Interpretability and Equivalence in Quantified Equilibrium Logic

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

[1] Computing Science and Artificial Intelligence,
Univ. Rey Juan Carlos, (Móstoles, Madrid), Spain.
davidandrew.pearce@urjc.es
[2] Dept. of Applied Mathematics, Univ. of Málaga, Spain.
a_valverde@ctima.uma.es

Abstract. The study of synonymy among propositional theories in equilibrium logic, begun in [36], is extended to the first-order case.

1 Introduction

Quantified equilibrium logic (QEL) has been developed in [37–39] as a logical foundation for answer set programs with variables. In particular, the version of QEL presented in [39] and [23] can be considered adequate for the general, first-order version of stable model semantics as given in [16]. This version of QEL is based on the logic $\mathbf{QHT}^s_=$, called quantified here-and-there logic with static domains and decidable equality. Logic programs or general theories are strongly equivalent with respect to QEL (or stable model semantics) if and only if they are logically equivalent in $\mathbf{QHT}^s_=$, [23].

In answer set programming (ASP) strong equivalence (and other forms of equivalence between programs) has been recognised as providing an important conceptual and practical tool for program simplification, transformation and optimisation. Following its initial study in [22], the concept of strong equivalence for logic programs in ASP has given rise to a substantial body of further work looking at different characterisations [15, 43], new variations and applications of the idea [8, 35, 44], as well as developing systems to test for strong equivalence [35, 9]. Recently, some of this work on program transformation [10, 45] has been extended to the first-order case.

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 idea was pursued in [36] which proposed a formal concept of synonymy applying to logic programs and propositional theories in equilibrium logic and answer set semantics. The aim of the present paper is to extend this

⋆ Partially supported by CICyT projects TIC-2003-9001-C02 and TIN2006-15455-CO3.
⋆⋆ Partially supported by CICyT project TIC-2003-9001-C01, TIN2006-15455-CO1 and Junta de Andalucia project TIC-115

work to theories formulated in first-order logic by using quantified equilibrium logic. We start following [36] by considering formal and informal desiderata that a concept of synonymy should fulfil. We then introduce QEL as a logical foundation for ASP and extensions, and present the main characterisation of strong equivalence from [23]. In §4 we propose a strong concept of equivalence or synonymy for theories in quantified equilibrium logic, give different characterisations of it, and show that it fulfils the adequacy conditions discussed in 2. The main characteristics of this concept are as follows. Theories Π_1 and Π_2 in distinct languages are said to be synonymous if each is bijectively interpretable in the other. In particular, this means that there is faithful interpretation of each theory in the other and a one-one correspondence between the models of the two theories. This correspondence preserves the property of being an equilibrium model or answer set. In addition, Π_1 has a definitional extensions that is strongly equivalent to a definitional extension of Π_2. Moreover, in a suitable sense, Π_1 and Π_2 remain equivalent or synonymous when extended by the addition of new formulas.

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.

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

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_1 is synonymous with Π_2 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 $\Pi_2 \cup \Pi$ with $\Pi_1 \cup \tau'(\Pi)$.

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 [30, 34, 41]). 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 classical predicate logic, see eg. [4, 40]. We adapt it here to the case of a nonmonotonic system based on a non-classical logic.

3 Quantified Equilibrium Logic (QEL)

Equilibrium logic for propositional theories and logic programs was presented in [31] as a foundation for answer set semantics and extended to the first-order case in [37, 38] and in slightly more general, modified form in [39]. For a survey of the main properties of equilibrium logic, see [32]. Usually in quantified equilibrium logic we consider a full first-order language allowing function symbols and we include a second, strong negation operator as occurs in several ASP dialects. For the present purpose we consider the function-free language with a single negation symbol, '$\neg$'. So, in particular, we shall work with a quantified version of the logic HT of *here-and-there*. In other respects we follow the treatment of [39].

3.1 General Structures for Quantified Here-and-There Logic

A *function-free first-order language* $\mathcal{L} = \langle C, P\rangle$ consists of a sets of constants C and predicate symbols P; each predicate symbol $p \in P$ has an assigned arity. Moreover, we assume a fixed countably infinite set of variables, the symbols, '$\rightarrow$', '$\vee$', '$\wedge$', '$\neg$', '$\exists$', '$\forall$' and auxiliary parentheses '$($','$)$'. Variables and constant are generically called *terms*. *Atoms* and *formulas* are constructed as usual; *closed* formulas, or *sentences*, are those where each variable is bound by some quantifier. A *theory* Π is a set of sentences.

