Synonymous Theories in Answer Set Programming and Equilibrium Logic

David Pearce[1] and **Agustín Valverde**[2]

Abstract. The study of strong equivalence between logic programs or nonmonotonic theories under answer set semantics, begun in [18], is extended to the case where the programs or theories concerned are formulated in different languages. We suggest that theories in different languages be considered equivalent in the strong sense or *synonymous* if and only if each is bijectively interpretable (hence translatable) into the other. Since the logic of here-and-there, which provides a suitable foundation for answer set programming, has the Beth property, we can easily give model-theoretic conditions that are equivalent to bijective interpretability. These conditions involve mappings between the models of the two theories that, in particular, preserve the property of being an answer set or equilibrium model.

1 Introduction

With the emergence of answer set solvers such as DLV [17], GnT [15], and smodels [30], answer set programming (ASP) now provides a practical and viable environment for tasks of knowledge representation and declarative problem solving. AI applications include planning and diagnosis, as exemplified in a prototype decision support system for the space shuttle [2], the management of heterogenous data in information systems, as performed in the INFOMIX project,[3] the representation of ontologies in the semantic web allowing for default knowledge and inference, as discussed in [5], as well as compact and fully declarative representations of hard combinatorial problems such as n-Queens, Hamiltonian paths, and so on[4]. In all these areas it may be important to know when different logic programs representing a given problem or state of affairs are equivalent and lead to essentially the same solutions (answer sets). Very often one would like to know that the equivalence is also robust, since two programs may have the same answer sets yet behave very differently once they are embedded in some larger context. For a robust or modular notion of equivalence one should require that programs behave similarly when extended by any further programs. This leads to the following concept of *strong* equivalence: programs Π_1 and Π_2 are strongly equivalent, in symbols $\Pi \equiv_s \Pi_2$, if and only if for any Σ, $\Pi_1 \cup \Sigma$ is equivalent to (has the same answer sets as) $\Pi_2 \cup \Sigma$. The concept of strong equivalence for logic programs in ASP was introduced and studied in [18] and has given rise to a substantial body of further work looking at different characterisations [13, 31], new variations and applications of the idea [7, 27, 32], as well as developing systems to test for strong equivalence [27, 8].

Currently, however, the concept of strong equivalence (and variants) has a rather severe limitation: it tacitly assumes that the programs being compared are formulated in the same language (vocabulary), or at least that any differences of vocabulary are semantically unimportant. This is restrictive. Even in basic areas of mathematics, like algebra and geometry, one is familiar with the idea that theories may be presented in different ways with different primitive concepts. Similarly, if one consideres taxonomies, classification schemes, ontologies and in general any knowledge-based system, there are often many different ways to represent apparently the same information. This motivates the search for a concept of equivalence or synonymy that applies to logic programs or nonmonotonic theories that are formulated in different vocabularies. This is the topic of the present paper, which proposes and studies a formal concept of synonymy applying to logic programs and theories under answer sets semantics, more generally under a system of nonmonotonic reasoning called *equilibrium logic*. The basic approach is quite similar to one explored some years ago in the context of monotonic theories in classical (elementary) logic and its extensions [4, 22, 23]. The main novelty here is to argue that it applies equally well in the new area of ASP and nonmonotonic reasoning.

The rest of the paper is laid out as follows. We start by considering formal and informal desiderata that a concept of synonymy should fulfil. Next we introduce equilibrium logic as a logical foundation for ASP and extensions, and present the main characterisation of strong equivalence from [18]. In §4 we propose a definition of synonymy for propositional theories in equilibrium logic, give different characterisations of it, and show that it fulfils the adequacy conditions discussed in 2. In §5 we look briefly at how the previous ideas can be extended to open programs with variables. In §6 and §7 we conclude by looking at related work and considering topics for future study.

2 Synonymous Theories

What does it mean to say that two programs or theories, Π_1 and Π_2, in different languages, $\mathcal{L}_1$ and $\mathcal{L}_2$, are synonymous? We consider six desiderata D1-D6 that we believe should be satisfied by any basic concept of synonymy. D1-D3 and D5-D6 are quite general and seem to be applicable to any theories describing or modelling some knowledge domain; D4 takes account of the special nature of a nonmonotonic or logic programming system.

D1. Translatability. The language $\mathcal{L}_1$ of Π_1 should be translatable, via a mapping, say τ, into the language $\mathcal{L}_2$ of Π_2. The translation τ should be uniform, so we require it to be recursive.

[1] Universidad Rey Juan Carlos, Madrid, Spain. Email: dpearce@escet.urjc.es.
Partially supported by CICyT project TIC-2003-9001-C02, URJC project PPR-2003-39 and WASP (IST-2001-37004)
[2] Universidad de Málaga, Málaga, Spain. Email: a_valverde@ctima.uma.es.
Partially supported by CICyT project TIC-2003-9001-C01 and Junta de Andalucía project TIC-115.

