algorithms

MDPI

Article

Comparing the Reasoning Capabilities of Equilibrium Theories and Answer Set Programs

Jorge Fandinno [1], David Pearce [2,*], Concepción Vidal [3] and Stefan Woltran [4]

[1] College of Information Science & Technology, University of Nebraska Omaha, 6001 Dodge St., Omaha, NE 68182, USA; jfandinno@unomaha.edu
[2] Departamento de Inteligencia Artificial, Universidad Politécnica de Madrid, 28040 Madrid, Spain
[3] Computer Science and Information Technologies, University of Corunna, 15008 A Coruña, Spain; concepcion.vidalm@udc.es
[4] Databases and Artificial Intelligence Group (E192-02), Institute of Logic and Computation, Technische Universität Wien, 1040 Vienna, Austria; stefan.woltran@tuwien.ac.at
* Correspondence: david.pearce@upm.es

Abstract: Answer Set Programming (ASP) is a well established logical approach in artificial intelligence that is widely used for knowledge representation and problem solving. Equilibrium logic extends answer set semantics to more general classes of programs and theories. When intertheory relations are studied in ASP, or in the more general form of equilibrium logic, they are usually understood in the form of comparisons of the answer sets or equilibrium models of theories or programs. This is the case for strong and uniform equivalence and their relativised and projective versions. However, there are many potential areas of application of ASP for which query answering is relevant and a comparison of programs in terms of what can be inferred from them may be important. We formulate and study some natural equivalence and entailment concepts for programs and theories that are couched in terms of inference and query answering. We show that, for the most part, these new intertheory relations coincide with their model-theoretic counterparts. We also extend some previous results on projective entailment for theories and for the new connective called *fork*.

Keywords: answer set programming; equilibrium logic; intertheory relations; projective entailment; forks in ASP

Citation: Fandinno, J.; Pearce, D.; Vidal, C.; Woltran, S. Comparing the Reasoning Capabilities of Equilibrium Theories and Answer Set Programs. *Algorithms* **2022**, *15*, 201. https://doi.org/10.3390/a15060201

Academic Editor: Giovanni Amendola

Received: 1 April 2022
Accepted: 2 June 2022
Published: 8 June 2022

1. Introduction

Answer Set Programming (ASP) is a popular environment for knowledge representation and problem solving in artificial intelligence. Thanks to efficient answer set solvers, there are now many applications of this technique in diverse domains. Equilibrium logic [1] provides a logical foundation for ASP and extends the stable model semantics to arbitrary propositional and first order theories. It has also proved instrumental in generating novel extensions for temporal and epistemic reasoning under answer set semantics. Answer set programs are typically employed to provide one or more representations of problem solutions in the form of models or answer sets. Unlike Prolog, ASP was not traditionally designed to be used as a query answering system. Accordingly, the study of logical relations between programs, beginning with [2], has focused mainly on a comparison of their answer sets. This is how the principle relations of strong and uniform equivalence as well as their relativised and projective versions are understood [2–5]. Strong and uniform equivalence were first considered in the Datalog domain [6,7]; since then, many nonmonotonic formalisms have been investigated in this regard, including default logic [8], causal logics [9], argumentation frameworks [10,11], and preference-based formalisms [12–14], to mention just a few examples. Results on strong equivalence and related notions have proved to be useful in many different contexts, such as program simplification [15] or forgetting [16].

Nevertheless, query answering is important in a number of domains where ASP can be applied. In such cases, we are interested in comparing the logical consequences of different programs, to see, for example, under what conditions they agree in their answering of queries. This motivates the challenge of defining and analysing equivalence and entailment relations between programs that are formulated in terms of inference and query answering. Similar notions turn up in other logic-based reasoning systems. In Datalog, there is the well-known concept of *query containment* [17]. In description logics for reasoning over ontologies, there is the concept of *query inseparability* (see e.g., [18]); similarly, in abstract argumentation, strong equivalence with respect to *argument acceptance* has been investigated [11].

In ASP, two programs are said to be *equivalent* if they have the same stable models and *strongly equivalent* if they remain equivalent under the addition of any new set of rules [2]. If only new facts can be added, the relation is known as *uniform equivalence*. Furthermore, *relativised* versions of strong and uniform equivalence can be defined to cover the case that the newly added rules or facts are in a specific language. *Projective* equivalence is the appropriate concept in case we are interested in model equivalence with respect to a restricted sublanguage of the programs. All these relations have been studied and characterised in the literature (as cited above). In [19], weak and strong forms of *entailment* between programs were also defined and analysed. In this case, we are concerned with the relative *strength* of theories.

In the rest of the article, we formulate new intertheory and inter-program relations based on inference and show that, in most cases, they coincide with their well-known model-theoretic counterparts. This means that they can be studied using familiar concepts and techniques. We work throughout in the non-classical logic **HT** of *here-and-there*, which provides a basis for equilibrium logic and hence for the stable model semantics of ASP. **HT** is of particular importance in this context due to the way in which **HT** models relate to theory equivalences. In Section 2, we recall the basic features of **HT** and equilibrium logic, and we define three types of consequence relations for equilibrium theories. Section 3 deals with strong and uniform equivalence relations, showing how the standard notions compare with their counterparts formulated in terms of inference. An analogous procedure for relativised equivalence follows in Section 4. In Section 5, we turn to entailment relations between theories. In other words, we deal with the relative *strength* of theories in terms of both their stable models and their question answering capabilities. Section 6 examines projective entailment and equivalence relations; here, we generalise some of the characterisations obtained in previous work. We include projective entailment and equivalence for standard logic programs and also cover two further cases. One is the extension to arbitrary propositional theories. The other deals with an extension of the usual vocabulary of programs to include a new type of disjunction connective, '$|$', called *fork*, introduced in [20]. As explained in Section 6.2, the intuitive meaning of this construct is that when we form the stable models of $\Pi_1 \mid \Pi_2$, they correspond to the union of the stable models from Π_1 and Π_2. This continues to be the case when further rules Π' are added.

ASP is well suited to formalise rule-based *policies* and often one is interested in the *consequences* that can be derived from such policies, given relevant background information and data. It follows that the kinds of intertheory correspondences we have been studying are relevant when we want to compare different policies in a logical manner. We include in Section 7 some paragraphs describing an outline of how such a policy formalisation might look in the case of access control policies, a domain in which logic programs have been successfully applied in the past. This sketch may help to illustrate how our intertheory relations may provide useful concepts for reasoning about such policies.

There is a substantial body of literature devoted to logical relations between answer set programs and between theories in equilibrium logic. It is beyond the remit of this article to describe all the many characterisation results and their applications. However, in the concluding Section 8, we list some additional research articles where the reader can find many of the most important results. We moved some of the longer proofs, especially those from Section 6, to an Appendix A.

2. Logical Preliminaries and Basic Definitions

We work in the logic **HT** of *here-and-there* first presented in [21]. This is a three-valued extension of intuitionistic propositional logic. It can be built up in a simple manner by considering two kinds of truth: provable truth and truth by default applying to propositions that are not false but not provably true [22]. We also rely on *equilibrium logic*, a nonmonotonic extension of **HT** based on a concept of minimal model [1]. Equilibrium logic captures the stable model semantics of ASP for arbitrary propositional theories. Moreover, **HT** is of great value when studying intertheory relations, because theories and programs are strongly equivalent if and only if they are logically equivalent in **HT**.

The language of **HT** is built up in the usual way from a set At of atoms called the *(propositional) signature*. A *(propositional) formula* φ is defined using the usual grammar:

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

where p is an atom $p \in At$. Greek letters φ, ψ, γ and their variants stand for formulas. We also consider derived operators $\neg\varphi \stackrel{\text{def}}{=} (\varphi \rightarrow \bot)$, $\top \stackrel{\text{def}}{=} \neg\bot$ and $\varphi \leftrightarrow \psi \stackrel{\text{def}}{=} (\varphi \rightarrow \psi) \wedge (\psi \rightarrow \varphi)$. A *literal* is an atom p or its negation $\neg p$. A *theory* is a set of formulas. A (general) *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 *disjunctive logic program* is a program such that for each of its implications $\alpha \rightarrow \beta$, β is a disjunction of *atoms*. In other words, the formulas of a disjunctive program have precisely the form of what are usually called logic programming *rules*, where α is the rule *body* and β is the rule *head*. We denote theories and programs by upper-case Greek letters, $\Gamma, \Pi, \Sigma \ldots$, and $At(\Pi)$ denotes the set of atomic formulas present in Π. Throughout the article, we restrict attention to *finite* languages, theories, and programs.

A model-theoretic semantics for **HT** can be based on the usual possible-worlds models for intuitionistic logic (see e.g., [23]), but **HT** is complete for frames $\mathcal{F} = \langle W, \leq \rangle$ (where, as usual, W is the set of points or worlds and $\leq$ is a partial-ordering on W) having exactly two worlds, say h ('here') and t ('there') with $h \leq t$. As usual, a *model* is a frame together with an assignment i that associates to each element of W a set of *atoms*, such that if $w \leq w'$, then $i(w) \subseteq i(w')$. An assignment is then extended inductively to all formulas via the usual rules for conjunction, disjunction, implication, and negation in intuitionistic logic, namely

$$\begin{aligned}
&\bot \notin i(w)\\
&(\varphi \wedge \psi) \in i(w) \ \text{ iff } \ \varphi \in i(w) \ \text{ and } \ \psi \in i(w)\\
&(\varphi \vee \psi) \in i(w) \ \text{ iff } \ \varphi \in i(w) \ \text{ or } \ \psi \in i(w)\\
&(\varphi \rightarrow \psi) \in i(w) \ \text{ iff for all } w' \text{ s.t. } w \leq w', \ \varphi \in i(w') \ \text{ implies } \ \psi \in i(w')\\
&\neg\varphi \in i(w) \ \text{ iff for all } w' \text{ s.t. } w \leq w', \ \varphi \notin i(w')
\end{aligned}$$

Although the final clause is obtained from those for $\rightarrow$ and $\bot$, we include it to make it clear that $\neg\varphi$ is true at either world just in case $\varphi \notin i(t)$. It is convenient to represent an **HT** model as an ordered pair $\langle H, T \rangle$ of sets of atoms, where $H = i(h)$ and $T = i(t)$ under a suitable assignment i; by $h \leq t$, it follows that $H \subseteq T$. When $H = T$, we say that the interpretation $\langle H, T \rangle$ is *total*. Note that in a model $\langle H, T \rangle$, H represents the set of certain or provable atoms whereas T represents the set of true atoms of either kind.

We write $\mathcal{M}, w \models \varphi$ to denote that a formula φ is true or *forced* at world w in an **HT** model $\mathcal{M}$, i.e., $\varphi \in i(w)$. Then, φ is true in $\mathcal{M}$, in symbols $\mathcal{M} \models \varphi$, if $\mathcal{M}, h \models \varphi$. A formula φ is said to be a *consequence* of a theory Π, in symbols $\Pi \models \varphi$, if $\mathcal{M} \models \varphi$ for each model $\mathcal{M}$ of Π. We denote by $Th(\mathcal{M})$ the collection of all formulas true in $\mathcal{M}$.

Equilibrium Logic

To define equilibrium logic, we first introduce a partial ordering $\trianglelefteq$ on **HT** models.