If D is a non-empty set, we denote by $\mathrm{At}_D(C, P)$ the set of atomic sentences of $\mathcal{L} = \langle C, P\rangle$ with additional constant symbols for each element of D. A *here-and-there* $\mathcal{L}$-structure with static domains is a tuple $\mathcal{I} = \langle (D, I), I^h, I^t\rangle$ where

- D is a non-empty set, called the *domain* of $\mathcal{I}$.
- $I\colon C \cup D \rightarrow D$ is called the *assignment* and verifies $I(d) = d$ for all $d \in D$.
- $I^h \subseteq I^t \subset \mathrm{At}_D(C, P)$.

We can think of $\mathcal{I}$ as a structure similar to a first-order classical structure, but having two parts or components h and t that correspond to two different points or "worlds", 'here' and 'there' in the sense of Kripke semantics for intuitionistic logic [7], where the worlds are ordered by $h \leq t$. At each world $w \in \{h, t\}$ one verifies a set of atoms I^w in the expanded language for the domain D. We call the model static, since, in contrast to say intuitionistic logic, the same domain serves each of the worlds.[1] Since $h \leq t$, whatever is verified at h remains true at t. The satisfaction relation for $\mathcal{I}$ is defined so as to reflect the two different components, so we write $\mathcal{I}, w \models \varphi$ to denote that φ is true in $\mathcal{I}$ with respect to the w component. Evidently we should require that an atomic sentence is true at w just in case it belongs to I^w. Formally, if $p(t_1, \ldots, t_n) \in \mathrm{At}_D$ then

$$\mathcal{I}, w \models p(t_1, \ldots, t_n) \quad \text{iff} \quad p(I(t_1), \ldots, I(t_n)) \in I^w.$$

Then $\models$ is extended recursively as follows[2]:

- $\mathcal{I}, w \models \varphi \wedge \psi$ iff $\mathcal{I}, w \models \varphi$ and $\mathcal{I}, w \models \psi$.
- $\mathcal{I}, w \models \varphi \vee \psi$ iff $\mathcal{I}, w \models \varphi$ or $\mathcal{I}, w \models \psi$.
- $\mathcal{I}, t \models \varphi \to \psi$ iff $\mathcal{I}, t \not\models \varphi$ or $\mathcal{I}, t \models \psi$.
- $\mathcal{I}, h \models \varphi \to \psi$ iff $\mathcal{I}, t \models \varphi \to \psi$ and $\mathcal{I}, h \not\models \varphi$ or $\mathcal{I}, h \models \psi$.
- $\mathcal{I}, w \models \neg\varphi$ iff $\mathcal{I}, t \not\models \varphi$.
- $\mathcal{I}, t \models \forall x \varphi(x)$ iff $\mathcal{I}, t \models \varphi(d)$ for all $d \in D$.
- $\mathcal{I}, h \models \forall x \varphi(x)$ iff $\mathcal{I}, t \models \forall x \varphi(x)$ and $\mathcal{I}, h \models \varphi(d)$ for all $d \in D$.
- $\mathcal{I}, w \models \exists x \varphi(x)$ iff $\mathcal{I}, w \models \varphi(d)$ for some $d \in D$.

Truth of a sentence in a structure is defined as follows: $\mathcal{I} \models \varphi$ iff $\mathcal{I}, w \models \varphi$ for each $w \in \{h, t\}$; in this case, $\mathcal{I}$ is said to be a *model* of φ. An structure $\mathcal{I}$ is a model of a theory Π if it is a model of every $\varphi \in \Pi$, denoted by $\mathcal{I} \models \Pi$. A sentence φ is *valid* if it is true in all structures, denoted by $\models \varphi$. A sentence φ is a *consequence* of a theory Π if every model of Π is a model of φ, in symbols $\Pi \models \varphi$. The resulting logic is called *Quantified Here-and-There Logic with static domains* denoted by $\mathbf{QHT}^s(\mathcal{L})$. In terms of satisfiability and validity this logic is equivalent to the logic introduced before in [38].

The logic $\mathbf{QHT}^s(\mathcal{L})$ can be axiomatised as follows. We start with the usual axioms and rules of intuitionistic propositional logic and add the axiom of Hosoi

$$\alpha \vee (\neg\beta \vee (\alpha \to \beta))$$

which determines 2-element, here-and-there models. This system is extended to first-order logic (see [38, 39]) by adding the following axiom to obtain the usual non-static version of first-order here-and-there logic:

$$\forall x \neg\neg\alpha(x) \to \exists x(\alpha(x) \to \forall x \alpha(x))$$

[1] Alternatively it is quite common to speak of a logic with *constant* domains. However this is ambiguous since it might suggest that the domain is composed only of constants, which is not intended here.

