(\mathcal{H}_2)_V \in \langle\!\langle L \rangle\!\rangle^T$ (Lemma 17), $\mathcal{H} \preceq (\mathcal{H}_1)_V \cap (\mathcal{H}_2)_V$ and $(\mathcal{H}_1)_V \in \langle\!\langle F \rangle\!\rangle^T_V$,
implies $\mathcal{H} \preceq \mathcal{H}'_1 \cap \mathcal{H}'_2$ with $\mathcal{H}'_1 \in \langle\!\langle F \rangle\!\rangle^T_V$ and $\mathcal{H}'_2 \in \langle\!\langle L \rangle\!\rangle^T$ (with $\mathcal{H}'_i = (\mathcal{H}_i)_V$).

"$\supseteq$" $\mathcal{H} \cap \mathcal{H}'$ with $\mathcal{H} \in \langle\!\langle F \rangle\!\rangle^T_V$ and $\mathcal{H}' \in \langle\!\langle L \rangle\!\rangle^T$

implies there are $S \subseteq At$, $\mathcal{H}_1 \in \langle\!\langle F \rangle\!\rangle^S$, $\mathcal{H}_2 \in \langle\!\langle L \rangle\!\rangle^S$ s.t. $T = S \cap V$, $\mathcal{H} \preceq (\mathcal{H}_1)_V$ and $\mathcal{H}' = \mathcal{H}'_V \preceq (\mathcal{H}_2)_V$ (Lemma 9 and Lemma 18),
implies $\mathcal{H} \cap \mathcal{H}' \preceq (\mathcal{H}_1)_V \cap (\mathcal{H}_2)_V \preceq (\mathcal{H}_1 \cap \mathcal{H}_2)_V$ with $\mathcal{H}_1 \cap \mathcal{H}_2 \in \langle\!\langle F \wedge L \rangle\!\rangle^S$ (Proposition 5 (ii) and Lemma 13),
implies $\mathcal{H} \cap \mathcal{H}' \in \langle\!\langle F \wedge L \rangle\!\rangle^T_V$. □

Corollary 11. *Given sets of atoms $T \subseteq V \subseteq At$, a fork F and a propositional formula φ such that $At(\varphi) \subseteq V$, we have:*

$$\langle\!\langle F \wedge \varphi \rangle\!\rangle^T_V = {}^{\downarrow}\{\, \mathcal{H} \cap [\![\varphi]\!]^T \;\big|\; \mathcal{H} \in \langle\!\langle F \rangle\!\rangle^T_V \,\}$$

Proof. It is a consequence of Lemma 19 and Lemma 1. □

Proof of Theorem 2. Suppose that $F \mathrel{|\!\sim}_V G$ and take $\mathcal{H} \in \langle\!\langle F \rangle\!\rangle^T_V$. If we apply Lemma 8, we know we can find φ with $At(\varphi) \subseteq V$ and such that $\mathcal{H} \cap [\![\varphi]\!]^T = [\ T\ ]$ and $\mathcal{H}' \cap [\![\varphi]\!]^T \neq [\ T\ ]$ for any $\mathcal{H}'$ such that $\mathcal{H} \not\preceq \mathcal{H}'$. Taking into account Corollary 11 and Proposition 14, it is clear that $[\ T\ ] \in \langle\!\langle F \wedge \varphi \rangle\!\rangle^T_V$ and $T \in SM_V[F \wedge \varphi] \subseteq SM_V[G \wedge \varphi]$ which implies that $T \in \langle\!\langle G \wedge \varphi \rangle\!\rangle^T_V$. This implies that there exists $\mathcal{H}' \in \langle\!\langle G \rangle\!\rangle^T_V$ such that $[\ T\ ] = \mathcal{H}' \cap [\![\varphi]\!]^T$. Finally, $\mathcal{H} \preceq \mathcal{H}'$ and $\mathcal{H} \in \langle\!\langle G \rangle\!\rangle^T_V$.

For the other direction, take L a fork with $At(L) \subseteq V$ and $T \in SM_V[F \wedge L]$. By applying Proposition 14 and Lemma 19, we have that $[\ T\ ] = \mathcal{H} \cap \mathcal{H}'$ with $\mathcal{H} \in \langle\!\langle F \rangle\!\rangle^T_V$ and $\mathcal{H}' \in \langle\!\langle L \rangle\!\rangle^T$. Since, by hypothesis, $\mathcal{H} \in \langle\!\langle G \rangle\!\rangle^T_V$, we have that $[\ T\ ] \in \langle\!\langle G \wedge L \rangle\!\rangle^T_V$ and finally $T \in SM_V[G \wedge L]$. □

Proof of Proposition 15. One implication is obvious. For the other one, it is enough to show that $\langle\!\langle F \rangle\!\rangle^T_V \subseteq \langle\!\langle G \rangle\!\rangle^T_V$ for any $T \subseteq V$. Take $\mathcal{H} \in \langle\!\langle F \rangle\!\rangle^T_V$, then, by applying Lemma 8, we can find γ verifying that $At(\gamma) \subseteq V$, $\mathcal{H} \cap [\![\gamma]\!]^T = [\ T\ ]$ and