Definition 1. *Given any two models, $\langle H, T \rangle$, $\langle H', T' \rangle$, we set $\langle H, T \rangle \trianglelefteq \langle H', T' \rangle$ if $T = T'$ and $H \subseteq H'$.*

This leads to the following notion of equilibrium.

Definition 2. *Let Π be a theory and $\langle H, T\rangle$ a model of Π. Then, $\langle H, T\rangle$ is said to be an* equilibrium *model of Π if it is minimal under $\trianglelefteq$ among models of Π, and it is total.*

In other words, a model $\langle T, T\rangle$ of Π is in equilibrium if there is no model $\langle H, T\rangle$ of Π with $H \subset T$. In this case, we say that T is a stable model or answer set of Π. Equilibrium logic is the logic determined by the equilibrium models of a theory. Our terminology is justified by the following property:

Proposition 1 ([24]). *Let Π be a disjunctive logic program. Then, a set of atoms T is a stable model (or answer set) of Π (in the standard sense [25]) if and only if $\langle T, T\rangle$ is an equilibrium model of Π.*

Because a theory or program under stable model semantics usually possesses more than one stable or equilibrium model, different notions of inference can be considered depending on the particular problem domain represented by the theory. The more usual relation in ASP is sceptical inference; however, there are applications where a credulous form of inference is more appropriate. We also include a third type of *prudent* inference. For a theory Π, we denote by $E(\Pi)$ the set of its equilibrium models. We say that a theory Π is *stable* if $E(\Pi) \neq \emptyset$. When it is clear that we are dealing with an equilibrium model or any total (i.e., classical) model $\mathcal{M}$, we also simply denote it by its corresponding set M of (true) atoms.

Definition 3 (Equilibrium consequence)**.** *The relations of* equilibrium consequence, *credulous ($\Pi \models_c \varphi$), sceptical ($\Pi \models_s \varphi$), and prudent ($\Pi \models_p \varphi$) are defined as follows. Let Π be a theory. Then,*

- $\Pi \models_c \varphi$ *if Π is stable and $\varphi \in \bigcup_{\mathcal{M} \in E(\Pi)} Th(\mathcal{M})$;*
- $\Pi \models_s \varphi$ *if Π is stable and $\varphi \in \bigcap_{\mathcal{M} \in E(\Pi)} Th(\mathcal{M})$*
- $\Pi \models_p \varphi$ *if Π is stable and $\varphi \in Th(\bigcap_{\mathcal{M} \in E(\Pi)} (\mathcal{M}))$*

These relations differ: Take for instance $\Pi = \{a \vee b\}$. We have, $E(\Pi) = \{\{a\}, \{b\}\}$. Thus, $\Pi \models_c a$ but not $\Pi \not\models_s a$ nor $\Pi \not\models_p a$. Moreover, $\Pi \models_s a \vee b$ but $\Pi \not\models_p a \vee b$. In general, we have that $\Pi \models_p \varphi$ implies $\Pi \models_s \varphi$ and $\Pi \models_s \varphi$ implies $\Pi \models_c \varphi$. In our example, $\Pi \models_p \neg a$ (because $\emptyset \models \neg a$) but $\Pi \not\models_s \neg a$ because $\{a\} \not\models \neg a$.

3. Equivalence Notions

Based on these relations, we now can define different notions of equivalence between theories.

Definition 4. *Let Π_1 and Π_2 be theories and $\alpha \in \{c, s, p\}$. Then, $\Pi_1 \equiv_\alpha \Pi_2$ states that for any formula φ, $\Pi_1 \models_\alpha \varphi$ iff $\Pi_2 \models_\alpha \varphi$.*

We are now able to compare theories in further different ways:

Definition 5. *Let Π_1 and Π_2 be theories and $\alpha \in \{c, s, p\}$. Then,*

- $\Pi_1 \equiv^s \Pi_2$ *holds iff for any further theory Π, $(\Pi_1 \cup \Pi) \equiv (\Pi_2 \cup \Pi)$.*
- $\Pi_1 \equiv^u \Pi_2$ *holds iff for any set X of atoms, $(\Pi_1 \cup X) \equiv (\Pi_2 \cup X)$.*
- $\Pi_1 \equiv^s_\alpha \Pi_2$ *holds iff for any further theory Π, $(\Pi_1 \cup \Pi) \equiv_\alpha (\Pi_2 \cup \Pi)$.*
- $\Pi_1 \equiv^u_\alpha \Pi_2$ *holds iff for any set X of atoms, $(\Pi_1 \cup X) \equiv_\alpha (\Pi_2 \cup X)$.*

The first two relations are well known. Following standard terminology, we say that Π_1 and Π_2 are *equivalent* if $E(\Pi_1) = E(\Pi_2)$, *strongly equivalent* if $\Pi_1 \equiv^s \Pi_2$ and *uniform equivalent* if $\Pi_1 \equiv^u \Pi_2$. These relations are well understood. Π_1 and Π_2 are *strongly equivalent* if and only if they are equivalent in the logic **HT**; in other words, they have the same **HT** models [2]. Uniform equivalence on the other hand is captured by a special set of **HT** countermodels [26] (see also [3,27]).

Evidently, if theories are equivalent, then their inference relations are also equivalent for all types of inference. However, what happens if two theories have different equilibrium models; can we always separate them in terms of sentences they entail? The following lemma answers this in the affirmative for credulous and sceptical inference. To simplify notation, we treat an equilibrium model as a set of atoms M.

Lemma 1. *Let Π_1 and Π_2 be stable theories such that $E(\Pi_1) \neq E(\Pi_2)$; say that Π_1 has an equilibrium model M that is not an equilibrium model of Π_2. Then:*

(i) There exists a sentence φ such that $M \models \varphi$ but φ is false in all equilibrium models of Π_2. Hence, $\Pi_1 \models_c \varphi$, $\Pi_2 \not\models_c \varphi$, and so $\Pi_1 \not\equiv_c \Pi_2$.

(ii) There exists a sentence ψ such that $M \not\models \psi$ but $M_i \models \psi$ for each equilibrium model $M_i \in E(\Pi_2)$. Hence, $\Pi_2 \models_s \psi$ whereas $\Pi_1 \not\models_s \psi$ and so $\Pi_1 \not\equiv_s \Pi_2$.

Proof. (i) For each $M_i \in E(\Pi_2)$, we know that $M \cap (U \setminus M_i) \neq \emptyset$ or $M_i \cap (U \setminus M) \neq \emptyset$ with $U = At(\Pi_1 \cup \Pi_2)$. Set

$$\varphi_i = \bigwedge_{a \in M \cap (U \setminus M_i)} a \wedge \bigwedge_{b \in M_i \cap (U \setminus M)} \neg b$$

Then, $M \models \varphi_i$, for each i, so $M \models \varphi$ where $\varphi = \bigwedge \varphi_i$ and $\Pi_1 \models_c \varphi$. However, for any $M_i \in E(\Pi_2)$, $M_i \not\models \varphi$ which implies that $\Pi_2 \not\models_c \varphi$.

(ii) Similarly, for each equilibrium model M_i of Π_2 either there is some atom $a \in M$ such that $M_i \models \neg a$ or else there is some atom $b \in U \setminus M$ such that $M' \models b$, where U is the set of atoms in $\Pi_1 \cup \Pi_2$. Set

$$\psi_i = \bigvee_{a \in M \cap (U \setminus M_i)} \neg a \vee \bigvee_{b \in (U \setminus M) \cap M_i} b$$

Then, $\psi = \bigvee \psi_i$ holds in each equilibrium model of Π_2 and so $\Pi_2 \models_s \psi$. By inspection $M \not\models \psi$ and so $\Pi_1 \not\models_s \psi$. □

Now it is straightforward to characterise entailment equivalence.

Proposition 2. *The following conditions hold:*

- $\Pi_1 \equiv_c \Pi_2$ *iff* $\Pi_1 \equiv_s \Pi_2$ *iff* $E(\Pi_1) = E(\Pi_2)$.
- $\Pi_1 \equiv_\alpha \Pi_2$ *implies* $\Pi_1 \equiv_p \Pi_2$ *for* $\alpha \in \{c, s\}$.

The next results show that in the strong-equivalence setting, the choice of the consequence operator does not play a role. Together with the observation from above, we conclude that all three notions are characterised by the logic of here-and-there.

Proposition 3. *The following propositions are equivalent:*

(i) $\Pi_1 \equiv^s \Pi_2$;
(ii) $\Pi_1 \equiv^s_c \Pi_2$;
(iii) $\Pi_1 \equiv^s_s \Pi_2$;
(iv) $\Pi_1 \equiv^s_p \Pi_2$.

Proof. (Parts (i)–(iii)) Clearly (i) implies (ii) and (iii). However, if (i) does not hold, then there is a theory Π such that $E(\Pi_1 \cup \Pi) \neq E(\Pi_2 \cup \Pi)$. Apply Lemma 1 to conclude that neither $\Pi_1 \equiv_c^s \Pi_2$ nor $\Pi_1 \equiv_s^s \Pi_2$ holds.

$$
\begin{aligned}
\Pi_1 \equiv^s \Pi_2 \;&\Leftrightarrow\; \forall \Pi,\, E(\Pi_1 \cup \Pi) = E(\Pi_2 \cup \Pi) \\
&\Leftrightarrow\; \forall \Pi,\, \Pi_1 \cup \Pi \equiv_c \Pi_2 \cup \Pi && \text{by Proposition 1} \\
&\Leftrightarrow\; \Pi_1 \equiv_c^s \Pi_2 \\
&\Leftrightarrow\; \forall \Pi,\, E(\Pi_1 \cup \Pi) = E(\Pi_2 \cup \Pi) \\
&\Leftrightarrow\; \forall \Pi,\, \Pi_1 \cup \Pi \equiv_s \Pi_2 \cup \Pi && \text{by Proposition 2} \\
&\Leftrightarrow\; \Pi_1 \equiv_s^s \Pi_2
\end{aligned}
$$

□

Proof. ($\Pi_1 \equiv^s \Pi_2 \Leftrightarrow \Pi_1 \equiv_p^s \Pi_2$) It suffices to show that $\Pi_1 \equiv_p^s \Pi_2 \Rightarrow \Pi_1 \equiv^s \Pi_2$. Suppose then that $\Pi_1 \not\equiv^s \Pi_2$. Hence, there exists a Π such that $(\Pi_1 \cup \Pi) \not\equiv_c (\Pi_2 \cup \Pi)$. By Proposition 2 we can assume $Y \in E(\Pi_1 \cup \Pi) \setminus E(\Pi_2 \cup \Pi)$. Consider

$$\Pi' = \{ \bigwedge_{y \in Y} \neg\neg y \wedge \bigwedge_{z \in U \setminus Y} \neg z \}$$

where U is the set of atoms occurring in $\Pi_1 \cup \Pi_2 \cup \Pi$. Then, Y is the only equilibrium model of $\Pi_1 \cup \Pi \cup \Pi'$ whereas $E(\Pi_2 \cup \Pi \cup \Pi') = \emptyset$. This can be seen as follows. The only **HT** models of Π' are of the form (X, Y) with $X \subseteq Y$. Because no $X \subset Y$ is an **HT**-model of $\Pi_1 \cup \Pi$ (by assumption $Y \in E(\Pi_1 \cup \Pi)$), no $X \subset Y$ is an **HT**-model of $\Pi_1 \cup \Pi \cup \Pi'$. Thus, Y is the only equilibrium model of $\Pi_1 \cup \Pi \cup \Pi'$. On the other hand, from assumption $Y \notin E(\Pi_2 \cup \Pi)$ we can have two cases:

(a) (Y, Y) is not an **HT**-model of $\Pi_2 \cup \Pi$. Then, Y obviously cannot become an equilibrium model of $\Pi_2 \cup \Pi \cup \Pi'$, or
(b) (Y, Y) is an **HT**-model of $\Pi_2 \cup \Pi$ but then there exists $X \subset Y$ such that (X, Y) is an **HT**-model of $\Pi_2 \cup \Pi$. By definition of Π', (X, Y) is then also an **HT**-model of $\Pi_2 \cup \Pi \cup \Pi'$, and so Y cannot be an equilibrium model of $\Pi_2 \cup \Pi \cup \Pi'$.