[2] The reader may easily check that the following correspond exactly to the usual Kripke semantics for intuitionistic logic given our assumptions about the two worlds h and t and the single domain D, see eg [6]

Finally, we add the following axiom for static domains, to obtain $\mathbf{QHT}^s(\mathcal{L})$:

$$\neg\neg\exists x\alpha(x) \rightarrow \exists x\neg\neg\alpha(x)$$

Ono proved in [28] that the system obtained by extending the propositional calculus with the axiom $\forall x(\alpha(x) \vee \beta) \rightarrow (\forall x\alpha(x) \vee \beta)$ is complete for $\mathbf{QHT}^s(\mathcal{L})$. In [23], another complete calculus is obtained by extending the propositional calculus with the axiom

$$\exists x(\alpha(x) \rightarrow \forall x\alpha(x))$$

In this paper we also consider the equality predicate, $\doteq \notin P$, interpreted by the following condition for every $w \in \{h, t\}$

– $\mathcal{M}, w \models a \doteq b$ iff $I(a) = I(b)$ for all constants a, b.

To obtain a complete axiomatisation, we then need to add the axiom of "decidible equality"

$$\forall x\forall y(x \doteq y \vee x \not\doteq y).$$

We denote the resulting logic by $\mathbf{QHT}^s_=(\mathcal{L})$ (see [23] for details).

As usual in first order logic, satisfiability and validity are independent from the language. If $\mathcal{I} = \langle (D, I), I^h, I^t \rangle$ is an $\mathcal{L}'$-structure and $\mathcal{L}' \supset \mathcal{L}$, we denote by $\mathcal{I}|_\mathcal{L}$ the restriction of $\mathcal{I}$ to the sublanguage $\mathcal{L}$:

$$\mathcal{I}|_\mathcal{L} = \langle (D, I|_\mathcal{L}), I^h|_\mathcal{L}, I^t|_\mathcal{L} \rangle$$

Proposition 1. *Suppose that $\mathcal{L}' \supset \mathcal{L}$, Π is a theory in $\mathcal{L}$ and $\mathcal{M}$ is an $\mathcal{L}'$-model of Π. Then $\mathcal{M}|_\mathcal{L}$ is a $\mathcal{L}'$-model of Π.*

Proposition 2. *Suppose that $\mathcal{L}' \supset \mathcal{L}$ and $\varphi \in \mathcal{L}$. Then φ is valid (resp. satisfiable) in $\mathbf{QHT}^s_=(\mathcal{L})$ if and only if is valid (resp. satisfiable) in $\mathbf{QHT}^s_=(\mathcal{L}')$.*

This proposition allows us to omit reference to the language in the logic so it can be denoted simply by $\mathbf{QHT}^s_=$.

3.2 Equilibrium Models

As in the propositional case, quantified equilibrium logic is based on a suitable notion of minimal model.

Definition 1. *Among quantified here-and-there structures we define the order $\trianglelefteq$ as follows: $\langle (D, I), I^h, I^t \rangle \trianglelefteq \langle (D', J), J^h, J^t \rangle$ if $D = D'$, $I = J$, $I^t = J^t$ and $I^h \subseteq J^h$. If the subset relation holds strictly, we write '$\triangleleft$'.*

Definition 2. *Let Π be a theory and $\mathcal{I} = \langle (D, I), I^h, I^t \rangle$ a model of Π.*

1. *$\mathcal{I}$ is said to be* total *if $I^h = I^t$.*
2. *$\mathcal{I}$ is said to be an* equilibrium *model of Π (or short, we say: "$\mathcal{I}$ is in equilibrium") if it is minimal under $\trianglelefteq$ among models of Π, and it is total. It is denoted by $\mathcal{I} \models_e \Pi$.*

Notice that a total here-and-there model of a theory Π is equivalent to a classical first order model of Π.

The logic defined by the equilibrium models is called *Quantified Equilibrium Logic* and it is also independent of the language, as seen by the following result.