$\mathcal{H}' \cap [\![\gamma]\!]^T \neq [\ T\ ]$ for every T-support $\mathcal{H}'$ such that $\mathcal{H} \not\preceq \mathcal{H}'$. This means that $[\ T\ ] \in \langle\!\langle F \wedge \gamma \rangle\!\rangle_V^T$ or equivalently $T \in SM_V[F \wedge \gamma]$. Our hypothesis implies that $T \in SM_V[G \wedge \gamma]$ or $[\ T\ ] \in \langle\!\langle G \wedge \gamma \rangle\!\rangle_V^T$, so, by Corollary 11, there exists $\mathcal{H}'' \in \langle\!\langle G \rangle\!\rangle_V^T$ such that $[\ T\ ] = \mathcal{H}'' \cap [\![\gamma]\!]^T$. Then, mandatorily $\mathcal{H} \preceq \mathcal{H}''$ and, finally, we get that $\mathcal{H} \in \langle\!\langle G \rangle\!\rangle_V^T$. □

From now on, take $At', V \subseteq At$ and $V' \stackrel{\text{def}}{=} V \cap At'$. Notice that, for any $X \subseteq At$, if $X \cap V \subseteq At'$, then $X \cap V = X \cap V'$. In particular, if $X \subseteq At'$, it holds that $X \cap V = X \cap V'$.

Lemma 20. *Take F a fork such that $At(F) \subseteq At'$ and $S \subseteq At$. Then, if $\mathcal{H}$ is an S-support, it holds that*

$$\mathcal{H} \in \langle\!\langle F \rangle\!\rangle^S \text{ is equivalent to } \mathcal{H}_{At'} \in \langle\!\langle F \rangle\!\rangle^{S \cap At'}$$

Proof. We only prove one of the directions. The proof of the other one is similar. We proceed by structural induction on F. If $F = \varphi$, the result is consequence of Lemma 11 and Lemma 5.
If $F = F_1 \wedge F_2$ and $\mathcal{H} \in \langle\!\langle F \rangle\!\rangle^S$, then $\mathcal{H} \preceq \mathcal{H}_1 \cap \mathcal{H}_2$ with $\mathcal{H}_i \in \langle\!\langle F_i \rangle\!\rangle^S$ $(i = 1, 2)$. We can take $\mathcal{H}_i$ maximal in $\langle\!\langle F_i \rangle\!\rangle^S$ so $\mathcal{H}_i$ is At'-respectful by Lemma 6 and $\mathcal{H}_{At'} \preceq (\mathcal{H}_1)_{At'} \cap (\mathcal{H}_2)_{At'}$ by Lemma 13. Since, by induction each $(\mathcal{H}_i)_{At'} \in \langle\!\langle F_i \rangle\!\rangle^{S \cap At'}$, we conclude that $\mathcal{H}_{At'} \in \langle\!\langle F \rangle\!\rangle^{S \cap At'}$.
Let $F = \varphi \to F_1$ with φ a propositional formula and F_1 a fork such that $At(\varphi), At(F_1) \subseteq At'$. First notice that $S \not\models \varphi$ iff $S \cap At' \not\models \varphi$ and $(2^S)_{At'} = 2^{S \cap At'}$. If $S \models \varphi$ and $\mathcal{H} \in \langle\!\langle F \rangle\!\rangle^S$, there exists $\mathcal{H}_1 \in \langle\!\langle F_1 \rangle\!\rangle^S$ such that $\mathcal{H} \preceq \overline{[\![\varphi]\!]^S} \cup \mathcal{H}_1$. Then

$$\mathcal{H}_{At'} \preceq (\overline{[\![\varphi]\!]^S} \cup \mathcal{H}_1)_{At'} = (\overline{[\![\varphi]\!]^S})_{At'} \cup (\mathcal{H}_1)_{At'} = \overline{[\![\varphi]\!]^{S \cap At'}} \cup (\mathcal{H}_1)_{At'} \in \langle\!\langle F \rangle\!\rangle^{S \cap At'}$$

since we can apply Lemma 14, Corollary 10 and induction.
Finally, suppose that $F = (F_1 \mid F_2)$ and $\mathcal{H}$ is maximal in $\langle\!\langle F \rangle\!\rangle^S = \langle\!\langle F_1 \rangle\!\rangle^S \cup \langle\!\langle F_2 \rangle\!\rangle^S$. Again, applying induction, we can deduce the result. □

Lemma 21. *Take $S \subseteq At$ such that $S \cap V \subseteq At'$. Then, if $\mathcal{H}$ is an S-support and $\mathcal{H}$ is V-feasible, then $\mathcal{H}_{At'}$ is V'-feasible and $\mathcal{H}_V = (\mathcal{H}_{At'})_{V'}$*