Then, we can conclude that $Y \notin E(\Pi_2 \cup \Pi \cup \Pi')$ and, because Y is the only classical (total) model of $\Pi_2 \cup \Pi \cup \Pi'$, it follows that $E(\Pi_2 \cup \Pi \cup \Pi') = \emptyset$.

Now, using the fact that $\Pi_1 \cup \Pi \cup \Pi'$ has a unique equilibrium model, there obviously exists a (non-tautological) φ such that $(\Pi_1 \cup (\Pi \cup \Pi')) \models_p \varphi$ and $(\Pi_2 \cup (\Pi \cup \Pi')) \not\models_p \varphi$. Hence, $\Pi_1 \not\equiv_p^s \Pi_2$. □

Note that this argument applies to disjunctive logic programs as well (using constraints).

Uniform Equivalence

For the sceptical and credulous cases, the situation with respect to uniform equivalence follows precisely the previous pattern.

Proposition 4. *The following conditions are equivalent:*

(i) $\Pi_1 \equiv^u \Pi_2$;
(ii) $\Pi_1 \equiv_c^u \Pi_2$;
(iii) $\Pi_1 \equiv_s^u \Pi_2$;

Proof.

$$
\begin{aligned}
\Pi_1 \equiv^u \Pi_2 \;&\Leftrightarrow\; \forall X \subseteq At,\, E(\Pi_1 \cup X) = E(\Pi_2 \cup X) \\
&\Leftrightarrow\; \forall X \subseteq At,\, \Pi_1 \cup X \equiv_c \Pi_2 \cup X \qquad \text{by Proposition 2} \\
&\Leftrightarrow\; \Pi_1 \equiv^u_c \Pi_2 \\
&\Leftrightarrow\; \forall X \subseteq At,\, E(\Pi_1 \cup X) = E(\Pi_2 \cup X) \\
&\Leftrightarrow\; \forall X \subseteq At,\, \Pi_1 \cup X \equiv_s \Pi_2 \cup \Pi \qquad \text{by Proposition 2} \\
&\Leftrightarrow\; \Pi_1 \equiv^u_s \Pi_2
\end{aligned}
$$

□

For the prudent case, consider the empty theory and the theory containing the formula $a \vee b$. These two theories are not uniformly equivalent. In fact, they are not even ordinarily equivalent because the former has as its unique stable model $\emptyset$ and the latter has two stable models $\{a\}$ and $\{b\}$. Note that the intersection of all stable models is in both cases $\emptyset$, so they are ordinarily equivalent with respect to prudent queries. Furthermore, we get the same stable models if we add to these theories the contexts $\{a\}$, $\{b\}$, and $\{a, b\}$. So, they are also uniformly equivalent for prudent consequence.

4. Relativised Equivalence

As usual, we consider theories Π_1, Π_2, etc. and now make explicit languages $\mathcal{L}$, $\mathcal{L}'$, etc. As before, we view a language as a set of atoms. A theory is said to be *in* the language $\mathcal{L}$ if all its atomic formulas belong to $\mathcal{L}$.

Definition 6. *Let Π_1 and Π_2 be theories.*

(*i*) Π_1 *and* Π_2 *are* strongly equivalent relative to $\mathcal{L}$, *in symbols* $\Pi_1 \equiv^{s\mathcal{L}} \Pi_2$, *iff for any (empty or non-empty) set* Σ *of* $\mathcal{L}$ *formulas,* $\Pi_1 \cup \Sigma$ *and* $\Pi_2 \cup \Sigma$ *are equivalent, i.e., have the same equilibrium models.*

(*ii*) Π_1 *and* Π_2 *are* uniformly equivalent relative to $\mathcal{L}$, *in symbols* $\Pi_1 \equiv^{u\mathcal{L}} \Pi_2$, *iff for any (empty or non-empty) set* X *of* $\mathcal{L}$ *literals,* $\Pi_1 \cup X$ *and* $\Pi_2 \cup X$ *are equivalent, i.e., have the same equilibrium models.*

We can now apply these definitions to different relativised versions of strong and uniform equivalence.

Definition 7. *Let Π_1 and Π_2 be theories and $\alpha \in \{c, s, p\}$. Then,*

- *we write* $\Pi_1 \equiv^{s,\mathcal{L}}_\alpha \Pi_2$ *if for any further theory* Π *in* $\mathcal{L}$, $(\Pi_1 \cup \Pi) \equiv_\alpha (\Pi_2 \cup \Pi)$;
- *we write* $\Pi_1 \equiv^{u,\mathcal{L}}_\alpha \Pi_2$ *if for any set* X *of* $\mathcal{L}$ *atoms,* $(\Pi_1 \cup X) \equiv_\alpha (\Pi_2 \cup X)$.

Proposition 5. *For any theories Π_1, Π_2 and $\alpha \in \{c, s\}$, $\Pi_1 \equiv^{s,\mathcal{L}}_\alpha \Pi_2$ iff $\Pi_1 \equiv^{s\mathcal{L}} \Pi_2$; similarly $\Pi_1 \equiv^{u,\mathcal{L}}_\alpha \Pi_2$ iff $\Pi_1 \equiv^{u\mathcal{L}} \Pi_2$.*

Proof. The right to left directions are obvious. For the other direction, suppose for instance that Π_1 and Π_2 are not strongly equivalent relative to $\mathcal{L}$. Then, for some set of $\mathcal{L}$ formulas Π, $\Pi_1 \cup \Pi$ and $\Pi_2 \cup \Pi$ have different sets of equilibrium models. We can apply Lemma 1 again to conclude that $\Pi_1 \cup \Pi$ and $\Pi_2 \cup \Pi$ can be separated by different queries, both in the sceptical and in the credulous sense, i.e., that $\Pi_1 \cup \Pi \not\equiv_s \Pi_2 \cup \Pi$ and $\Pi_1 \cup \Pi \not\equiv_c \Pi_2 \cup \Pi$. The uniform case follows the same pattern.

$$
\begin{aligned}
\Pi_1 \equiv^{s,\mathcal{L}}_\alpha \Pi_2 \;&\Leftrightarrow\; \forall \Pi \in \mathcal{L},\, \Pi_1 \cup \Pi \equiv_\alpha \Pi_2 \cup \Pi \\
&\Leftrightarrow\; \forall \Pi \in \mathcal{L},\, E(\Pi_1 \cup \Pi) = E(\Pi_2 \cup \Pi) \qquad \text{by Proposition 2} \\
&\Leftrightarrow\; \Pi_1 \equiv^{s\mathcal{L}} \Pi_2
\end{aligned}
$$

□

5. Entailment Relations

In [19], various nonmonotonic entailment relations are defined in terms of (equilibrium) models. In particular:

Definition 8 (strong and uniform entailment)**.** Π_1 *strongly entails* Π_2*, in symbols* $\Pi_1 \models_S \Pi_2$*, (respective* Π_1 *uniformly entails* Π_2*, in symbols* $\Pi_1 \models_U \Pi_2$*) if for any set* Γ *of formulas (respective atoms)*

$$E(\Pi_1 \cup \Gamma) \subseteq E(\Pi_2 \cup \Gamma) \tag{1}$$

We can also consider entailment relations attuned to query answering.

Definition 9 (weak theory entailment)**.** *Let* Π_1 *and* Π_2 *be theories and* $\omega \in \{c, s, p\}$*. Then, we write* $\Pi_1 \mid\sim_\omega \Pi_2$ *if for any formula* φ*, if* $\Pi_2 \models_\omega \varphi$ *then,* $\Pi_1 \models_\omega \varphi$*.*

As in the case of Definition 2, we can compare theories for their deductive strength.

Definition 10 (strong and uniform theory entailment)**.** *Let* Π_1 *and* Π_2 *be theories and* $\omega \in \{c, s, p\}$*. Then,*

- *we write* $\Pi_1 \mid\sim^s_\omega \Pi_2$ *if for any further theory* Π $(\Pi_1 \cup \Pi) \mid\sim_\omega (\Pi_2 \cup \Pi)$*;*
- *we write* $\Pi_1 \mid\sim^u_\omega \Pi_2$ *if for any set* X *of atoms,* $(\Pi_1 \cup X) \mid\sim_\omega (\Pi_2 \cup X)$*.*

Proposition 6. *For any theories* Π*,* Π_2*:*

1. $\Pi_1 \mid\sim^s_s \Pi_2 \Leftrightarrow \Pi_1 \models_S \Pi_2$
2. $\Pi_1 \mid\sim^u_s \Pi_2 \Leftrightarrow \Pi_1 \models_U \Pi_2$

Proof. (1). The implication from right to left holds by inspection. For the other direction, suppose that $\Pi_1 \not\models_S \Pi_2$. Then, for some Γ, $E(\Pi_1 \cup \Gamma) \not\subseteq E(\Pi_2 \cup \Gamma)$. Let $M \in E(\Pi_1 \cup \Gamma) \setminus E(\Pi_2 \cup \Gamma)$. Apply Lemma 1(ii) to conclude that there is a sentence ψ that holds in each equilibrium model of $\Pi_2 \cup \Gamma$ and so $\Pi_2 \cup \Gamma \models_s \psi$. Inspection $M \not\models \psi$, so $\Pi_1 \cup \Gamma \not\models_s \psi$, and hence $\Pi_1 \not\mid\sim^s_s \Pi_2$.

For (2), the proof is entirely analogous. □

The relation $\Pi_1 \models_S \Pi_2$ has been characterised in terms of **HT**-models.

Proposition 7 ([19])**.** $\Pi_1 \models_S \Pi_2$ *holds iff the following two conditions are satisfied:*

(i) Π_1 *classically entails* Π_2*.*
(ii) *For any model* $\langle H, T \rangle$ *of* Π_2 *such that* $\langle T, T \rangle \models \Pi_1$*,* $\langle H, T \rangle \models \Pi_1$*.*

Although the (strong) equivalence concepts are captured in the monotonic logic **HT**, the same is not true for strong entailment, because $\Pi_1 \models \Pi_2$ does not imply $\Pi_1 \models_S \Pi_2$. However, it is clear that

$$\Pi_1 \models_S \Pi_2 \;\;\&\;\; \Pi_2 \models_S \Pi_1 \Rightarrow \Pi_1 \equiv^s_s \Pi_2 \tag{2}$$

By Propositions 1 and 2, we can see that $\Pi_1 \equiv^s_s \Pi_2$ if and only if Π_1 and Π_2 are strongly equivalent theories. In turn, by the well-known characterisation [2], this means that Π_1 and Π_2 are equivalent in **HT**. It follows that, whereas $\Pi_1 \models \Pi_2$ does not entail $\Pi_1 \models_S \Pi_2$, it is easy to see that $\Pi_1 \equiv_{HT} \Pi_2$ implies both $\Pi_1 \models_S \Pi_2$ and $\Pi_2 \models_S \Pi_1$.