Proposition 3. *Let Π be a theory in $\mathcal{L}$ and $\mathcal{M}$ an equilibrium model of Π in $\mathbf{QHT}^s_=(\mathcal{L}')$ with $\mathcal{L}' \supset \mathcal{L}$. Then $\mathcal{M}|_\mathcal{L}$ is an equilibrium model of Π in $\mathbf{QHT}^s_=(\mathcal{L})$.*

3.3 Strong equivalence for theories

We say that two sets Γ, Δ of first-order sentences are *strongly equivalent* if for every set Σ of first-order sentences, possibly of a larger signature, the sets $\Gamma \cup \Sigma$, $\Delta \cup \Sigma$ have the same equilibrium models.

Theorem 1 (Strong Equivalence of theories, [23]). *For any sets Γ, Δ of first-order sentences, the following conditions are equivalent:*

(i) the sets Γ and Δ are satisfied by the same here-and-there structures;
(ii) for every set Σ of first-order sentences, possibly of a larger signature, the sets $\Gamma \cup \Sigma$ and $\Delta \cup \Sigma$ have the same equilibrium models, ie Γ and Δ are strongly equivalent.

Note that the above notion of equilibrium model coincides with the concept of stable model for logic programs with variables presented in [16]. The concept of strong equivalence and its characterisation can be found in [23]. By strong completeness, condition (i) of Theorem 1 means that Γ and Δ are logically equivalent in $\mathbf{QHT}^s_=$.

4 Interpretability and Synonymy

We use the following notation and terminology. Boldface $\mathbf{x}$ stands for a tuple of variables, $\mathbf{x} = (x_1, \ldots, x_n)$, while $\varphi(\mathbf{x}) = \varphi(x_1, \ldots, x_n)$ is a formula whose free variables are $x_1,\ldots, x_n$, and $\forall \mathbf{x} = \forall x_1 \ldots \forall x_n$. If t_i are terms, then $\mathbf{t} = (t_1, \ldots, t_n)$ denotes a *vector* of terms. Let $\mathcal{L} = \langle C, P\rangle$ be a first-order language, $p \notin P$ a new predicate symbol and $\mathcal{L}' = \langle C, P \cup \{p\}\rangle$. Let Π be a theory in $\mathcal{L}'$. Explicit and implicit definability are understood as follows

(i) p is said to be *explicitly definable* in Π, if there is an $\mathcal{L}$-formula $\delta^\tau_p(\mathbf{x})$ such that

$$\Pi \models \forall \mathbf{x}(p(\mathbf{x}) \leftrightarrow \delta^\tau_p(\mathbf{x})).$$

δ^τ_p is called the *definition* of p.

(ii) p is said to be *implicitly definable* in Π if for any models $\mathcal{M}_1$ and $\mathcal{M}_2$ of Π such that $\mathcal{M}_1|_\mathcal{L} = \mathcal{M}_2|_\mathcal{L}$ we have $\mathcal{M}_1 = \mathcal{M}_2$.
By the strong completeness theorem for $\mathbf{QHT}^s_=$ proved in [23], this definition is equivalent to the following one.

(ii') p is *implicitly definable* in Π if

$$\Pi \cup \Pi[p/q] \models \forall \mathbf{x}(p(\mathbf{x}) \leftrightarrow q(\mathbf{x}))$$

where $q \notin P$ is a new predicate symbol with the same arity as p and $\Pi[p/q]$ is the theory obtained by replacing every occurrence of p by q.

In other words, p is implicitly definable if whenever the interpretation of the $\mathcal{L}$ predicates 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 predicates are definable in a theory.

4.1 Interpolation and Beth properties in superintuitionistic logics

When the conditions (i) and (ii') of explicit and implicit definability are always equivalent, the logic in question is said to have the *Beth property*, [18]. Closely related to Beth is the property of *interpolation*. A logic is said to have the interpolation property if whenever

$$\vdash \varphi \rightarrow \psi$$

there exists a sentence ξ (the *interpolant*) such that

$$\vdash \varphi \rightarrow \xi \quad \text{and} \quad \vdash \xi \rightarrow \psi$$

where all predicate and constant symbols of ξ are contained in both φ and ψ.