Proof. Notice that, if $H \in \mathcal{H}$, then $H \cap V \subseteq S \cap V \subseteq At'$ and $H \cap V = H \cap V' = H \cap At' \cap V'$. □

Lemma 22. *Take $S' \subseteq At'$. Then, if $\mathcal{H}$ is an S'-support and $\mathcal{H}$ is V'-feasible, then $\mathcal{H}$ is V-feasible and $\mathcal{H}_V = \mathcal{H}_{V'}$*

Proof. Notice that, if $H \in \mathcal{H}$, then $H \subseteq S' \subseteq At'$, so $H \cap V = H \cap V'$. □

Lemma 23. *Take $T \subseteq V'$ and F a fork such that $At(F) \subseteq At'$. Then:*

$$\langle\!\langle F \rangle\!\rangle_V^T = \langle\!\langle F \rangle\!\rangle_{V'}^T$$

Proof. "$\subseteq$" Take $\mathcal{H}_V$ with $\mathcal{H} \in \langle\!\langle F \rangle\!\rangle^S$, V-feasible and $S \subseteq At$ such that $S \cap V = T$. Then $\mathcal{H}_{At'} \in \langle\!\langle F \rangle\!\rangle^{S \cap At'}$ is V'-feasible by Lemma 20 and Lemma 21. Since $S \cap At' \cap V' = S \cap V' \cap At' = S \cap V = T$, we have that $\mathcal{H}_V = (\mathcal{H}_{At'})_{V'} \in \langle\!\langle F \rangle\!\rangle_{V'}^T$
"$\supseteq$" Take $\mathcal{H}_{V'}$ with $\mathcal{H} \in \langle\!\langle F \rangle\!\rangle^{S'}$, V'-feasible and $S' \subseteq At'$ such that $S' \cap V' = T$. Then $\mathcal{H}$ is V-feasible by Lemma 22 and $\mathcal{H}_{V'} = \mathcal{H}_V \in \langle\!\langle F \rangle\!\rangle_V^T$ since $S' \cap V = S' \cap V' = T$. □

Lemma 24. *Suppose that $At = At' \cup \{a\}$ with $a \notin At'$. Take $S, V \subseteq At$ such that $a \in S \cap V$. It holds that:*

1. *If $\mathcal{H}$ is an At'-respectful, V-feasible S-support, then the support $\mathcal{H}_{At'}$ is V'-feasible,*
2. *If F is a fork with $At(F) \subseteq At'$ and $\mathcal{H}$ is an S-support maximal in $\langle\!\langle F \rangle\!\rangle^S$ and V-feasible, then $\mathcal{H}_{At'}$ is V'-feasible.*

Proof.

1. Take $H' \in \mathcal{H}_{At'}$ verifying that $H' \cap V' = S \cap At' \cap V' = S \cap V'$. We know that $H' = H \cap At'$ for some $H \in \mathcal{H}$. Since $(H' \cup \{a\}) \cap At' = H \cap At'$ and $\mathcal{H}$ is At'-respectful, then $H' \cup \{a\} \in \mathcal{H}$. Moreover:

$$(H' \cup \{a\}) \cap V = (H' \cap V) \cup \{a\} = (H' \cap V') \cup \{a\} = (S \cap V') \cup \{a\} = S \cap V$$

   which implies that $H' \cup \{a\} = S$ and $H' = S \cap At'$.
2. If $\mathcal{H}$ is maximal in $\langle\!\langle F \rangle\!\rangle^S$, then $\mathcal{H}$ is At'-respectful by Lemma 6, so by the previous item $\mathcal{H}_{At'}$ is V'-feasible if $\mathcal{H}$ is V-feasible. □

Lemma 25. *Suppose that $At = At' \cup \{a\}$ with $a \notin At'$, $S' \subseteq At'$ and $V \subseteq At$ such that $a \in V$. If $\mathcal{H}$ is an S'-support, we can define the S-support, where $S = S' \cup \{a\}$:*

$$\mathcal{H}(a) := \{\, H \subseteq S' \cup \{a\} \;\big|\; H \setminus \{a\} = H \cap At' \in \mathcal{H} \,\}$$

It holds that:

1. *If $\mathcal{H}$ is V'-feasible, then $\mathcal{H}(a)$ is V-feasible,*
2. *$\mathcal{H}(a)_{At'} = \mathcal{H}$, and*
3. *$\mathcal{H}(a)_V = \{\, H \cup \{a\} \;\big|\; H \in \mathcal{H}_{V'} \,\} \cup \{\, H \;\big|\; H \in \mathcal{H}_{V'} \,\}$.*

Proof.