[3] http://sv.mat.unical.it/infomix/

[4] For these and other examples as well as a thorough introduction to ASP, see [1]

D2. Semantic correspondence. There should be a corresponding correlation between the models of $\mathcal{L}_1$ and $\mathcal{L}_2$, in particular a mapping F from $\mathcal{L}_2$ models to $\mathcal{L}_1$ models that respects the translation τ in the sense that for any $\mathcal{L}_2$-model $\mathcal{M}$ and $\mathcal{L}_1$-formula φ,

$$F(\mathcal{M}) \models \varphi \Leftrightarrow \mathcal{M} \models \tau(\varphi).$$

D3. Equivalence. Under translation, Π_1 and Π_2 should be in an obvious sense equivalent.

D4. Intended models. The semantic correlation should respect the intended models of the two theories. In the present case this means preserving the property of being an equilibrium model or answer set: $\mathcal{M}$ is an answer set Π_2 iff $F(\mathcal{M})$ is an answer set of Π_1.

D5. Idempotence. If Π and Π_2 are synonymous under the previous mappings, then under corresponding mappings, say τ' and F', Π_2 should be synonymous with Π_1.

D6. Robustness. Π_1 and Π_2 should remain synonymous under the addition of new formulas, ie. for any Σ, $\Pi_1 \cup \Sigma$ should be synonymous with $\Pi_2 \cup \tau(\Sigma)$, similarly $\Pi_2 \cup \Gamma$ with $\Pi_1 \cup \tau'(\Gamma)$.

The first two conditions provide the cornerstone of any formal approach to intertheory relations. Different kinds of relations between theories are obtained by specifying additional conditions that the mappings should satisfy (see eg [23, 26, 29]). In this case we require (D3,D5) that theories are in an obvious sense equivalent once the translation maps are made available. Since we are dealing here with logic programs and their generalisations in the ASP framework, we can understand this either in the weaker sense of having the same answer sets, or in the sense of strong equivalence explained earlier. The problem is that if we choose the weaker variant then we have virtually no hope to fulfil condition D6 which requires that the theories remain equivalent when embedded in any richer context. On the other hand, if we interpret D3 to mean that under suitable translation manuals, Π_1 and Π_2 are strongly equivalent, then we may expect that Π_1 and Π_2 remain synonymous when extended with new rules.

Perhaps somewhat surprisingly we shall approach the problem of synonymy via the classical theory of interpretations. Briefly we shall say that theories are synonymous if each is faithfully interpreted in the other in such a way that the interpretations are idempotent (see below); this is basically the standard approach followed in ordinary predicate logic, see eg. [4, 28], adapted here to the propositional case. In the case of a nonmonotonic system like ASP there are several special issues to consider. First, to apply the classical theory of interpretations we need to represent programs and theories in a suitable logical form, which means identifying an underlying base logic. Second, we need to cope with nonmonotonicity, including the fact that only certain models (answer sets) are selected as intended. Third, it will be advantageous if the chosen base logic has similar metalogical properties to classical logic, especially the properties of *interpolation* and *Beth* that are of special significance in interpretability and definability theory.

3 Equilibrium Logic

As a logical foundation for answer set programming we use the non-classical logic of here-and-there $\mathbf{HT}$ and its nonmonotonic extension, *equilibrium logic* [24], which generalises answer set semantics for logic programs to arbitrary propositional theories, see eg [18]; we give only a very brief overview here, for more details the reader is referred to [24, 18, 25] and the logic texts cited below.[5]

[5] The standard version of equilibrium logic has two kinds of negation, intuitionistic and strong negation. For simplicity we deal here with the restricted version containing just the first negation and based on the logic of here-and-there. So we do not consider here eg logic programs with strong or explicit negation.

We denote non-logical vocabularies or *propositional languages* by $\mathcal{L}$, $\mathcal{L}'$, etc; they are simply sets of propositional variables. The $\mathcal{L}$–formulas of $\mathbf{HT}$ are built-up in the usual way using the logical constants: $\wedge$, $\vee$, $\rightarrow$, $\neg$, standing respectively for conjunction, disjunction, implication, and negation, the elements of $\mathcal{L}$ being the *atoms* of the language. A set of formulas is called a *theory*.

The axioms and rules of inference for $\mathbf{HT}$ are those of intuitionistic logic (see eg [6]) together with: the axiom schema

$$(\neg\alpha \rightarrow \beta) \rightarrow (((\beta \rightarrow \alpha) \rightarrow \beta) \rightarrow \beta)$$

which characterises the 3-valued here-and-there logic of Heyting [12], and Gödel [11] (hence it is sometimes known as Gödel's 3-valued logic).

The model theory of $\mathbf{HT}$ is based on the usual Kripke semantics for intuitionistic logic (see eg. [6]), but $\mathbf{HT}$ is complete for Kripke 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. It is convenient to represent an $\mathbf{HT}$ 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 $H \subseteq T$. If $\mathcal{M}$ is a $\mathbf{HT}$ model and φ is a formula, we denote by $\mathcal{M}(\varphi)$ the set of worlds where φ is true (either $\varnothing$ or $\{t\}$ or $\{h, t\}$); ie $w \in \mathcal{M}(\varphi)$ iff $\mathcal{M}, w \models \varphi$.