Although the strong equivalence concepts for credulous, sceptical, and prudent reasoning all agree, in the case of strong entailment, credulous and sceptical reasoning behave in a kind of dual form.

Proposition 8. $\Pi_1 \mid\sim^s_s \Pi_2 \Leftrightarrow \Pi_2 \mid\sim^s_c \Pi_1$.

Proof. From Proposition 6, we established that $\Pi_1 \mid\!\sim^s_s \Pi_2$ holds if and only if (1) is true. Clearly, if (1) holds then $\Pi_2 \mid\!\sim^s_c \Pi_1$ by inspection. Suppose then that $\Pi_1 \mid\!\sim^s_s \Pi_2$ does not hold. Then, there is an extension Γ of Π_1 such that there is an equilibrium model in $E(\Pi_1 \cup \Gamma)$ that is not in $E(\Pi_2 \cup \Gamma)$. From the proof of Proposition 3, we can conclude that there is an extension Γ' of $\Pi_1 \cup \Gamma$ and non-tautological formula ψ such that $\Pi_1 \cup \Gamma \cup \Gamma' \models_c \psi$ but $\Pi_2 \cup \Gamma \cup \Gamma' \not\models_c \psi$. Furthermore, $\Pi_2 \not\mid\!\sim^s_c \Pi_1$. □

6. Projective Concepts

6.1. Basic Definitions

Very often we are interested only in certain parts of answer sets or equilibrium models, and the output of solvers may suppress the unwanted parts. If our query is expressed in a sublanguage B of the theory, we need only consult the projection of equilibrium models onto that sublanguage, i.e., we deal with the B-reducts of the equilibrium models (In the remainder of the article, we use upper-case Latin letters, A, B, etc., to denote (sub)languages, i.e., regarded as sets of atoms). This is justified by the next lemma. For notation, let M be a classical model for a language $\mathcal{L}$ and B a sublanguage of $\mathcal{L}$. We denote by $M|B$ the B-reduct of M, i.e., $M \cap B$, and where now truth and falsity in $M|B$ is defined only for formulas expressible in B. Likewise, for any set X of classical models, $X|B$ denotes the set of their B-reducts.

Lemma 2. *Let M be an equilibrium model (of some theory in $\mathcal{L}$) and B a sublanguage of $\mathcal{L}$. Then, for any B-formula φ, $M \models \varphi \Leftrightarrow M|B \models \varphi$.*

Definition 11 (B-consequence)**.** *For $\omega \in \{c, s, p\}$*

(i) We say that Π_2 is a B-consequence *of Π_1, in symbols $\Pi_1 \mid\!\sim_{\omega,B} \Pi_2$, if for any B-formula φ, $\Pi_2 \models_\omega \varphi \Rightarrow \Pi_1 \models_\omega \varphi$.*

(ii) We say that Π_1 and Π_2 are B-inseparable *(for ω), in symbols $\Pi_1 \equiv_{\omega,B} \Pi_2$, if $\Pi_1 \mid\!\sim_{\omega,B} \Pi_2$ and $\Pi_2 \mid\!\sim_{\omega,B} \Pi_1$.*

Strong versions of B consequence and inseparability are obtained in the obvious way.

Definition 12 (strong B-consequence)**.** *For $\omega \in \{c, s, p\}$:*

- *We write $\Pi_1 \mid\!\sim^s_{\omega,B} \Pi_2$ (for* strong B-consequence*) if for any Π and B-formula φ, $\Pi_2 \cup \Pi \models_\omega \varphi \Rightarrow \Pi_1 \cup \Pi \models_\omega \varphi$.*
- *Similarly for* strong B-inseparability*: $\Pi_1 \equiv^s_{\omega,B} \Pi_2$ if for any Π and B-formula φ, $\Pi_1 \cup \Pi \mid\!\sim_{\omega,B} \Pi_2$ and $\Pi_2 \cup \Pi \mid\!\sim_{\omega,B} \Pi_1$.*
- *Relativised versions are easily obtained. e.g.,* strong B-consequence, relative to A, *in symbols $\Pi_1 \mid\!\sim^{s,A}_{\omega,B} \Pi_2$, obtains when for any set Π of A formulas and B-formula φ, $\Pi_2 \cup \Pi \models_\omega \varphi \Rightarrow \Pi_1 \cup \Pi \models_\omega \varphi$.*
- *Similarly,* strong inseparability relative to A *is denoted by $\Pi_1 \equiv^{s,A}_{\omega,B} \Pi_2$.*

Our previous notion of relativised strong equivalence has a straightforward projective version.

Definition 13. *Let Π_1 and Π_2 be theories. Π_1 and Π_2 are* strongly equivalent relative to A projected onto B, *in symbols $\Pi_1 \equiv^{s,A}_B \Pi_2$, if for any (empty or non-empty) set Σ of A formulas, $E(\Pi_1 \cup \Sigma)|B = E(\Pi_2 \cup \Sigma)|B$.*

Proposition 9. *Two theories, Π_1 and Π_2, are strongly B-inseparable relative to A iff they are strongly equivalent relative to A, projected onto B.*

Proof. See Appendix A. □

Definition 14 (strong B-entailment)**.** *Let Π_1 and Π_2 be theories. We say that Π_1* strongly entails Π_2 relative to A, projected onto B, *in symbols $\Pi_1 \mid\!\sim_B^{s,A} \Pi_2$, if for any set Σ of A formulas, $E(\Pi_1 \cup \Sigma)|B \subseteq E(\Pi_2 \cup \Sigma)|B$.*

Strong B-entailment and (relativised) strong B-consequence coincide for $\omega = s$.

Proposition 10. *For any theories Π_1, Π_2:*

$$\Pi_1 \mid\!\sim_B^{s,A} \Pi_2 \Leftrightarrow \Pi_1 \mid\!\sim_{s,B}^{s,A} \Pi_2 \tag{3}$$

Proof. See Appendix A. □

6.2. Forks and Projective B-Entailment for Theories

We have seen that there is good agreement between equivalence and entailment concepts defined in terms of equilibrium models and their analogous counterparts couched in terms of consequence or query answering. As expected, in the case of (projective) strong entailment, the agreement is with the sceptical version of strong B-consequence. For the standard model-theoretic concepts, many characterisation results are known (see Section 8). In the case of projective entailment and equivalence, the main results are those of [3], which apply to disjunctive logic programs (see also the recent work [28]). For the remainder of the article, we consider projective concepts for programs and propositional theories and also make use of the concept of *fork*.

In [20], the language of logic programs was extended to include a new construct '$|$', called *fork*, whose intuitive meaning is that the stable models of $\Pi_1 \mid \Pi_2$ correspond to the union of stable models from Π_1 and Π_2 in any context Π', that is $SM[(\Pi_1 \mid \Pi_2) \wedge \Pi']$ $= SM[\Pi_1 \wedge \Pi'] \cup SM[\Pi_2 \wedge \Pi']$. (Remark: $SM[\Pi]$ denotes the collection of stable models of Π. Because a program (or theory) is in our case finite, we can also regard it as a conjunction of its formulas. This allows us to write expressions such as $\Pi \wedge \Gamma$ or $\Pi \mid \Gamma$ with the obvious meaning). Using the construct of fork, [20] 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. This property corresponds to the one defined in Definition 13 for the case $A = B$. [20] also provides a semantic characterisation of PSE that allows one to prove that it is always possible to forget (under strong persistence) an auxiliary atom in a fork—something shown to be false in standard **HT**. Now, we recall some definitions from [19,20].

Definition 15. *Given $T \subseteq At$, a T*-support *$\mathcal{H}$ is a set of subsets of T, that is $\mathcal{H} \subseteq 2^T$, satisfying $\mathcal{H} \neq \emptyset$ iff $T \in \mathcal{H}$.*

To increase the readability of examples, we can 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 $[\,]$.

It is well-known that, given a propositional formula φ, the set:

$$\{H \subseteq T \mid \langle H, T\rangle \models \varphi\}$$

is always a T-support denoted by $[\![\, \varphi \,]\!]^T$. Moreover, in [3], it was shown that $\mathcal{H} \subseteq 2^T$ is a T-support iff there exists a propositional formula φ such that $\mathcal{H} = [\![\, \varphi \,]\!]^T$.

Example 1.

$$[\![\neg p \to q]\!]^T = \begin{cases} [\,] & \textit{if } T = \emptyset \\ [\emptyset, \{p\}\,] & \textit{if } T = \{p\} \\ [\,\{q\}\,] & \textit{if } T = \{q\} \\ 2^T & \textit{if } T = \{p,q\} \end{cases} \qquad [\![p \vee q]\!]^T = \begin{cases} [\,] & \textit{if } T = \emptyset \\ [\,\{p\}\,] & \textit{if } T = \{p\} \\ [\,\{q\}\,] & \textit{if } T = \{q\} \\ [\,\{p\}, \{q\}, \{p,q\}\,] & \textit{if } T = \{p,q\} \end{cases}$$

If $A \subseteq At$, we say that a T-support $\mathcal{H}$ is *A-feasible* iff there is no $H \subset T$ in $\mathcal{H}$ satisfying $H \cap A = T \cap A$. (Remark: If $\mathcal{H} = [\![\varphi]\!]^T$, for some φ and $T \subseteq V$, suppose that $\mathcal{H}$ is V-unfesasible. Then, there exists $H \subset T$ with $H \cap V = T \cap V = T$ such that $H \in \mathcal{H}$. In this case, T would never be a stable model of $\varphi \wedge \lambda$ if $\lambda \in \mathcal{L}(V)$. Notice that $\langle H, T \rangle \models \varphi$ and $\langle H, T \rangle \models \lambda$).

Lemma 3 (Lemma 7 from [20])**.** *Given $T \subseteq A \subseteq At$ and any T-support $\mathcal{H}$, there is a propositional formula $\varphi_T \in \mathcal{L}_A$ such that $[\![\varphi_T]\!]^T = \mathcal{H}$ and $[\![\varphi_T]\!]^Y = [\,]$ for any $Y \subseteq A$ and $Y \neq T$.*

Definition 16. *Given $T, A \subseteq At$, we say that a T-support $\mathcal{H}$ is A-respectful, if for any $H, H' \subseteq T$, with $H \cap A = H' \cap A$, it follows that $H \in \mathcal{H}$ iff $H' \in \mathcal{H}$.*

Notice that, when $\varphi \in \mathcal{L}_A$, then, for any $T \subseteq At$, $[\![\varphi]\!]^T$ is A-respectful.