1. Suppose that $H \subseteq S' \cup \{a\}$ verifies that $H \cap V = (S' \cup \{a\}) \cap V$. Then $a \in H$ and $H \cap At' \cap V' = H \cap V' = S' \cap V'$ so $H \cap At' = S'$ because $\mathcal{H}$ is V'-feasible and $H = (H \cap At') \cup \{a\} = S' \cup \{a\}$.
2. $H' \in \mathcal{H}(a)_{At'}$ iff $H' = H \cap At'$ for some $H \in \mathcal{H}(a)$ iff $H' = H \cap At' \in \mathcal{H}$.
3. "$\subseteq$" Take $H \cap V \in \mathcal{H}(a)_V$ with $H \in \mathcal{H}(a)$. If we suppose that $a \in H$, then:

$$H \cap V = ((H \setminus \{a\}) \cup \{a\}) \cap V = ((H \setminus \{a\}) \cap V) \cup \{a\} = ((H \setminus \{a\}) \cap V') \cup \{a\}$$

   and $H \setminus \{a\} \in \mathcal{H}$. If $a \notin H$, then $H = H \setminus \{a\} \in \mathcal{H}$ and

$$H \cap V = (H \setminus \{a\}) \cap V = (H \setminus \{a\}) \cap V'$$

   "$\supseteq$" Take $H \in \mathcal{H}_{V'}$. Then $H = \tilde{H} \cap V'$ with $\tilde{H} \in \mathcal{H}$. On the one hand:

$$H = \tilde{H} \cap V' = \tilde{H} \cap V$$

   and $\tilde{H} = \tilde{H} \cap At' \in \mathcal{H}(a)$. On the other hand:

$$H \cup \{a\} = (\tilde{H} \cap V') \cup \{a\} = (\tilde{H} \cap V) \cup \{a\} = (\tilde{H} \cup \{a\}) \cap V$$

   and $\tilde{H} \cup \{a\} \in \mathcal{H}(a)$. □

Lemma 26. *Suppose that $At = At' \cup \{a\}$ with $a \notin At'$. Take $S \subseteq At$ such that $a \in S$. If $\mathcal{H}$ is an At'-respectful S-support, then $\mathcal{H}_{At'}(a) = \mathcal{H}$.*

Proof. Take $H \subseteq S$. Then, $H \in \mathcal{H}_{At'}(a)$ iff $H \cap At' \in \mathcal{H}_{At'}$ iff $H \cap At' = H_1 \cap At'$ for some $H_1 \in \mathcal{H}$. This is equivalent to $H \in \mathcal{H}$ since $\mathcal{H}$ is At'-respectful. □

Proof of Theorem 3. We will prove that the two conditions:

(i) For all $T \subseteq V$, $\langle\!\langle F \rangle\!\rangle^T_V \subseteq \langle\!\langle G \rangle\!\rangle^T_V$ under signature At
(ii) For all $T \subseteq V'$, $\langle\!\langle F \rangle\!\rangle^T_{V'} \subseteq \langle\!\langle G \rangle\!\rangle^T_{V'}$ under signature At'

are equivalent. To prove (i)⇒(ii), take $T \subseteq V'$. Then, by Lemma 23

$$\langle\!\langle F \rangle\!\rangle^T_{V'} = \langle\!\langle F \rangle\!\rangle^T_V \subseteq \langle\!\langle G \rangle\!\rangle^T_V = \langle\!\langle G \rangle\!\rangle^T_{V'}.$$

To prove (ii)⇒(i), take $T \subseteq V$. If $T \subseteq At'$, by Lemma 23, we have that

$$\langle\!\langle F \rangle\!\rangle^T_V = \langle\!\langle F \rangle\!\rangle^T_{V'} \subseteq \langle\!\langle G \rangle\!\rangle^T_{V'} = \langle\!\langle G \rangle\!\rangle^T_V.$$

If $T \subseteq V$ but $T \nsubseteq At'$, we know that $T = (T \cap At') \cup (T \cap \{a\}) = T' \cup \{a\}$ with $T' = T \cap At'$. Take $\tilde{\mathcal{H}} \in \langle\!\langle F \rangle\!\rangle^T_V$. We know that there exists $S \subseteq At$ with $S \cap V = T$ and $\mathcal{H} \in \langle\!\langle F \rangle\!\rangle^S$ V-feasible such that $\tilde{\mathcal{H}} = \mathcal{H}_V$. First of all, notice that, without loss of generality, we can suppose that $\mathcal{H}$ is maximal in $\langle\!\langle F \rangle\!\rangle^S$ and then At'-respectful because of Lemma 6. If we apply Lemma 24 and Lemma 20, we can say that $\mathcal{H}_{At'}$ is V'-feasible and $\mathcal{H}_{At'} \in \langle\!\langle F \rangle\!\rangle^{S \cap At'}$, so $(\mathcal{H}_{At'})_{V'} \in \langle\!\langle F \rangle\!\rangle^{T'}_{V'} \subseteq \langle\!\langle G \rangle\!\rangle^{T'}_{V'}$. Now, take $\tilde{S} \subseteq At'$ and $\mathcal{H}_1 \in \langle\!\langle G \rangle\!\rangle^{\tilde{S}}$ V'-feasible such that $\tilde{S} \cap V' = T'$ and $(\mathcal{H}_{At'})_{V'} \preceq (\mathcal{H}_1)_{V'}$ or, equivalently $(\mathcal{H}_1)_{V'} \subseteq (\mathcal{H}_{At'})_{V'}$. By Lemma 25, the support