A formula φ is true in a $\mathbf{HT}$ model $\mathcal{M} = \langle H, T \rangle$ in symbols $\mathcal{M} \models \varphi$, if it is true at each world in $\mathcal{M}$. A formula φ is said to be *valid* in $\mathbf{HT}$, in symbols $\models \varphi$, if it is true in all $\mathbf{HT}$ models. Logical consequence for $\mathbf{HT}$ is understood as follows: φ is said to be an $\mathbf{HT}$ consequence of a theory Π, written $\Pi \models \varphi$, iff for all models $\mathcal{M}$ and any world $w \in \mathcal{M}$, $\mathcal{M}, w \models \Pi$ implies $\mathcal{M}, w \models \varphi$. Equivalently this can be expressed by saying that φ is true in all models of Π.

Let $\mathcal{L}$ be a proper sublanguage of $\mathcal{L}'$, ie $\mathcal{L} \subset \mathcal{L}'$; for any $\mathbf{HT}$ $\mathcal{L}'$-model $\mathcal{M} = \langle H, T \rangle$ we denote by $\mathcal{M}\restriction\mathcal{L}$ the $\mathbf{HT}$ $\mathcal{L}$-model formed by omitting the interpretation of all atoms in $\mathcal{L}' \setminus \mathcal{L}$ and we call this the *reduct* of $\mathcal{M}$ to $\mathcal{L}$. We can regard $\restriction\mathcal{L}$ as a (reduct) function mapping $\mathcal{L}'$-models onto $\mathcal{L}$-models.

Equilibrium models are special kinds of minimal $\mathbf{HT}$ models. We first define a partial ordering $\trianglelefteq$ on $\mathbf{HT}$ 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 $H \subseteq H'$.*

Equivalently, $\mathcal{M} \trianglelefteq \mathcal{M}'$ if for every atom p the following conditions hold: (i) if $\mathcal{M}, h \models p$ then $\mathcal{M}', h \models p$, and (ii) $\mathcal{M}, t \models p$ iff $\mathcal{M}', t \models p$.

Definition 2 *Let Π be a set of formulas and $\langle H, T \rangle$ a model of Π.*
(i) $\langle H, T \rangle$ is said to be total *if $H = T$.*
(ii) $\langle H, T \rangle$ is said to be an equilibrium *model if it is minimal under $\trianglelefteq$ among models of Π, and it is total.*

In other words a model $\langle H, T \rangle$ of Π is in equilibrium if it is total and there is no model $\langle H', T \rangle$ of Π with $H' \subset H$. Equilibrium logic is the logic determined by the equilibrium models of a theory.

It generalises answer set semantics in the following sense. For all the usual classes of logic programs, including normal, disjunctive and nested programs, equilibrium models correspond to answer sets. The 'translation' from the syntax of programs to $\mathbf{HT}$ propositional formulas is the trivial one, eg. a ground rule of a disjunctive program of the form

$$K_1 \vee \ldots \vee K_k \leftarrow L_1, \ldots L_m, notL_{m+1}, \ldots, notL_n$$

where the L_i and K_j are atoms corresponds to the $\mathbf{HT}$ sentence

$$L_1 \wedge \ldots \wedge L_m \wedge \neg L_{m+1} \wedge \ldots \wedge \neg L_n \rightarrow K_1 \vee \ldots \vee K_k$$

Proposition 3 ([24, 18]) *For any logic program Π, an $\mathbf{HT}$ model $\langle T, T \rangle$ is an equilibrium model of Π if and only if T is an answer set of Π.*

Two theories, Π and Π' are said to be *logically equivalent*, in symbols $\Pi \equiv \Pi'$, if they have the same $\mathbf{HT}$ models, and simply *equivalent* if they have the same equilibrium models. *Strong* equivalence is defined as in section 1. The main result we need is the following.

Proposition 4 ([18]) *Any two theories, Π and Π' are strongly equivalent iff they are logically equivalent, ie. $\Pi \equiv_s \Pi'$ iff $\Pi \equiv \Pi'$.*

4 Interpretability and Synonymy

Let $\mathcal{L}$ be a propositional language and $p \notin \mathcal{L}$ a new propositional variable. Let Π be a theory in $\mathcal{L} \cup \{p\}$. Explicit and implicit definability are understood as follows ($\alpha \leftrightarrow \beta$ simplifies $(\alpha \rightarrow \beta) \wedge (\beta \rightarrow \alpha)$).