Lemma 4 (Lemma 13 from [20])**.** *Let $T, A \subseteq At$ be two sets of atoms and $\mathcal{H}, \mathcal{H}'$ be a pair of T-supports. Then:*

(i) $(\mathcal{H} \cap \mathcal{H}')_A \subseteq \mathcal{H}_A \cap \mathcal{H}'_A$,

(ii) *In addition, if $\mathcal{H}$ is A-*respectful, then

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

We can define an order relation $\preceq$ between T-supports by saying that, given two T-supports, $\mathcal{H}$ and $\mathcal{H}'$, $\mathcal{H} \preceq \mathcal{H}'$ iff either $\mathcal{H} = [\,]$ or $[\,] \neq \mathcal{H}' \subseteq \mathcal{H}$. It is clear that $[\,]$ and $[T]$ are the bottom and top elements, respectively, in the class of all T-supports. Going back to Example 1, it is clear that $[\![\neg p \to q]\!]^T \preceq [\![p \vee q]\!]^T$, for any $T \subseteq \{p,q\}$.
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 consider the *ideal* of $\mathcal{H}$:

$$\downarrow\!\mathcal{H} = \{\mathcal{H}' \mid \mathcal{H}' \preceq \mathcal{H}\} \setminus \{\,[\,]\,\}.$$

Note that the empty support $[\,]$ is not included in the ideal, so $\downarrow\![\,] = \emptyset$.
If Δ is any set of supports:

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

Definition 17. *A T-*view *is a set of T-supports $\Delta \subseteq \mathbf{H}_T$ that is $\preceq$-closed, i.e., ${}^{\downarrow}\Delta = \Delta$.*

A *fork* is defined using the grammar:

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

where φ is a propositional formula and $p \in At$ is an atom. For the definition of T-denotation of a fork, we use 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 $\Delta = \emptyset$.

Definition 18 (***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:*

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

Given $A, B \subseteq At$, the concept of (A, B)-certificate of a program Π is used in [3] to characterise correspondence relations between disjunctive programs. Using denotations, we can say that a pair $(\mathcal{X}, T)$, where $\mathcal{X}$ is a set of interpretations and $T \subseteq A \cup B$, is an (A, B)-certificate of a program Π iff there exists $Z \subseteq At$, such that:

- $Z \cap (A \cup B) = T$ or $Z \cap A = T \cap A$ and $Z \cap B = T \cap B$.
- $[\![\, \Pi \,]\!]^Z \neq \emptyset$ is A-feasible
- $\mathcal{X} = ([\![\, \Pi \,]\!]^Z_A) \setminus \{Z \cap A\}$

It is easy to prove that (minimal) (A, B)-certificates of a program Π correspond to (maximal) elements of the view $\langle\!\langle\, \Pi \,\rangle\!\rangle^T_{A,B}$ where $T \subseteq (A \cup B)$ and:

$$\langle\!\langle\, \Pi \,\rangle\!\rangle^T_{A,B} \stackrel{\text{def}}{=} {}^{\downarrow}\{\, [\![\, \Pi \,]\!]^Z_A \mid \text{ s.t. } Z \cap (A \cup B) = T \text{ and } [\![\, \Pi \,]\!]^Z \text{ is } A\text{−feasible} \,\}$$

In [3], certificates were used to prove:

Lemma 5 (Lemma 1 from [3])**.** *Given two disjunctive programs Π_1 and Π_2 and two sets A, B, it holds that: $\Pi_1 \mid\!\sim^{s,A}_B \Pi_2$ iff, for each (A, B)-certificate $(\mathcal{X}, Y)$ of Π_1, there exists an (A, B)-certificate $(\mathcal{X}', Y)$ of Π_2 with $\mathcal{X}' \subseteq \mathcal{X}$.*

We can now extend this result to show that it holds for general programs and theories and not only for disjunctive programs:

Theorem 1. *Given two programs, Π_1 and Π_2, and two sets, $A, B \subseteq At$, we have that:*

$$\Pi_1 \mid\!\sim^{s,A}_B \Pi_2 \textit{ iff } \langle\!\langle\, \Pi_1 \,\rangle\!\rangle^T_{A,B} \subseteq \langle\!\langle\, \Pi_2 \,\rangle\!\rangle^T_{A,B}, \textit{for any } T \subseteq A \cup B.$$

Consequently, we also have:

Theorem 2. *Given two programs, Π_1 and Π_2, and two sets, $A, B \subseteq At$, it holds that:*

$$\Pi \equiv^{s,A}_B \Pi_2 \textit{ iff } \langle\!\langle\, \Pi_1 \,\rangle\!\rangle^T_{A,B} = \langle\!\langle\, \Pi_2 \,\rangle\!\rangle^T_{A,B}, \textit{for each } T \subseteq A \cup B$$

6.3. Projective B-Entailment for Forks

In order to extend Definition 14 and Theorem 1 to the case of forks, we need some extra definitions and results.

Definition 19 (Definition 7 from [20])**.** *Given a fork F, we say that $Z \subseteq At$ is a* stable model *of F ($Z \in E(F)$) iff $\langle\!\langle\, F \,\rangle\!\rangle^Z = \downarrow\![\, Z \,]$ or, equivalently, $[\, Z \,] \in \langle\!\langle\, F \,\rangle\!\rangle^Z$.*

Definition 20. *Let F and G be forks and $A, B \subseteq At$ two sets. We say that F* strongly entails *G* relative to *A* projected onto *B, in symbols $F \mid\!\sim^{s,A}_B G$, if for any fork L in $\mathcal{L}_A$, $E(F \wedge L)|B \subseteq E(G \wedge L)|B$.*

Definition 21. *Let F and G be forks and $A, B \subseteq At$ two sets. We say that F and G* are strongly equivalent relative to A projected onto B, *in symbols $F \equiv_B^{s,A} G$, if for any fork L in $\mathcal{L}_A$, $E(F \wedge L)|B = E(G \wedge L)|B$.*

In [20], it was shown that, in case $A = B$, we have:

Theorem 3. *Given F and G two forks and $A \subseteq At$, the following holds:*

$$F \mid\!\sim_A^{s,A} G, \textit{ iff, for any } Y \subseteq A, \langle\!\langle F \rangle\!\rangle_A^Y \subseteq \langle\!\langle G \rangle\!\rangle_A^Y$$

We recall from [20] that when F is a fork and $Y \subseteq A \subseteq At$:

$$\langle\!\langle F \rangle\!\rangle_A^Y = {}^{\downarrow}\{\, \mathcal{H}_A \mid \mathcal{H} \in \langle\!\langle F \rangle\!\rangle^Z \text{ s.t. } Z \cap A = Y \text{ and } \mathcal{H} \text{ is } A\text{-feasible} \,\}$$

In order to extend Theorem 3, suppose that F is a fork, $A, B \subseteq At$, and $Y \subseteq A \cup B$. We can define the A-view:

$$\langle\!\langle F \rangle\!\rangle_{A,B}^Y \stackrel{\text{def}}{=} {}^{\downarrow}\{\, \mathcal{H}_A \mid \mathcal{H} \in \langle\!\langle F \rangle\!\rangle^Z \text{ s.t. } Z \cap (A \cup B) = Y \text{ and } \mathcal{H} \text{is } A\text{-feasible} \,\}$$

The following theorem generalizes Theorem 2 from [20] (which would be the case for $A = B$) because

$$\langle\!\langle F \rangle\!\rangle_{A,A}^Y = \langle\!\langle F \rangle\!\rangle_A^Y$$

Theorem 4. *Given F and G two forks and sets $A, B \subseteq At$, the following holds:*

$$F \mid\!\sim_B^{s,A} G \textit{ iff } \langle\!\langle F \rangle\!\rangle_{A,B}^Y \subseteq \langle\!\langle G \rangle\!\rangle_{A,B}^Y, \textit{ for any } Y \subseteq A \cup B$$

We will need the following auxiliary lemmas in order to prove the above theorem.

Lemma 6 (Lemma 6 from [20])**.** *Let $A, S \subseteq At$ be sets of atoms and let L be a fork such that $At(L) \subseteq A$. Then, any $\mathcal{H}$ maximal in $\langle\!\langle L \rangle\!\rangle^S$ is A*-respectful.

Lemma 7 (Lemma 18 from [20])**.** *Let $A, S, S' \subseteq At$ be sets of atoms such that $S \cap A = S' \cap A$ and let F be a fork such that $At(F) \subseteq A$. 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 $\mathcal{H}_A \preceq \mathcal{H}'_A$.*

Proof of Theorem 4. See Appendix A. □

Corollary 1. *Given the two forks and sets of F and G and sets $A, B \subseteq At$, the following relation holds:*

$$F \equiv_B^{s,A} G \textit{ iff } \langle\!\langle F \rangle\!\rangle_{A,B}^Y = \langle\!\langle G \rangle\!\rangle_{A,B}^Y, \textit{ for any } Y \subseteq A \cup B$$

7. An Example Case: Reasoning about Policies

To illustrate briefly the practical relevance of our intertheory relations, let us consider the case of rule-based *policies*. In particular, ASP is well suited to represent defaults, typicalities, and exceptions that may be involved in policy formulations. One policy area where logical approaches have been employed with success is in the domain of security and access control. This area has been active for some time, as far back as [29]. Later works include [30–34]. Bonatti [31] has surveyed the area and suggested several reasoning problems that can be studied in languages such as Datalog and answer set programming. We loosely follow his approach and extend it somewhat.

In the case of access control, let us suppose there is a logic program Π expressing the basic policy in the form of a set of rules. Let us say it might express conditions for accessing certain restricted Web pages in the University of South Wolverhampton. In addition, there are contexts $\mathcal{C}$ that express additional facts that hold at some times; perhaps this particular Web area allows different types of access in different periods. Then, there are credentials $\mathcal{D}$ that are also (atomic) facts. Let us say that, in general, only faculty members are allowed to access the restricted area, so a credential might be $faculty\_member(Pedro)$. Completing the picture, there are authorisations $\mathcal{A}$, usually statements saying whether a subject can/cannot perform the operation on the object—for instance, whether Pedro can access the Web area. It may be 2- or 3-valued, depending on the context.

Using a logic-based language such as ASP allows us to analyse in a straightforward manner different kinds of reasoning problems that may arise. As Bonatti observes, the most basic problem is one of *entailment*. Is an authorisation φ granted by Π and $\mathcal{C}$? This is the case if $\Pi \cup \mathcal{C} \models \varphi$, where $\models$ is a suitable nonmonotonic inference relation, such as the relation $\models_s$ associated with stable model semantics.

The second problem is in fact an *abductive*, *satisfaction* problem. Roughly speaking, given an authorisation request, the problem is to deliver a set of conditions (credentials) that are sufficient to answer the authorisation positively, if such a set exists. Thus, given a set $\mathcal{D}$ of digital credentials, the abduction problem is to find a subset $\mathcal{D}' \subseteq \mathcal{D}$ of credentials for a given authorisation φ and context C, such that

$$\Pi \cup \mathcal{D}' \cup C \models_s \varphi$$

A solution to the abduction problem can provide a suitable *explanation*. Suppose that Pedro has only recently joined the faculty but is not yet registered in the appropriate database. He is denied access with the explanation that a registering process is required first, i.e., he is informed of a missing credential that will grant him access.

Third, there is the *conservative extension* problem. Suppose that conditions have changed and now a new type of user may be admitted, e.g., some students can now access the area providing they belong to a specific committee. The program Π is enlarged to a new program, Π', specifying the new conditions. The context and the set of credentials is also enlarged. However, we want to be sure that all authorisations that were valid previously continue to hold in the new situation and also that no loopholes in the system have been created that would allow unintended authorisations that were previously barred. In other words, the new program *conservatively* extends the previous one.