$$\mathcal{H}_1(a) := \{\, H \subseteq \tilde{S} \cup \{a\} \;\big|\; H \setminus \{a\} = H \cap At' \in \mathcal{H}_1 \,\}$$

is V-feasible and:

$$\mathcal{H}_1(a)_V = \{\, H \cup \{a\} \mid H \in (\mathcal{H}_1)_{V'} \,\} \cup \{\, H \mid H \in (\mathcal{H}_1)_{V'} \,\}$$

Since $(\mathcal{H}_1)_{V'} \subseteq (\mathcal{H}_{At'})_{V'}$, we can deduce that

$$\mathcal{H}_1(a)_V \subseteq (\mathcal{H}_{At'}(a))_V = \mathcal{H}_V$$

by applying Lemma 26. Finally, notice that $\tilde{H} = \mathcal{H}_V \preceq \mathcal{H}_1(a)_V \in \langle\!\langle\, G \,\rangle\!\rangle_V^T$ taking account that $(\tilde{S} \cup \{a\}) \cap V = (\tilde{S} \cap V) \cup \{a\} = T$ and by applying Lemma 20 and Lemma 25. □

Proof of Corollary 5. We proceed by induction on $n = |At \setminus At'|$. When $n = 1$, we apply Theorem 3.
Now, suppose that $At = At' \cup \{a_1, \dots, a_n\} = At'' \cup \{a_n\}$ with $At'' = At' \cup \{a_1, \dots, a_{n-1}\}$. Take $V \subseteq At$ and denote by $V' = V \cap At'$ and $V'' = V \cap At''$. It holds that $F \mathrel{|\!\!\sim}_V G$ under signature At is equivalent to $F \mathrel{|\!\!\sim}_{V''} G$ under signature At'' since $At(F) \cup At(G) \subseteq At'' \subseteq At = At'' \cup \{a_n\}$ by applying Theorem 3. Finally, by induction hypothesis we have that $F \mathrel{|\!\!\sim}_{V''} G$ under signature At'' is equivalent to $F \mathrel{|\!\!\sim}_{V'} G$ under signature At' since $At(F) \cup At(G) \subseteq At' \subseteq At'' = At' \cup \{a_1, \dots, a_{n-1}\}$ and $V' = V'' \cap At'$. □

Proof of Proposition 16. If $T \subseteq V$ and $\mathcal{H}$ is maximal in $\sigma(T)$, we know by applying Lemma 7 that the set $S_i = \{\, \langle H, T\rangle \mid H \in \mathcal{H} \,\}$ defines a propositional formula Φ_{S_i} which satisfies that $[\![\, \Phi_{S_i} \,]\!]^T = \mathcal{H}$ and $[\![\, \Phi_{S_i} \,]\!]^Y = [\;]$ for any $Y \subseteq V$ such that $Y \neq T$. If $G = (\Phi_{S_1} \mid \cdots \mid \Phi_{S_n})$ and $T \subseteq V$, it follows that:

$$\langle\!\langle\, G \,\rangle\!\rangle^T = \bigcup_{i=1}^{n} {\downarrow}[\![\, \Phi_{S_i} \,]\!]^T = {}^{\downarrow}\{\, \mathcal{H} \mid \mathcal{H} \text{ is maximal in } \sigma(T) \,\} = \sigma(T) \qquad \square$$

Proof of Theorem 4. Any fork F defines an assignment σ_F so that $\sigma_F(T) := \langle\!\langle\, F \,\rangle\!\rangle_V^T$ for each $T \subseteq V$. We know by Proposition 16, that there exists a fork G with $At(G) \subseteq V$ and such that $\langle\!\langle\, G \,\rangle\!\rangle^T = \sigma_F(T)$. Finally, we have that:

$$\langle\!\langle\, G \,\rangle\!\rangle_V^T = \langle\!\langle\, G \,\rangle\!\rangle^T = \sigma_F(T) = \langle\!\langle\, F \,\rangle\!\rangle_V^T$$

And, by Proposition 13 and Theorem 2, if follows that $G \cong_V F$. □

Proof of Proposition 17. Take Δ a T-view with a $\preceq$-maximum element $\mathcal{H}$. Then, by Lemma 7, we know that there is a propositional formula φ with $At(\varphi) \subseteq V$ that satisfies $[\![\, \varphi \,]\!]^T = \mathcal{H}$ and $[\![\, \varphi \,]\!]^{T'} = [\;]$ for every $T' \subseteq V$ with $T' \neq T$. Finally $\Delta = {\downarrow}\mathcal{H} = {\downarrow}[\![\, \varphi \,]\!]^T = \langle\!\langle\, \varphi \,\rangle\!\rangle^T = \langle\!\langle\, \varphi \,\rangle\!\rangle_V^T$ and, for every $T' \subseteq V$ with $T' \neq T$, we have that $\langle\!\langle\, \varphi \,\rangle\!\rangle_V^{T'} = \langle\!\langle\, \varphi \,\rangle\!\rangle^{T'} = {\downarrow}[\![\, \varphi \,]\!]^{T'} = {\downarrow}[\;] = \emptyset$. □