Definition 5 *(i) p is said to be* explicitly definable *in Π, if there is an $\mathcal{L}$-formula φ such that*

$$\Pi \models \varphi \leftrightarrow p$$

(ii) p is said to be implicitly definable *in Π if for any models $\mathcal{M}$ and $\mathcal{M}'$ of Π such that $\mathcal{M} \restriction \mathcal{L} = \mathcal{M}' \restriction \mathcal{L}$ we have $\mathcal{M}(p) = \mathcal{M}'(p)$.*

In other words, an atom p is implicitly definable if whenever the interpretation of the $\mathcal{L}$ atoms in models of Π is fixed, the interpretation of p becomes fixed also. The above definitions are readily extended to the case where several new variables are definable in a theory.

It is well-known that the logic $\mathbf{HT}$, like any extension of intuitionistic logic, has the *Beth property*, ie. if $\mathcal{L}$, p and Π are as in Definition 5 above, then p is explicitly definable in Π if and only if it is implicitly definable in Π.[6]

Let $\mathcal{L}_1$ and $\mathcal{L}_2$ be disjoint languages.[7] By an *interpretation* of $\mathcal{L}_1$ in $\mathcal{L}_2$ we mean a mapping τ from $\mathcal{L}_1$-formulas to $\mathcal{L}_2$-formulas defined recursively as follows: $\tau(\varphi \wedge \psi) = \tau(\varphi) \wedge \tau(\psi)$, $\tau(\varphi \vee \psi) = \tau(\varphi) \vee \tau(\psi)$, $\tau(\varphi \rightarrow \psi) = \tau(\varphi) \rightarrow \tau(\psi)$, $\tau(\neg\varphi) = \neg\tau(\varphi)$.

Any interpretation τ of $\mathcal{L}_1$ in $\mathcal{L}_2$ induces a mapping F_τ from $\mathcal{L}_2$-models to $\mathcal{L}_1$ models: $F_\tau(\mathcal{M})(p) = \mathcal{M}(\tau(p))$; in terms of $\mathbf{HT}$ models, $F_\tau(\mathcal{M})$ is the model such that for any $\mathcal{L}_1$–formula φ and any $w \in \{h, t\}$

$$F_\tau(\mathcal{M}), w \models \varphi \quad \Leftrightarrow \quad \mathcal{M}, w \models \tau(\varphi) \tag{1}$$

[6] In [20] it is shown that $\mathbf{HT}$ has the stronger property of projective Beth, which allows that definability may be with respect to a specific sublanguage. This holds in far fewer logics. Note that in the literature on nonclassical logics implicit definability is often formulated as in [20] in a proof-theoretic style. The equivalence to the above semantic version can be shown via the compactness and completeness theorems for $\mathbf{HT}$.

[7] Any languages can be made disjoint by renaming. Alternatively we can allow that $\mathcal{L}_1$ and $\mathcal{L}_2$ have a common sublanguage which any translations simply leave untouched, ie the sublanguage is always translated by the identity map.

Therefore the following property holds

$$F_\tau(\mathcal{M}) \models \varphi \quad \Leftrightarrow \quad \mathcal{M} \models \tau(\varphi) \tag{2}$$

Let Π_1 and Π_2 be theories in $\mathcal{L}_1$ and $\mathcal{L}_2$ respectively and let τ be an interpretation of $\mathcal{L}_1$ in $\mathcal{L}_2$. Then τ is said to be an *interpretation of* Π_1 *in* Π_2 if for all $\mathcal{L}_1$–formula φ,

$$\Pi_1 \models \varphi \quad \Rightarrow \quad \Pi_2 \models \tau(\varphi). \tag{3}$$

In this case it is evident that

$$\mathcal{M} \models \Pi_2 \quad \Rightarrow \quad F_\tau(\mathcal{M}) \models \Pi_1. \tag{4}$$

An interpretation of Π_1 in Π_2 is said to be *faithful* if the converse of (3) also holds, ie we have $\Pi_1 \models \varphi$ iff $\Pi_2 \models \tau(\varphi)$. As in classical interpretability theory, further special cases of interpretation can be obtained by imposing additional conditions on the syntactic and semantic translations.

Proposition 6 *Let τ be an interpretation of Π_1 in Π_2. Then the following are equivalent.*
(i) For every $\mathcal{L}_2$-formula ψ there is an $\mathcal{L}_1$-formula φ such that $\Pi_2 \models \psi \leftrightarrow \tau(\varphi)$; ie τ is surjective.
(ii) There is an interpretation σ of $\mathcal{L}_2$ in $\mathcal{L}_1$ such that for every $\mathcal{L}_2$-formula ψ, $\Pi_2 \models \psi \leftrightarrow \tau\sigma(\psi)$.
(iii) The mapping F_τ from models of Π_2 into models of Π_1 is an injection.

Proof sketch. (ii) implies (i) and (i) implies (iii) are straightforward. To show that (iii) implies (ii), one applies the Beth property. In particular one observes that in $\mathcal{L}_2 \cup \mathcal{L}_1$ the theory $\Pi_2 \cup \{p \leftrightarrow \tau(p); p \in \mathcal{L}_1\}$ implicitly defines the atoms of $\mathcal{L}_2$. Therefore there is an interpretation, say σ of $\mathcal{L}_2$ in $\mathcal{L}_1$ which moreover satisfies $\Pi_2 \models \varphi \leftrightarrow \tau\sigma(\varphi)$.