Fourth, there is a related problem of *relative strength*. We can say that in a given context $\mathcal{C}$, a policy Π is at least as strong as Π' if every authorisation request accepted by Π is also accepted by Π'. So, if Π' rejects authorisation A, then so does Π. For simplicity, let us suppress contexts for the moment and consider a *policy framework* $\mathcal{P}$ to be a triple $(\Pi, \mathcal{D}, \mathcal{A})$, where Π is a theory, possibly in the form of a set of program rules in language $\mathcal{L}$, $\mathcal{D}$ is a set of credentials, comprising certain atomic sentences of $\mathcal{L}$, and $\mathcal{A}$ are authorisations. Let $\mathcal{P}_1 = (\Pi_1, \mathcal{D}, \mathcal{A})$, and $\mathcal{P}_2 = (\Pi_2, \mathcal{D}, \mathcal{A})$ be policy frameworks. Then, we can say that $\mathcal{P}_1$ is at least as strong as $\mathcal{P}_2$ if for any $\varphi \in \mathcal{A}$, and $D \subseteq \mathcal{D}$:

$$\Pi_2 \cup D \not\models_s A \Rightarrow \Pi_1 \cup D \not\models_s A \quad (4)$$

Fifth, there is the problem of policy *equivalence* which may come in different degrees. Two policies that admit exactly the same authorisations and rejections in a given context can be said to be equivalent in that context. A stronger property is that they are equivalent in all contexts. Furthermore, a still stronger property is that they remain equivalent when they are extended by adding new policy rules.

Inter-Policy Relations

ASP provides a suitable framework for studying these kinds of reasoning problems. Aside from being able to deal with issues of entailment, abduction, consistency, and completeness, the logical approach is well adapted to handle the inter-policy relations described above. Weak and strong forms of *entailment* between programs are relevant for capturing the relation express by (4). For example, a sufficient condition for the relation to obtain is that for any $D \subseteq \mathcal{D}$, $\Pi_2 \cup D$ weakly entails $\Pi_1 \cup D$ in the sense of Definition 9. This also means that the relation holds if Π_2 strongly entails Π_1. However, to characterise this notion precisely we can use the notion of relativised uniform entailment and consider projections onto the authorisations $\mathcal{A}$.

We can say that two access policies covering the same credentials and authorisations, $\mathcal{P}_1 = (\Pi_1, \mathcal{D}, \mathcal{A})$, and $\mathcal{P}_2 = (\Pi_2, \mathcal{D}, \mathcal{A})$, are *equivalent* if they generate the same authorisations, and *strongly equivalent* if they are equivalent when expanded by any new set of policy rules Π. If Π_1 and Π_2 are relativised uniform equivalent with respect to $\mathcal{D}$, then $\mathcal{P}_1$ and $\mathcal{P}_2$ are strongly equivalent. To consider the converse relation, because we only require policies to deliver the same authorisations, they only need to be equivalent when projected onto $\mathcal{A}$. So, in this case we are interested in uniform or relativised uniform equivalence with projection.

8. Conclusions

We defined a selection of correspondence relations between equilibrium theories and answer set programs. They are based on the inferential capabilities of theories, i.e., how they answer queries and derive formulas, rather than on their sets of stable models. This is important for many applications of ASP. In particular, by including relativised and projective correspondences, we cover many cases that arise in practical applications of ASP. Not only is theory equivalence of interest but also entailment (and consequence) relations between theories.

We have shown that these new relations, including both equivalence and consequence relations, are actually for the most part equivalent to the standard types of correspondence defined in terms of stable and equilibrium models that have been studied in the past. Finite theories and programs that are not equivalent or not in an entailment relation can be separated by queries of the following kind: conjunctions of literals in the case of credulous inference and disjunctions of literals in the case of sceptical inference.

What this implies is that the large body of known results that characterise intertheory relations in ASP, as well as the accompanying techniques for deciding whether these relations obtain in practice, are directly applicable to the types of relations defined here. These results include the original characterisation of strong equivalence for programs and theories in the logic **HT** [2] and the studies of uniform equivalence in terms of **HT** models [4,27] and in terms of **HT** countermodels in [26]. Relativised equivalence was treated in [5] and more general correspondences including projection in [3] and [28]. Woltran's work in [5] was extended in [35] to cover general propositional theories. For an extensive bibliography of further work on program correspondences, see especially [28].

In Section 6, we extended previous work on projective relations [3,28] in two respects. First, building on [20], we applied the notions of T-support and T-views to give alternative characterisations of projective entailment and equivalence that are now extended to general propositional theories in equilibrium logic. Secondly, we extended previous work on forks to yield a more general characterisation of projective entailment between forks.

There are many open challenges left for the future. For example, it remains to be investigated how these new intertheory relations generalise to the case of first-order theories and programs with variables. Already strong equivalence for first-order theories was characterised in a quantified version of **HT** in [36] and uniform equivalence was treated in [26]. More recently, [37] studied relativised and projective versions of equivalence for non-ground programs.

Another avenue for study would be to extend the present framework to accommodate infinite languages and theories. In the case of credulous consequence, for instance, it seems that finite queries will separate non-equivalent theories in many cases. Consider the formula $\bigwedge \varphi_i$ in the proof of Lemma 1(i). This formula would become an infinite conjunction in the case of infinite theories, but for each $M_i \in E(\Pi_2)$, there is some l_i from $\bigwedge \varphi_i$ that is false in M_i. So, if the set of equilibrium models of Π_2 is finite, we can build a finite conjunction of literals false in each equilibrium model M_i of Π_2 but true in an equilibrium model M of Π_1. So, there is a finite query that can separate the two theories.

Author Contributions: J.F., D.P., C.V. and S.W. have contributed substantially and in equal measure to the work reported here. This applies in particular to the conceptualisation, methodology, formal analysis, and writing. The final editing was supervised by D.P. All authors have read and agreed to the published version of the manuscript.

Funding: This research has received partial support from the European Cooperation in Science & Technology (COST) Action CA17124. The third author acknowledges the funding of project PID 2020-116201GB-I00 (Ministerio de Ciencia e Innovación, Spain) and also the financial support supplied by the Consellería de Educación, Universidade e Formación Profesional (accreditations GPC ED431B 2022/23 and 2019–2022 ED431G-2019/01). The last author has been supported by the Austrian Science Fund (FWF) grant Y698.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable

Data Availability Statement: Not applicable.

Acknowledgments: We are grateful for the advice and suggestions made by the anonymous referees of this article.

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

The following abbreviations are used in this manuscript:

ASP	Answer Set Programming
HT	the logic of here-and-there
PSE	projective strong equivalence

Appendix A. Proofs of Results

Proof of Proposition 9. (Right to left). Clearly, if, for any set of A formulas Π, $E(\Pi_1 \cup \Pi)|B = E(\Pi_2 \cup \Pi)|B$, then these theories answer all B-queries in the same way and so are strongly inseparable.

(Left to right). Suppose that for some Π, $E(\Pi_1 \cup \Pi)|B \neq E(\Pi_2 \cup \Pi)|B$. Then, we can apply Lemma 1(i) to some $M|B$, say, in $E(\Pi_1 \cup \Pi)|B \setminus E(\Pi_2 \cup \Pi)|B$ and build a B-sentence $\bigwedge_i \varphi_i$ that is true in $M|B$ but false in all models in $E(\Pi_2 \cup \Pi)|B$. Note that the formulas φ_i are built as in Lemma 1(i) where now $U = B$. Likewise for Lemma 1(ii). So, the theories are not strongly B, ω inseparable relative to A, for $\omega \in \{c, s\}$. □

Proof of Proposition 10. (Right to left). Suppose that $\Pi_1 \mid\!\sim_B^{s,A} \Pi_2$ does not hold and choose a model $M \in E(\Pi_1 \cup \Sigma)|B \setminus E(\Pi_2 \cup \Pi)|B$. Apply Lemma 1(ii), setting $U = B$, to obtain a B-formula ψ satisfied by all models in $E(\Pi_2 \cup \Sigma)|B$ but not in M. It follows that

$$\Pi_1 \not\mid\!\sim_{s,B}^{s,A} \Pi_2$$

□

The following lemma will be useful later.

Lemma A1. *Given a program Π and sets $\mathcal{L}, Z \subseteq At$, such that $Z \models \Pi$, for any $\Sigma \in \mathcal{L}$ the following assertions are equivalent:*

1. $Z \in E(\Pi \cup \Sigma)$
2. $[\![\Pi]\!]^Z$ *is A-feasible and*

$$[\![\Pi]\!]_A^Z \cap [\![\Sigma]\!]^{Z \cap A} = [Z \cap A].$$

Proof. 1. $\Rightarrow$ 2.

Take $Z \in E(\Pi \cup \Sigma)$ and suppose that $\langle H, Z\rangle \models \Pi$ for some $H \subseteq Z$ such that $H \cap A = Z \cap A$. Because

$$\langle H, Z\rangle \models \Sigma \text{ iff } \langle H \cap A, Z \cap A\rangle \models \Sigma \text{ iff } \langle Z \cap A, Z \cap A\rangle \models \Sigma \text{ iff } \langle Z, Z\rangle \models \Sigma$$

we have that $\langle H, Z\rangle \models \Pi \cup \Sigma$ which implies that $H = Z$. Moreover, if $X \subseteq Z$ satisfies $\langle X, Z\rangle \models \Pi$ and $\langle X \cap A, Z \cap A\rangle \models \Sigma$, then $\langle X, Z\rangle \models \Pi \cup \Sigma$ and $X = Z$. Consequently, $X \cap A = Z \cap A$.

2. $\Rightarrow$ 1.

Suppose that $\langle H, Z\rangle \models \Pi \cup \Sigma$ for some $H \subseteq Z$. Then, $\langle H \cap A, Z \cap A\rangle \models \Sigma$ and $H \cap A \in [\![\Pi]\!]^Z_A \cap [\![\Sigma]\!]^{Z \cap A}$. Consequently, $H \cap A = Z \cap A$ and $H = Z$ because $[\![\Pi]\!]^Z$ is A-feasible. □

Proof of Theorem 1. (Right to left) Suppose that $\tilde{Z} = Z \cap B \in E(\Pi_1 \cup \Sigma)|B$ for some $\Sigma \in \mathcal{L}_A$ and $Z \in E(\Pi_1 \cup \Sigma)$. By Lemma A1, we know that $[\![\Pi_1]\!]^Z$ is A-feasible and

$$[\![\Pi_1]\!]^Z_A \cap [\![\Sigma]\!]^{Z \cap A} = [Z \cap A].$$

Take $T = Z \cap (A \cup B)$, then $[\![\Pi_1]\!]^Z_A \in \langle\!\langle \Pi_1 \rangle\!\rangle^T_{A,B} \subseteq \langle\!\langle \Pi_2 \rangle\!\rangle^T_{A,B}$.