Proof of Proposition 18. Take $T \subseteq At(F)$. First note that, from Proposition 13, it follows that

$$\langle\!\langle\, F \,\rangle\!\rangle_{At(F)}^T \;=\; \langle\!\langle\, F \,\rangle\!\rangle^T \;=\; {}^{\downarrow}\{\; [\![\, \varphi_1 \,]\!]^T, \dots, [\![\, \varphi_n \,]\!]^T \;\}$$

Note that, for any $T \subseteq At(F)$, we have that $[\![\, a_1 \vee \cdots \vee a_n \,]\!]^T = [\;]$ and, thus, $\langle\!\langle\, \gamma(F) \,\rangle\!\rangle^T = \emptyset$ also holds. Furthermore, for any a_i and $T_i = T \cup \{\, a_i \,\}$, we have that $[\![\, a_1 \vee \cdots \vee a_n \,]\!]^{T_i} = \{\, H \subseteq T_i \mid a_i \in H \,\}$. Hence,

$$[\![\, (a_1 \vee \cdots \vee a_n) \wedge (a_i \to \varphi_i) \,]\!]^{T_i} = \{\, H \cup \{\, a_i \,\} \mid H \in [\![\, \varphi_i \,]\!]^T \,\}$$
$$[\![\, (a_j \to \varphi_j) \,]\!]^{T_i} = 2^{T_i}$$

for all $i \neq j$. This implies that $[\![\, \gamma(F) \,]\!]^{T_i} = \{\, H \cup \{\, a_i \,\} \mid H \in [\![\, \varphi_i \,]\!]^T \,\}$ and, thus, we get

$$\langle\!\langle\, \gamma(F) \,\rangle\!\rangle_{At(F)}^T \;\supseteq\; {}^{\downarrow}\{\; [\![\, \varphi_1 \,]\!]^T, \dots, [\![\, \varphi_n \,]\!]^T \;\}$$

Let us take now $T' \supset T \cup \{\, a_i \,\}$ for any $1 \leq i \leq n$. If $[\![\, a_j \to \varphi_j \,]\!]^{T'} = [\;]$ for some $1 \leq j \leq n$, we have that $[\![\, \gamma(F) \,]\!]^{T'} = [\;]$ and that T' does not contribute to $\langle\!\langle\, \gamma(F) \,\rangle\!\rangle_{At(F)}^T$. Hence, we may assume without loss of generality that $[\![\, a_j \to \varphi_j \,]\!]^{T'} \neq [\;]$ for all $1 \leq j \leq n$. Note also that $H = T \cup \{\, a_i \,\}$ belongs to $[\![\, a_1 \vee \cdots \vee a_n \,]\!]^{T'}$ and that $H \cap At(F) = T' \cap At(F)$. Then, to show that $[\![\, \gamma(F) \,]\!]^{T'}$ is $At(F)$-unfeasible we just need to show that $H \in [\![\, a_j \to \varphi_j \,]\!]^{T'}$ for all $1 \leq j \leq n$. Note that $[\![\, a_j \to \varphi_j \,]\!]^{T'} \neq [\;]$ implies $T' \in [\![\, \varphi_j \,]\!]^{T'}$ and

$$[\![\, a_j \to \varphi_j \,]\!]^{T'} = \overline{[\![\, a_j \,]\!]^{T'}} \cup [\![\, \varphi_j \,]\!]^{T'}$$

Furthermore, from Lemma 5 and the facts $H \cap At(F) = T' \cap At(F)$ and $At(\varphi_j) \subseteq At(F)$, we have that $T' \in [\![\, \varphi_j \,]\!]^{T'}$ implies $H \in [\![\, \varphi_j \,]\!]^{T'}$ and, thus, also $H \in [\![\, a_j \to \varphi_j \,]\!]^{T'}$. Consequently, $[\![\, \gamma(F) \,]\!]^{T'}$ is $At(F)$-unfeasible for any $T' \supset T \cup \{\, a_i \,\}$ and

$$\langle\!\langle\, \gamma(F) \,\rangle\!\rangle_{At(F)}^T \;=\; {}^{\downarrow}\{\; [\![\, \varphi_1 \,]\!]^T, \dots, [\![\, \varphi_n \,]\!]^T \;\} \;=\; \langle\!\langle\, F \,\rangle\!\rangle_{At(F)}^T$$