Suppose that any of (i)-(iii) hold and additionally τ is a faithful interpretation. Then τ is said to be a *bijective interpretation* of Π_1 in Π_2. It is easy to verify that if τ is a bijective interpretation of Π_1 in Π_2, then the interpretation σ of Π_2 in Π_1, defined by condition (ii), is also bijective and

$$\Pi_1 \models \varphi \leftrightarrow \sigma\tau(\varphi) \quad \text{and} \quad \Pi_2 \models \psi \leftrightarrow \tau\sigma\ (\psi) \tag{5}$$

for all formulas φ, ψ. The interpretation σ is called the *inverse* of τ and we say that the two programs or theories are *synonymous* with respect to τ and σ.

Proposition 7 *If τ is a bijective interpretation of Π_1 in Π_2 then the mapping F_τ is a one-one correspondence between models of Π_1 and models of Π_2.*

Given an inverse interpretation σ, we can map models $\mathcal{M}$ of $\mathcal{L}_1$ to models $F_\sigma(\mathcal{M})$ of $\mathcal{L}_2$ in the same way as before. It is readily seen that $F_\sigma(F_\tau(\mathcal{M})) = \mathcal{M}$ if $\mathcal{M}$ is a model of Π_2; however the equality need not hold for other models (even in the classical case).

4.1 Verifying the adequacy conditions

Let us now consider synonymy in light of the adequacy conditions D1-D6. First we consider the sense in which two synonymous theories can be considered equivalent. Given interpretations τ and σ as above, let $\overline{\tau}$ be the set of definitions $\{p \leftrightarrow \tau(p); p \in \mathcal{L}_1\}$; similarly let $\overline{\sigma}$ be the set $\{q \leftrightarrow \tau(q); q \in \mathcal{L}_2\}$.

Proposition 8 *Let Π_1 and Π_2 be synonymous wrt τ and σ. Then $\Pi_2 \cup \overline{\tau}$ is strongly equivalent with $\Pi_1 \cup \overline{\sigma}$. Thus Π_1 and Π_2 have a common definitional extension, ie there is a theory Π in $\mathcal{L}_2 \cup \mathcal{L}_1$, such that $\Pi_2 \cup \overline{\tau} \equiv \Pi_1 \cup \overline{\sigma} \equiv \Pi$.*

Proof sketch. The proof that synonymous theories have a common definitional extension is a straightforward adaptation of the standard proofs for classical logic, see eg. [4, 28]. The strong equivalence claim follows from Proposition 4.

In fact Proposition 8 can be strengthened to an equivalence: two theories are bijectively interpretable if and only if they have a common definitional extension. This expresses one way in which the two theories are in an obvious sense equivalent once enriched with suitable translation manuals. Notice too that there is a close relationship between Π_2 and the translation $\tau(\Pi_1)$ of Π_1 (similarly between Π_1 and the translation $\sigma(\Pi_2)$ of Π_2). It is already clear that $\Pi_2 \models \tau(\Pi_1)$. Although it is not generally true, even in the classical case, that $\Pi_2 \equiv \tau(\Pi_1)$), we do however have:

Corollary 9 *Let Π_1 and Π_2 be synonymous wrt τ and σ. For any $\mathcal{L}_2$-formula φ, $\Pi_2 \models \varphi \leftrightarrow \tau\sigma(\varphi)$, and $\Pi_2 \models \varphi \Rightarrow \tau(\Pi_1) \models \tau\sigma(\varphi)$.*

Next we turn to condition D4.

Proposition 10 *Let Π_1 and Π_2 be theories in $\mathcal{L}_1$ and $\mathcal{L}_2$ respectively, synonymous wrt τ and σ. Then the bijective mapping F_τ from models of Π_2 to models of Π_1 preserves the equilibrium property, ie. $\mathcal{M} \models_e \Pi_2$ iff $F_\tau(\mathcal{M}) \models_e \Pi_1$.*

Proof. Let $\mathcal{M}$ be an equilibrium model of Π_2 and assume that $F_\tau(\mathcal{M})$ is not in equilibrium. So there is $\mathcal{M}' \trianglelefteq F_\tau(\mathcal{M})$ with $\mathcal{M}' \models \Pi_1$ and φ such that $\mathcal{M}', t \models \varphi$ and $\mathcal{M}', h \not\models \varphi$. For every q in $\mathcal{L}_2$, using (1), and part (ii) of proposition 6 yields that $F_\sigma(\mathcal{M}'), t \models q \Leftrightarrow \mathcal{M}, t \models q$ and thus $F_\sigma(\mathcal{M}') \trianglelefteq \mathcal{M}$, which contradicts the equilibrium of $\mathcal{M}$, because $F_\sigma(\mathcal{M}')$ is not total: $\mathcal{M}'(p) = \mathcal{M}'(\sigma(\tau(p))) = F_\sigma(\mathcal{M}')(\tau(p))$. Reciprocally, if $\mathcal{M}$ is a model of Π_2 such that $F_\tau(\mathcal{M})$ is an equilibrium model of Π_1 and $\mathcal{M}' \triangleleft \mathcal{M}$, then $F_\tau(\mathcal{M}') \triangleleft F_\tau(\mathcal{M})$, which contradicts the equilibrium of $F_\tau(\mathcal{M})$.