It follows that there is some $Z_2 \subseteq At$ such that $T = Z_2 \cap (A \cup B)$ (which implies that $Z_2 \cap A = T \cap A = Z \cap A$ and $Z_2 \cap B = T \cap B = Z \cap B = \tilde{Z}$) being $[\![\Pi_2]\!]^{Z_2}$ A-feasible and $[\![\Pi_2]\!]^{Z_2}_A \subseteq [\![\Pi_1]\!]^Z_A$ (or $[\![\Pi_1]\!]^Z_A \preceq [\![\Pi_2]\!]^{Z_2}_A$). Notice that $Z_2 \models \Sigma$ because $Z_2 \cap A = Z \cap A$ and $Z \models \Sigma$. Moreover:

$$[\![\Pi_2]\!]^{Z_2}_A \cap [\![\Sigma]\!]^{Z_2 \cap A} \subseteq [\![\Pi_1]\!]^Z_A \cap [\![\Sigma]\!]^{Z \cap A} = [Z \cap A] = [Z_2 \cap A]$$

This implies that $Z_2 \in E(\Pi_2 \cup \Sigma)$ and $\tilde{Z} = Z_2 \cap B \in E(\Pi_2 \cup \Sigma)|B$.

(Left to right)

Take $T \subseteq A \cup B$ and $[\![\Pi_1]\!]^Z_A \in \langle\!\langle \Pi_1 \rangle\!\rangle^T_{A,B}$, for some $Z \cap (A \cup B) = T$ such that $[\,] \neq [\![\Pi_1]\!]^Z$ is Afeasible. Let us denote by

$$D(Z) = \{Z' \subseteq At \mid Z' \cap (A \cup B) = Z \cap (A \cup B) \text{ and } [\,] \neq [\![\Pi_2]\!]^{Z'} \text{ is } A\| - \|feasible\}$$

Take

$$\mathcal{S} = \{(X', Z') \mid Z' \in D(Z) \text{ and } X' \cap A \in [\![\Pi_2]\!]^{Z'}_A \cap \overline{[\![\Pi_1]\!]^Z_A}\} \cup \{(X, Z) \mid X \cap A \in \overline{[\![\Pi_1]\!]^Z_A}\}$$

Notice that $(Z, Z) \in \mathcal{S}$ and $(X', Z') \in \mathcal{S}$ implies $(Z', Z') \in \mathcal{S}$. Then, $\mathcal{S}$ is a total-closed set of interpretations (A set $\mathcal{S}$ of **HT** interpretations is total-closed if for any $\langle H, T\rangle \in \mathcal{S}$, $\langle T, T\rangle \in \mathcal{S}$). Denote by Π' the program with signature in A such that models of Π' correspond to interpretations in $\mathcal{S}|A$.

First of all, we have that $Z \in E(\Pi_1 \cup \Pi')$ because $(Z \cap A, Z \cap A) \in \mathcal{S}|A$, so $Z \cap A \models \Pi'$ or $Z \models \Pi'$. Moreover, take $X \subseteq Z$ such that $\langle X, Z\rangle \models \Pi_1 \cup \Pi'$. Then, $\langle X \cap A, Z \cap A\rangle \models \Pi'$ so $X \cap A \in \overline{[\![\Pi_1]\!]^Z_A} \cap [\![\Pi_1]\!]^Z_A$ which implies that $X \cap A = Z \cap A$ and $X = Z$ because $[\![\Pi_1]\!]^Z$ is A-feasible.

Now $T \cap B = Z \cap B \in E(\Pi_1 \cup \Pi')|B \subseteq E(\Pi_2 \cup \Pi')|B$, so there exists $\tilde{Z} \in E(\Pi_2 \cup \Pi')$ such that $\tilde{Z} \cap B = Z \cap B$. Because $\tilde{Z} \models \Pi'$, we know that $(\tilde{Z}, \tilde{Z}) \in \mathcal{S}$ so $\tilde{Z} \in D(Z)$. We are going to show that $[\![\Pi_2]\!]^{\tilde{Z}}_A \subseteq [\![\Pi_1]\!]^Z_A$ or equivalently $[\![\Pi_1]\!]^Z_A \preceq [\![\Pi_2]\!]^{\tilde{Z}}_A \in \langle\!\langle \Pi_2 \rangle\!\rangle^T_{A,B}$. On the contrary, suppose that $X' \cap A \notin [\![\Pi_1]\!]^Z_A$ with $X' \in [\![\Pi_2]\!]^{\tilde{Z}}$. Then ,$X' \cap A \in [\![\Pi_2]\!]^{\tilde{Z}}_A \cap \overline{[\![\Pi_1]\!]^Z_A}$ so $(X' \cap A, \tilde{Z} \cap A) \in \mathcal{S}|A$. Because: $\langle X', \tilde{Z}\rangle \models \Pi_2 \cup \Pi'$, we can deduce that $X' = \tilde{Z}$ because $\tilde{Z} \in E(\Pi_2 \cup \Pi')$. Finally, $X' \cap A = \tilde{Z} \cap A = Z \cap A \in [\![\Pi_1]\!]^Z_A$. □

Proof of Theorem 4. (Left to right)

Take $Y \subseteq A \cup B$ and $\mathcal{H}_A \in \langle\!\langle F \rangle\!\rangle^Y_{A,B}$, where $\mathcal{H} \in \langle\!\langle F \rangle\!\rangle^Z$ is A-feasible and $Z \cap (A \cup B) = Y$.

We will denote by

$$D(Y) = \{Z' \subseteq At \mid Z' \cap (A \cup B) = Y\}.$$

Let us consider the following $Y \cap A$-support $\mathcal{H}_0$:

$$\mathcal{H}_0 = \overline{\mathcal{H}_A} \cap \bigcup\{\mathcal{H}'_A \mid \mathcal{H}' \in \langle\!\langle\, G \,\rangle\!\rangle^{Z'} \text{ with } \mathcal{H}' A- \text{ feasible and } Z' \in D(Y)\}$$

By Lemma 3, we know that there exists a formula $\varphi \in \mathcal{L}_A$ such that $[\![\, \varphi \,]\!]^{Y \cap A} = \mathcal{H}_0$ and $[\![\, \varphi \,]\!]^T = [\,]$, for any $T \subseteq A$ with $T \neq Y \cap A$.

Let us prove that $Z \in E(F \wedge \varphi)$ by showing that:

$$[Z] = \mathcal{H} \cap [\![\, \varphi \,]\!]^Z \in \langle\!\langle\, F \wedge \varphi \,\rangle\!\rangle^Z.$$

Suppose that $H \in \mathcal{H} \cap [\![\, \varphi \,]\!]^Z$. Then:

$$\begin{array}{ll} H \cap A \in \mathcal{H}_A \cap [\![\, \varphi \,]\!]^Z_A & \text{implies} \\ H \cap A \in \mathcal{H}_A \cap \mathcal{H}_0 & \text{implies} \\ H \cap A \in \mathcal{H}_A \cap \overline{\mathcal{H}_A} & \text{implies} \\ H \cap A = Z \cap A & \text{implies} \\ H = Z & \end{array}$$

Consequently $Y \cap B = Z \cap B \in E(F \wedge \varphi)|B \subseteq E(G \wedge \varphi)|B$, so there exists $\tilde{Z} \in E(G \wedge \varphi)$ verifying that $Z \cap B = \tilde{Z} \cap B$. Notice that $[\tilde{Z}] \in \langle\!\langle\, G \wedge \varphi \,\rangle\!\rangle^{\tilde{Z}}$, so

$$[\tilde{Z}] = \mathcal{H}_1 \cap [\![\, \varphi \,]\!]^{\tilde{Z}}$$

for some $\tilde{Z}$-support $\mathcal{H}_1 \in \langle\!\langle\, G \,\rangle\!\rangle^{\tilde{Z}}$. This implies that

$$\tilde{Z} \cap A \in (\mathcal{H}_1)_A \cap [\![\, \varphi \,]\!]^{\tilde{Z}}_A \subseteq [\![\, \varphi \,]\!]^{\tilde{Z} \cap A}.$$

Then, $\tilde{Z} \cap A = Y \cap A$ and so $\tilde{Z} \in D(Y)$. Now we will show that $\mathcal{H}_1$ is A-feasible. Suppose that $H \in \mathcal{H}_1$ satisfies $H \cap A = \tilde{Z} \cap A$. Then, $H \in [\![\, \varphi \,]\!]^{\tilde{Z}}$ because $\varphi \in \mathcal{L}_A$ and $\tilde{Z} \cap A \in [\![\, \varphi \,]\!]^{\tilde{Z}}_A = [\![\, \varphi \,]\!]^{\tilde{Z} \cap A}$. Then, $H \in \mathcal{H}_1 \cap [\![\, \varphi \,]\!]^{\tilde{Z}} = [\tilde{Z}]$.

Finally, suppose that $(\mathcal{H}_1)_A \not\subseteq \mathcal{H}_A$, so there exists $H \in \mathcal{H}_1$ such that $H \cap A \neq Z \cap A$ and $H \cap A \in (\mathcal{H}_1)_A \cap \overline{\mathcal{H}_A} \subseteq \mathcal{H}_0$. This would imply that $H \in \mathcal{H}_1 \cap [\![\, \varphi \,]\!]^{\tilde{Z}}$, so $H = \tilde{Z}$ which is a contradiction.

(Right to left)

Suppose that $T \in E(F \wedge L)|B$ for some fork $L \in \mathcal{L}_A$. Then, $T = Z \cap B$ with $Z \in E(F \wedge L)$. Take $Y := Z \cap (A \cup B)$. Because:

$$[Z] = \mathcal{H} \cap \mathcal{H}'$$

for some $\mathcal{H}$ in $\langle\!\langle\, F \,\rangle\!\rangle^Z$ and $\mathcal{H}' \in \langle\!\langle\, L \,\rangle\!\rangle^Z$, which we can suppose maximal in $\langle\!\langle\, L \,\rangle\!\rangle^Z$ and then A-respectful by Lemma 6, we can say that $\mathcal{H}$ is A-feasible because if $H \in \mathcal{H}$ satisfies $H \cap A = Z \cap A$; then, $H \in \mathcal{H}' \cap \mathcal{H}$ and $H = Z$. This implies that $\mathcal{H}_A \in \langle\!\langle\, F \,\rangle\!\rangle^Y_{A,B} \subseteq \langle\!\langle\, G \,\rangle\!\rangle^Y_{A,B}$, so there exists $Z' \in D(Y)$ and $\mathcal{H}_1 \in \langle\!\langle\, G \,\rangle\!\rangle^{Z'}$ being A-feasible such that $\mathcal{H}_A \preceq (\mathcal{H}_1)_A$ or $(\mathcal{H}_1)_A \subseteq \mathcal{H}_A$. We can apply Lemma 7 to deduce that there exists $\mathcal{H}'' \in \langle\!\langle\, L \,\rangle\!\rangle^{Z'}$ such that $\mathcal{H}'_A \preceq \mathcal{H}''_A$.