follow. □

Proof of Proposition 23. For the only if direction. Let $\langle \mathcal{X}, T\rangle$ be a minimal (V, V)-certificate. From Proposition 22, it follows that there exists some $S \subseteq At$ such that $T = S \cap V$, $\emptyset \neq [\![\varphi]\!]^S$ is V-feasible and $[\![\varphi]\!]^S_V \setminus \{T\} = \mathcal{X}$. Take $\mathcal{H} = [\![\varphi]\!]^S_V$. Suppose that there exists $S' \subseteq At$ such that $S' \cap V = T$, $\emptyset \neq [\![\varphi]\!]^{S'}$ is V-feasible and $[\![\varphi]\!]^S_V \preceq [\![\varphi]\!]^{S'}_V$ or $[\![\varphi]\!]^{S'}_V \subseteq [\![\varphi]\!]^{S'}_V$. Then $\langle \mathcal{Y}, T\rangle$ with $\mathcal{Y} = [\![\varphi]\!]^{S'}_V \setminus \{T\}$ is a (V, V) certificate of φ and $\mathcal{Y} \subseteq \mathcal{X}$. This implies that $\mathcal{Y} = \mathcal{X}$ or $[\![\varphi]\!]^S_V = [\![\varphi]\!]^{S'}_V$.
The other way around. Let $\mathcal{H}$ maximal in $\langle\!\langle \varphi \rangle\!\rangle^T_V$ such that $\mathcal{H} \setminus \{T\} = \mathcal{X}$. We know that $\mathcal{H} = [\![\varphi]\!]^S_V$ for some $S \subseteq At$ such that $T = S \cap V$ and $\emptyset \neq [\![\varphi]\!]^S$ V-feasible. It is clear that, if $\mathcal{X} = [\![\varphi]\!]^S_V \setminus \{T\}$, the tuple $\langle \mathcal{X}, T\rangle$ is a (V, V)-certificate of φ. Moreover, if $\langle \mathcal{Y}, T\rangle$ is another (V, V)-certificate of φ with $\mathcal{Y} \subseteq \mathcal{X}$, then $\mathcal{Y} = [\![\varphi]\!]^{S'}_V \setminus \{T\}$ for some $S' \subseteq At$ such that $S' \cap V = T$ and satisfying that $\emptyset \neq [\![\varphi]\!]^{S'}$ is V-feasible. Since $\mathcal{H} \preceq [\![\varphi]\!]^{S'}_V$, by maximality of $\mathcal{H}$, it follows that $\mathcal{H} = [\![\varphi]\!]^{S'}_V$ or, equivalently, $\mathcal{X} = \mathcal{Y}$. $\square$

Proof of Theorem 6. Any fork in normal form $F = (\varphi_1 \mid \cdots \mid \varphi_n)$ can be transformed into the (log-space constructible) V-strongly equivalent propositional formula

$$\gamma(F) = (a_1 \vee \cdots \vee a_n) \wedge (a_1 \rightarrow \varphi_1) \wedge \cdots \wedge (a_n \rightarrow \varphi_n)$$

where each a_j for $j = 1, \ldots, n$ is a new fresh atom not included in V (Proposition 18). Applying the same transformation $\gamma(G)$ to G we have reduced the problem to projective strong entailment/equivalence of propositional formulas, and then Proposition 24 applies. $\square$