Clearly, condition D5 is satisfied and (5) describes the sense in which the correspondence between Π_1 and Π_2 is idempotent. Lastly we consider D6.

Proposition 11 *Let Π_1 and Π_2 be theories in $\mathcal{L}_1$ and $\mathcal{L}_2$ respectively synonymous wrt τ and σ. Let Γ a set of $\mathcal{L}_1$-formulas. Then $\Pi_1 \cup \Gamma$ is synonymous with $\Pi_2 \cup \tau(\Gamma)$ wrt τ and σ.*

Proof. If $\Pi_1 \cup \Gamma \models \varphi$ and $\mathcal{M} \models \Pi_2 \cup \tau(\Gamma)$, then $F_\tau(\mathcal{M}) \models \Pi_1$ and $F_\tau(\mathcal{M}) \models \Gamma$; thus $F_\tau(\mathcal{M}) \models \varphi$ and $\mathcal{M} \models \tau(\varphi)$. Therefore τ is an interpretation of Π_1 in Π_2. If $\Pi_2 \cup \tau(\Gamma) \models \tau(\varphi)$ and $\mathcal{M} \models \Pi_1 \cup \Gamma$, then $\mathcal{M} = F_\sigma(F_\tau(\mathcal{M}))$, $F_\sigma(\mathcal{M}) \models \tau(\Gamma)$ and $F_\sigma(\mathcal{M}) \models \Pi_2$; thus $F_\sigma(\mathcal{M}) \models \tau(\varphi)$, $\mathcal{M} \models \sigma(\tau(\varphi))$ and $\mathcal{M} \models \varphi$. Therefore τ is faithful. Finally we prove that F_τ is injective: if $\mathcal{M}$ and $\mathcal{M}'$ are models of $\Pi_2 \cup \tau(\Gamma)$ such that $F_\tau(\mathcal{M}) = F_\tau(\mathcal{M}')$ then for all formula φ we have $\mathcal{M}'(\varphi) = F_\tau(\mathcal{M}')(\sigma(\varphi)) = F_\tau(\mathcal{M})(\sigma(\varphi)) = \mathcal{M}(\varphi)$.

5 More Realism

In this paper we have dealt with propositional theories in equilibrium logic. In virtue of proposition 3, we have therefore covered the case of ground, propositional logic programs under answer set semantics. In answer set programming, however, one usually deals with open predicates containing free variables which become instantiated during the process of grounding. In this case, to deal with the translation of an open predicate, $P(x)$, we need to add further conditions on the mapping τ. To give a simple example, suppose in one language, say for speaking about graphs, we have a primitive predicate $Connected(x, y)$ expressing that two nodes are connected, while in another language we have a predicate $Path(x, y)$ to express that there is a directed path from x to y. Then we translate from the former vocabulary into the latter by means of the definition

$$Connected(x, y) \leftrightarrow Path(x, y) \vee Path(y, x). \tag{6}$$

When our programs are grounded (6) becomes instantiated, so for example if τ is the translation respecting (6) and a, b are nodes, we have $\tau(C(a, b)) = P(a, b) \vee P(b, a)$, where C and P abbreviate $Connected$ and $Path$ respectively. Similarly, for any c, d the translation of the atom $C(c, d)$ will take the same form. This uniformity of translation across all instantiations of a predicate can be added to our requirements for synonymy. We sketch this idea as follows. Suppose that $\mathcal{L}_1$ and $\mathcal{L}_2$ contain disjoint sets of predicate symbols, no function symbols and a shared set of individual constants or names. Let Π_1 and Π_2 be logic programs (of any kind) in $\mathcal{L}_1$ and $\mathcal{L}_2$ respectively. Let $\overline{\tau}$ be a set of definitions of the following form

$$P(x_1, \ldots, x_n) \leftrightarrow \varphi(x_1, \ldots, x_n), \tag{7}$$

one for each n-place predicate $P(x_1, \ldots, x_n)$ of $\mathcal{L}_1$, where φ is an $\mathcal{L}_2$-formula whose free variables are among $x_1, \ldots, x_n$. Likewise suppose that $\overline{\sigma}$ is a set of definitions