In order to finish the proof, we only have to show that $Z' \in E(G \wedge L)$ or, equivalently that:

$$[Z'] = \mathcal{H}_1 \cap \mathcal{H}'' \in \langle\!\langle\, G \wedge L \,\rangle\!\rangle^{Z'}$$

Take $H \in \mathcal{H}_1 \cap \mathcal{H}''$. Then:

$$\begin{array}{ll} H \cap A \in (\mathcal{H}_1 \cap \mathcal{H}'')_A & \text{implies} \\ H \cap A \in (\mathcal{H}_1)_A \cap (\mathcal{H}'')_A & \text{implies} \\ H \cap A \in \mathcal{H}_A \cap (\mathcal{H}')_A & \text{implies} \\ H \cap A \in (\mathcal{H} \cap \mathcal{H}')_A & \text{implies} \\ H \cap A = Z \cap A = Z' \cap A & \text{implies} \\ H = Z' & \end{array}$$

Notice that we used Lemma 6. □

References

1. Pearce, D. Equilibrium logic. *Ann. Math. Artif. Intell.* **2006**, *47*, 3–41. [CrossRef]
2. Lifschitz, V.; Pearce, D.; Valverde, A. Strongly equivalent logic programs. *ACM Trans. Comput. Log.* **2001**, *2*, 526–541. [CrossRef]
3. Eiter, T.; Tompits, H.; Woltran, S. On solution correspondences in answer-set programming. In Proceedings of the Nineteenth International Joint Conference on Artificial Intelligence (IJCAI'05), Edinburgh, UK, 30 July–5 August 2005; Kaelbling, L.P., Saffiotti, A., Eds.; Professional Book Center: London, UK, 2005; pp. 97–102.
4. Eiter, T.; Fink, M. Uniform equivalence of logic programs under the stable model semantics. In *Logic Programming, Proceedings of the 19th International Conference, ICLP 2003, Mumbai, India, 9–13. December 2003*; Palamidessi, C., Ed.; Lecture Notes in Computer Science; Springer: Berlin/Heidelberg, Germany, 2003; Volume 2916, pp. 224–238.
5. Woltran, S. Characterizations for relativized notions of equivalence in answer set programming. In *Logics in Artificial Intelligence, Proceedings of the 9th European Conference, JELIA 2004, Lisbon, Portugal, 27–30 September 2004*; Alferes, J.J., Leite, J.A., Eds.; Lecture Notes in Computer Science; Springer: Berlin/Heidelberg, Germany, 2004; Volume 3229, pp. 161–173.
6. Maher, M. Equivalences of Logic Programs. In *Foundations of Deductive Databases and Logic Programming*; Minker, J., Ed.; Morgan Kaufmann: Burlington, MA, UDA, 1988; pp. 627–658.
7. Sagiv, Y. Optimising DATALOG Programs. In *Foundations of Deductive Databases and Logic Programming*; Minker, J., Ed.; Morgan Kaufmann: Burlington, MA, UDA, 1988; pp. 659–698
8. Truszczynski, M. Strong and uniform equivalence of nonmonotonic theories—An algebraic approach. *Ann. Math. Artif. Intell.* **2006**, *48*, 245–265. [CrossRef]
9. Turner, H. Strong equivalence for causal theories. In Proceedings of the 7th International Conference on Logic Programming and Nonmonotonic Reasoning (LPNMR'04), Fort Lauderdale, FL, USA, 6–8 January 2004; Lifschitz, V., Niemelä, I., Eds.; Lecture Notes in Computer Science; Springer: Berlin/Heidelberg, Germany, 2004; Volume 2923, pp. 289–301.
10. Baumann, R. Normal and strong expansion equivalence for argumentation frameworks. *Artif. Intell.* **2012**, *193*, 18–44. [CrossRef]
11. Oikarinen, E.; Woltran, S. Characterizing strong equivalence for argumentation frameworks. *Artif. Intell.* **2011**, *175*, 1985–2009. [CrossRef]
12. Bernreiter, M.; Maly, J.; Woltran, S. Choice logics and their computational properties. In Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, Virtual, 19–27 August 2021; pp. 1794–1800.
13. Faber, W.; Konczak, K. Strong equivalence for logic programs with preferences. In Proceedings of the Nineteenth International Joint Conference on Artificial Intelligence, Edinburgh, UK, 30 July–5 August 2005; Professional Book Center: London, UK, 2005; pp. 430–435.
14. Faber, W.; Truszczynski, M.; Woltran, S. Abstract preference frameworks—A unifying perspective on separability and strong equivalence. In Proceedings of the Twenty-Seventh AAAI Conference on Artificial Intelligence, Bellevue, WA, USA, 14–18 July 2013; des Jardins, M., Littman, M.L., Eds.; AAAI Press: Palo Alto, CA, USA, 2013; pp. 297–303.
15. Eiter, T.; Fink, M.; Tompits, H.; Woltran, S. Simplifying logic programs under uniform and strong equivalence. In Proceedings of the LPNMR 2004, Fort Lauderdale, FL, USA, 6–8 January 2004; Lifschitz, V., Niemelä, I. Eds.; Springer: Berlin/Heidelberg, Germany, 2004; Volume 2923, pp. 87–99.
16. Gonçalves, R.; Knorr, M.; Leite, J. You can't always forget what you want: On the limits of forgetting in answer set programming. In Proceedings of the ECAI 2016, The Hague, The Netherlands, 29 August–2 September 2016; Fox, M.S., Kaminka, G.A., Eds.; IOS Press: Amsterdam, The Netherlands, 2016.
17. Shmueli, O. Decidability and Expressiveness Aspects of Logic Queries. In Proceedings of the 6th ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems (PODS'87), San Diego, CA, USA, 23–25 March 1987; ACM Press: New York, NY, USA, 1987; pp. 237–249.
18. Botoeva, E.; Lutz, C.; Ryzhikov, V.; Wolter, F.; Zakharyaschev, M. Query inseparability for ALC ontologies. *Artif. Intell.* **2019**, *272*, 1–51. [CrossRef]
19. Aguado, F.; Cabalar, P.; Pearce, D.; Pérez, G.; Vidal, C. A denotational semantics for equilibrium logic. *Theory Pract. Log. Program.* **2015**, *15*, 620–634. [CrossRef]
20. Aguado, F.; Cabalar, P.; Fandinno, J.; Pearce, D.; Pérez, G.; Vidal, C. Forgetting auxiliary atoms in forks. *Artif. Intell.* **2019**, *275*, 575–601. [CrossRef]
21. Heyting, A. Die formalen Regeln der intuitionistischen Logik. *Sitzungsberichte der Preussischen Akademie der Wissenschaften zu Berlin* **1930**, *16*, 42–56.
22. Cabalar, P.; Pearce, D.; Valverde, A. Stable reasoning. *J. Appl. Non Class. Logics* **2017**, *27*, 238–254. [CrossRef]

23. van Dalen, D. *Logic and Structure*; Springer: Berlin/Heidelberg, Germany, 2013.
24. Pearce, D. A new logical characterisation of stable models and answer sets. In *Non-Monotonic Extensions of Logic Programming, Proceedings of the NMELP '96, Bad Honnef, Germany, 5–6 September 1996*; Dix, J., Pereira, L.M., Przymusinski, T.C., Eds.; Lecture Notes in Computer Science; Springer: Berlin/Heidelberg, Germany, 1996; Volume 1216, pp. 57–70.
25. Gelfond, M.; Lifschitz, V. The stable models semantics for logic programming. In Proceedings of the 5th International Conference on Logic Programming, Seattle, WA, USA, 15–19 August 1988; Kowalski, R., Bowen, A., Eds.; MIT Press: Cambridge, MA, USA, 1988; pp. 1070–1108.
26. Fink, M. Equivalences in answer-set programming by countermodels in the logic of here-and-there. In *Logic Programming, Proceedings of the 24th International Conference, ICLP 2008, Udine, Italy, 9–13 December 2008*; Banda, M.G.d., Pontelli, E., Eds.; Lecture Notes in Computer Science; Springer: Berlin/Heidelberg, Germany, 2008; Volume 5366, pp. 99–113.
27. Pearce, D.; Valverde, A. Uniform equivalence for equilibrium logic and logic programs. In *Logic Programming and Nonmonotonic Reasoning, Proceedings of the 7th International Conference, LPNMR 2004, Fort Lauderdale, FL, USA, 6–8 January 2004*; Lifschitz, V., Niemelä, I., Eds.; Lecture Notes in Computer Science; Springer: Berlin/Heidelberg, Germany, 2004; Volume 2923, pp. 194–206.
28. Oetsch, J.; Seidl, M.; Tompits, H.; Woltran, S. Beyond uniform equivalence between answer-set programs. *ACM Trans. Comput. Log.* **2021**, *22*, 2:1–2:46. [CrossRef]
29. Woo, T.Y.C.; Lam, S.S. Authorization in distributed systems: A new approach. *J. Comput. Secur.* **1993**, *2*, 107–136. [CrossRef]
30. Bonatti, P.A.; Samarati, P. Logics for authorization and security. In *Logics for Emerging Applications of Databases*; Chomicki, J., Meyden, R.V., Saake, G., Eds.; Springer: Berlin/Heidelberg, Germany, 2003; pp. 277–323.
31. Bonatti, P.A. Datalog for security, privacy and trust. In *Datalog Reloaded*; de Moor, O., Gottlob, G., Furche, T., Sellers, A., Eds.; Springer: Berlin/Heidelberg, Germany, 2011; pp. 21–36.
32. Bonatti, P.A. Logic-based authorization languages. In *Encyclopedia of Cryptography and Security*, 2nd ed.; Henk, C., van Tilborg, A., Jajodia, S., Eds.; Springer: Berlin/Heidelberg, Germany, 2011; pp. 734– 736.
33. Craven, R.; Lobo, J.; Ma, J.; Russo, A.; Lupu, E.C.; Bandara, A.K. Expressive policy analysis with enhanced system dynamicity. In Proceedings of the 2009 ACM Symposium on Information, Computer and Communications Security, ASIACCS 2009, Sydney, Australia, 10–12 March 2009; Li, W., Susilo, W., Tupakula, U.K., Safavi-Naini, R., Varadharajan, V., Eds.; ACM: New York, NY, USA, 2009; pp. 239–250.
34. Gelfond, M.; Lobo, J. Authorization and obligation policies in dynamic systems. In *Logic Programming, Proceedings of the 24th International Conference, ICLP 2008, Udine, Italy, 9–13 December2008, Proceedings*; Banda, M.G.d., Pontelli, E., Eds.; Lecture Notes in Computer Science; Springer: Berlin/Heidelberg, Germany, 2008; Volume 5366, pp. 22–36.
35. Pearce, D.; Tompits, H.; Woltran, S. Relativised equivalence in equilibrium logic and its applications to prediction and explanation: Preliminary report. In Proceedings of the LPNMR'07 Workshop on Correspondence and Equivalence for Nonmonotonic Theories (CENT2007), Tempe, AZ, USA, 14 May 2007; Volume 265.
36. Lifschitz, V.; Pearce, D.; Valverde, A. A characterization of strong equivalence for logic programs with variables. In *Logic Programming and Nonmonotonic Reasoning, Proceedings of the 9th International Conference, LPNMR 2007, Tempe, AZ, USA, 15–17 May 2007*; Baral, C., Brewka, G., Schlipf, J.S., Eds.; Lecture Notes in Computer Science; Springer: Berlin/Heidelberg, Germany, 2007; Volume 4483, pp. 188–200.
37. Geibinger, T.; Tompits, H. Characterising relativised strong equivalence with projection for non-ground answer-set programs. In *Logics in Artificial Intelligence, Proceedings of the 16th European Conference, JELIA 2019, Rende, Italy, 7–11 May 2019*; Calimeri, F., Leone, N., Manna, M., Eds.; of *Lecture Notes in Computer Science*; Springer: Berlin/Heidelberg, Germany, 2019; Volume 11468, pp. 542–558.