References

[1] C. Baral, Knowledge Representation, Reasoning and Declarative Problem Solving, Cambridge University Press, 2003.
[2] M. Gebser, B. Kaufmann, T. Schaub, Conflict-driven answer set solving: from theory to practice, Artif. Intell. 187–188 (2012) 52–89.
[3] W. Faber, G. Pfeifer, N. Leone, T. Dell'Armi, G. Ielpa, Design and implementation of aggregate functions in the DLV system, Theory Pract. Log. Program. 8 (5–6) (2008) 545–580.
[4] E. Erdem, M. Gelfond, N. Leone, Applications of answer set programming, AI Mag. 37 (3) (2016) 53–68.
[5] M. Gelfond, V. Lifschitz, The stable model semantics for logic programming, in: R. Kowalski, K. Bowen (Eds.), Proceedings of the 19th International Conference and Symposium of Logic Programming (ICLP'88), MIT Press, 1988, pp. 1070–1080.
[6] D. Pearce, A new logical characterisation of stable models and answer sets, in: J. Dix, L.M. Pereira, T.C. Przymusinski (Eds.), Selected Papers from the Non-Monotonic Extensions of Logic Programming (NMELP'96), in: Lecture Notes in Artificial Intelligence, vol. 1216, Springer-Verlag, 1996, pp. 57–70.
[7] D. Pearce, Equilibrium logic, Ann. Math. Artif. Intell. 47 (1–2) (2006) 3–41.
[8] P. Ferraris, J. Lee, V. Lifschitz, A new perspective on stable models, in: R. Sangal, H. Mehta, R.K. Bagga (Eds.), Proceedings of the 20th International Joint Conference on Artificial Intelligence (IJCAI'07), Morgan Kaufmann Publishers Inc., 2007, pp. 372–379.
[9] A. Harrison, V. Lifschitz, M. Truszczynski, On equivalence of infinitary formulas under the stable model semantics, Theory Pract. Log. Program. 15 (1) (2015) 18–34.
[10] R. Gonçalves, M. Knorr, J. Leite, You can't always forget what you want: on the limits of forgetting in answer set programming, in: G.A. Kaminka, M. Fox, P. Bouquet, E. Hüllermeier, V. Dignum, F. Dignum, F. van Harmelen (Eds.), Proceedings of 22nd European Conference on Artificial Intelligence (ECAI'16), in: Frontiers in Artificial Intelligence and Applications, vol. 285, IOS Press, 2016, pp. 957–965.
[11] V. Lifschitz, D. Pearce, A. Valverde, Strongly equivalent logic programs, ACM Trans. Comput. Log. 2 (4) (2001) 526–541.
[12] T. Eiter, H. Tompits, S. Woltran, On solution correspondences in answer-set programming, in: L.P. Kaelbling, A. Saffiotti (Eds.), Proceedings of the Nineteenth International Joint Conference on Artificial Intelligence (IJCAI'05), Professional Book Center, 2005, pp. 97–102.
[13] P. Cabalar, D. Pearce, A. Valverde, Reducing propositional theories in equilibrium logic to logic programs, in: C. Bento, A. Cardoso, G. Dias (Eds.), Proceedings of the 12th Portuguese Conference on Progress in Artificial Intelligence (EPIA'05), in: Lecture Notes in Computer Science, vol. 3808, Springer, 2005, pp. 4–17.
[14] P. Cabalar, P. Ferraris, Propositional theories are strongly equivalent to logic programs, Theory Pract. Log. Program. 7 (6) (2007) 745–759.
[15] P. Cabalar, D. Pearce, A. Valverde, Minimal logic programs, in: V. Dahl, I. Niemelä (Eds.), Proceedings of the 23rd International Conference on Logic Programming, (ICLP'07), Springer, 2007, pp. 104–118.
[16] P. Ferraris, Answer sets for propositional theories, in: C. Baral, G. Greco, N. Leone, G. Terracina (Eds.), Proc. of the 8th Intl. Conf. on Logic Programming and Nonmonotonic Reasoning (LPNMR'05), in: Lecture Notes in Computer Science, vol. 3662, Springer, 2005, pp. 119–131.
[17] F. Aguado, P. Cabalar, D. Pearce, G. Pérez, C. Vidal, A denotational semantics for equilibrium logic, Theory Pract. Log. Program. 15 (4–5) (2015) 620–634.
[18] D. Pearce, Algebra matters in ASP and equilibrium logic, in: M. Ojeda Aciego (Ed.), Inma P. de Guzmán Festschrift (Liber Amicorum), 2010, http://sevein.matap.uma.es/~aciego/liber/resources/Algebra-EL.pdf.
[19] R. Ben-Eliyahu, R. Dechter, Propositional semantics for disjunctive logic programs, Ann. Math. Artif. Intell. 12 (1–2) (1994) 53–87.
[20] T. Eiter, M. Fink, S. Woltran, Semantical characterizations and complexity of equivalences in answer set programming, ACM Trans. Comput. Log. 8 (3) (2007) 17.
[21] D. Pearce, H. Tompits, S. Woltran, Characterising equilibrium logic and nested logic programs: reductions and complexity, Theory Pract. Log. Program. 9 (5) (2009) 565–616.
[22] J. Leite, A Bird's-eye view of forgetting in answer-set programming, in: M. Balduccini, T. Janhunen (Eds.), Proc. of the 14th Intl. Conf. on Logic Programming and Nonmonotonic Reasoning (LPNMR'17), in: Lecture Notes in Computer Science, vol. 10377, Springer, 2017, pp. 10–22.
[23] M. Knorr, J.J. Alferes, Preserving strong equivalence while forgetting, in: E. Fermé, J. Leite (Eds.), Logics in Artificial Intelligence - 14th European Conference, JELIA 2014, Funchal, Madeira, Portugal, September 24–26, 2014. Proceedings, in: Lecture Notes in Computer Science, vol. 8761, Springer, 2014, pp. 412–425.
[24] M. Fink, A general framework for equivalences in answer-set programming by countermodels in the logic of here-and-there, Theory Pract. Log. Program. 11 (2–3) (2011) 171–202.
[25] J. Vennekens, M. Denecker, M. Bruynooghe, CP-logic: a language of causal probabilistic events and its relation to logic programming, Theory Pract. Log. Program. 9 (3) (2009) 245–308.
[26] A. Bochman, On disjunctive causal inference and indeterminism, in: Proceedings of the Workshop on Nonmonotonic Reasoning, Actions and Change (NRAC'03), 2003, pp. 45–50.