$$Q(x_1, \ldots, x_m) \leftrightarrow \psi(x_1, \ldots, x_m), \tag{8}$$

one for each $\mathcal{L}_2$-predicate Q, in terms of $\mathcal{L}_1$-formulas. Let the ground versions of Π_1 and Π_2 be $g(\Pi_1)$ and $g(\Pi_2)$ respectively. The set of all ground instances of definitions in $\overline{\tau}$, $\overline{\sigma}$ of form (7) and (8) give rise in the obvious way to corresponding translations τ, from the language of $g(\Pi_1)$ into that of $g(\Pi_2)$ and σ from the language of $g(\Pi_2)$ into that of $g(\Pi_1)$. We can then say that Π_1 and Π_2 are *synonymous* with respect to $\overline{\tau}, \overline{\sigma}$ if τ is a bijective interpretation of $g(\Pi_1)$ into $g(\Pi_2)$ with inverse σ. Clearly, if Π_1 and Π_2 are *synonymous* with respect to $\overline{\tau}, \overline{\sigma}$ then the common definitional extension of $g(\Pi_1)$ and $g(\Pi_2)$ is equivalent to $g(\Pi_2 \cup \overline{\tau})$ and $g(\Pi_1 \cup \overline{\sigma})$.[8]

6 Literature and Related Work

In classical logic there is a large and well-developed body of work on interpretability dating from the 1950s. The first systematic treatments of synonymous theories in this context can be found in [3, 4], a more algebraic approach can be found in [16]. The classical version of Proposition 6 is essentially contained in [3], though a more detailed statement and proof can be found in [28]. Outside the field of mathematics, the classical theory of interpretability and definitional equivalence was extended and applied to empirical forms of knowledge in [22, 26, 23]; see also [29] for a more recent account of translatability issues in such contexts. The theory of interpretations and equivalence in nonclassical logics is less developed, however especially in the case of superintuitionistic logics much is known about key properties, such as interpolation and Beth, on which interpretability theory

[8] Note that if Π_1 and Π_2 are disjunctive programs and the formulas φ, ψ in all definitions of form (7) and (8) are implication-free, then $\Pi_2 \cup \overline{\tau}, \Pi_1 \cup \overline{\sigma}$ have the form of logic programs with nested expressions.

depends, see eg. [19, 20]. In the context of nonmonotonic logic programming the study of different kinds of equivalence between programs is relatively new (see references in section 1) and until now has not, to our knowledge, considered the case of programs in different languages. In two recent works, [14, 10], there has been some discussion of the role and properties of definitions in ASP.

7 Concluding Remarks

We have tried to show how formal approaches to intertheory relations developed for mathematical and scientific knowledge might be applied to systems of logic programming and nonmonotonic reasoning used for practical problem solving and knowledge representation in AI. In particular, we have argued that the classical theory of interpretability and definitional equivalence can be applied in the context of propositional logic programs under answer set semantics and more generally in the system of equilibrium logic. In this setting we regard theories as synonymous if each is bijectively interpretable in the other, and we have characterised this relation in different ways. We also showed that this reconstruction satisfies a number of intuitive, informal adequacy conditions. The applicability of what is essentially a classical logical approach in a nonclassical context relies on two essential features: first, our underlying logic has several properties such as *Beth* that help to relate the syntax to the semantics of definitions and translations; secondly, in ASP and equilibrium logic the strong concept of equivalence between theories is fully captured in the underlying monotonic logic (*here-and-there*). This allows us to define a robust or modular concept of equivalence across different languages.

Many avenues are left open for future exploration. For example, for a more realistic treatment of the translation of open predicates one may want to relax the restriction that the languages concerned share the same set of individual constants or that the defining formulas (φ, ψ in (7), (8)) may contain no new free variables. Eventually a full first-order treatment would be desirable. Secondly, one might search for simple structural properties on the models of two programs or theories that are equivalent to or sufficient for synonymy. Thirdly, based on these or other properties of the theories concerned, it would be useful to develop systems for checking synonymy, thereby extending current methods for checking strong equivalence in the case of programs in the same language [8, 27].

REFERENCES

[1] C. Baral *Knowlewdge Representation, Reasoning and Declarative Problem Solving* Cambridge University Press, 2003.

[2] M. Balduccini, M. Gelfond, R. Watson and M. Noguiera. The USA-Advisor: A case study in answer set planning. In *Logic Programming and Nonmonotonic Reasoning*, LPNMR 2001, Springer LNAI 2173, 2001.

[3] K. de Bouvère. Logical Synonymity. *Indagationes Mathematicae* 27 (1965), 622-629.

[4] K. de Bouvère. Synonymous Theories. In J. Addison, L. Henkin and A.Tarski (eds), *Symposium of the Theory of Models*, North-Holland, Amsterdam, 1965, 402-406.

[5] F. Calimeri, S. Galizia, M. Ruffolo, P. Rullo Enhancing Disjunctive Logic Programming for Ontology Specification. Proceedings AGP 2003

[6] D. van Dalen. Intuitionistic logic. In *Handbook of Philosophical Logic, Volume III: Alternatives in Classical Logic*, Kluwer, Dordrecht, 1986.

[7] T. Eiter and M. Fink. Uniform equivalence of logic programs under the stable model semantics. In *Int. Conf. of Logic Programming, ICLP'03*, Mumbay, India. Springer, 2003.

[8] T. Eiter, M. Fink, H. Tompits and S. Woltran Simplifying Logic Programs under Uniform and Strong Equivalence In V. Lifschitz and I. Niemelä (eds), *Logic Programming and Nonmonotonic Reasoning*, LPNMR 2004, Springer LNAI 2923, 2004.

[9] H. Enderton. *A Mathematical Introduction to Logic* Academic Press, 1972.

[10] S. Erdogem and V. Lifschitz. Definitions in Answer Set Programming in V. Lifschitz and I. Niemelä (eds), *Proceedings LPNMR 2004*, Springer LNAI 2923, 2004.

[11] K. Gödel. Zum intuitionistischen aussagenkalkül. *Anzeiger der Akademie der Wissenschaften Wien, mathematisch, naturwissenschaftliche Klasse*, 69:65–66, 1932.

[12] A. Heyting. Die formalen regeln der intuitionistischen logik. *Sitzungsberichte der Preussischen Akademie der Wissenschaften, Physikalisch-mathematische Klasse*, pages 42–56, 1930.

[13] D. De Jongh and L. Hendriks. Characterization of strongly equivalent logic programs in intermediate logics. *Theory and Practice of Logic Programming*, 3(3):259–270, 2003.

[14] P. Ferraris and V. Lifschitz. Weight Constraints as Nested Expressions *Theory and Practice of Logic Programming*, to appear, 2004.

[15] T. Janhumen, I. Niemelä, D. Seipel, P. Simons, and J.-H. You. Unfolding partiality and disjunctions in stable model semantics. CoRR: cs.AI/0303009, March, 2003.

[16] S. Kanger. Equivalent Theories. *Theoria*, 38 (1972), 1-6.

[17] N. Leone, G. Pfeifer, W. Faber, T. Eiter, G, Gottlob, S. Perri and F. Scarcello. The DLV System for Knowledge Representation and Reasoning. CoRR: cs.AI/0211004, September, 2003.

[18] V. Lifschitz, D. Pearce, and A. Valverde. Strongly equivalent logic programs. *ACM Transactions on Computational Logic*, 2(4):526–541, 2001.

[19] L. Maksimova. Interpolation Properties in Superintuitionistic Logics. *Studia Logic* 38(4): 419-428, 1979.

[20] L. Maksimova. Intuitionistic logic and implicit definability. *Annals of Pure and Applied Logic* 105(1-3): 83-102, 2000

[21] E. Oikarinen and T. Janhunen. Verifying the Equivalence Logic Programs in the Disjunctive Case. In V. Lifschitz and I. Niemela (eds), *Logic Programming and Nonmonotonic Reasoning*, LPNMR 2004, Springer LNAI 2923, 2004.

[22] D. Pearce. Some Relations between Empirical Systems. *Epistemologia* 4 (1981), 363-380.

[23] D. Pearce. *Roads to Commensurability* Kluwer (Synthese Library Vol. 187),1987.

[24] D. Pearce. A new logical characterization of stable models and answer sets. In *Non-Monotonic Extensions of Logic Programming, NMELP 96*, LNCS 1216, pages 57–70. Springer, 1997.

[25] D. Pearce, I. P. de Guzmán, and A. Valverde. A tableau calculus for equilibrium entailment. In *Automated Reasoning with Analytic Tableaux and Related Methods, TABLEAUX 2000*, LNAI 1847, pages 352–367. Springer, 2000.

[26] D. Pearce and V. Rantala. New Foundations for Metascience. *Synthese*, 1985.

[27] D. Pearce and A. Valverde. Uniform Equivalence for Equilibrium Logic and Logic Programs. In V. Lifschitz and I. Niemelaä (eds) *Logic Programming and Nonmonotonic Reasoning*, LPNMR 2004, Springer LNAI 2923, 2004.

[28] C. Pinter Properties Preserved under Definitional Equivalence and Interpretations. In *Zeitschr. f. math. Logik und Grundlagen d. Math.*, 24: 481–488, 1978.

[29] V. Rantala. *Explanatory Translation*. Kluwer (Synthese Library Vol 312), 2002.

[30] P. Simons, I. Niemelä, and T. Soininen. Extending and implementing the stable model semantics. *Artificial Intellingence*, 138(1–2):181–234, 2002.

[31] H. Turner. Strong equivalence for logic programs and default theories (made easy). In *Proc. of the Logic Programming and Nonmonotonic Reasoning, LPNMR'01*, Springer LNAI 2173, pages 81–92, 2001.

[32] H. Turner Strong Equivalence for Causal Theories In V. Lifschitz and I. Niemelä (eds) *Logic Programming and Nonmonotonic Reasoning*, LPNMR 2004, Springer LNAI 2923, 2004.