It can be shown that the interpolation property implies the Beth property in all superintuionistic predicate logics [18]. Moreover, Ono [28] showed that interpolation holds in the logic $\mathbf{QHT}^s$ of quantified here-and-there with constant domains.[3] Consequently, $\mathbf{QHT}^s$ also has the Beth property. Lastly, Maksimova showed in [24, 25] that adding pure equality axioms, eg decidible equality axiom, to any superintuitionistic logic preserves the interpolation and Beth properties (see also [18]). We conclude therefore

Proposition 4. *The logic $\mathbf{QHT}^s_=$ possesses the Beth property.*

Let $\mathcal{L}_1 = \langle C_1, P_1 \rangle$ and $\mathcal{L}_2 = \langle C_2, P_2 \rangle$ be disjoint languages.[4] By an *interpretation* of $\mathcal{L}_1$ in $\mathcal{L}_2$ we mean

1. For each predicate $p \in P_1$, an $\mathcal{L}_2$-formula δ_p^τ explicitly defining p by the formula $\forall \mathbf{x}(p(\mathbf{x}) \leftrightarrow \delta_p^\tau(\mathbf{x}))$; we denote by $\overline{\tau}$ the set of all definitions.
2. An induced mapping, also denoted by τ, from $\mathcal{L}_1$-formulas (resp. $\mathcal{L}_1$-terms) to $\mathcal{L}_2$-formulas (resp. $\mathcal{L}_2$-terms) such that
   (a) $\tau(x) = x$ and for every $a \in C_1$, $\tau(a) \in C_2$; if $\mathbf{t} = (t_1, \ldots, t_n)$ is a vector of terms, $\tau(\mathbf{t})$ denotes $(\tau(t_1), \ldots, \tau(t_n))$;
   (b) if $\mathbf{t}$ is a vector of terms, then $\tau(p(\mathbf{t})) = \delta_p^\tau(\tau(\mathbf{t}))$; $\tau(t_1 \doteq t_2) = \tau(t_1) \doteq \tau(t_2)$;
   (c) τ is extended recursively by $\tau(\varphi \wedge \psi) = \tau(\varphi) \wedge \tau(\psi)$, $\tau(\varphi \vee \psi) = \tau(\varphi) \vee \tau(\psi)$, $\tau(\varphi \rightarrow \psi) = \tau(\varphi) \rightarrow \tau(\psi)$, $\tau(\neg\varphi) = \neg\tau(\varphi)$, $\tau(\forall x \varphi) = \forall x \tau(\varphi)$ and $\tau(\exists x \varphi) = \exists x \tau(\varphi)$.

Any interpretation τ of $\mathcal{L}_1$ in $\mathcal{L}_2$ induces a mapping F_τ from $\mathcal{L}_2$-structures to $\mathcal{L}_1$-structures: if $\mathcal{I} = \langle (D, I), I^h, I^t \rangle$, then $F_\tau(\mathcal{I}) = \langle (D, J), J^h, J^t \rangle$ is defined as follows:

[3] Ono's axiomatisation of $\mathbf{QHT}^s$ uses the constant domains axiom $\forall x(\alpha(x) \vee \beta) \rightarrow (\forall x \alpha(x) \vee \beta)$, as well as alternative axioms for propositional here-and-there, viz. $p \vee (p \rightarrow (q \vee \neg q))$ and $(p \rightarrow q) \vee (q \rightarrow p) \vee (p \leftrightarrow \neg q)$. However, the axioms given here are equivalent to Ono's.

[4] 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.

– For every $a \in C_1$, $J(a) = I(\tau(a))$
– $p(\mathbf{t}) \in J^w$ iff $\mathcal{I}, w \models \delta_p^\tau(\tau(\mathbf{t}))$

It is easy to check that for any $\mathcal{L}_1$-sentence φ and any $w \in \{h, t\}$:

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

and therefore

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

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 $\mathcal{L}_1$-sentence φ,

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

In this case it is evident that

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

Generally speaking the map F_τ associated with an interpretation τ of $\mathcal{L}_1$ in $\mathcal{L}_2$ does not preserve the ordering $\trianglelefteq$ between $\mathcal{L}_2$-structures. However the following properties are easy to check and will be useful later:

Lemma 1. *Let τ be an interpretation of $\mathcal{L}_1$ in $\mathcal{L}_2$, and let $\mathcal{I}$ be a total $\mathcal{L}_2$-structure. Then (i) $F_\tau(\mathcal{I})$ is a total $\mathcal{L}_1$-structure; and (ii) if $\mathcal{I}' \trianglelefteq \mathcal{I}$, then $F_\tau(\mathcal{I}') \trianglelefteq F_\tau(\mathcal{I})$.*

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 5. *Let τ be an interpretation of Π_1 in Π_2. Then the following are equivalent.*
(i) For every $\mathcal{L}_2$-formula $\psi(\mathbf{x})$ there is an $\mathcal{L}_1$-formula $\varphi(\mathbf{x})$ such that $\Pi_2 \models \forall\mathbf{x}(\psi(\mathbf{x}) \leftrightarrow \tau(\varphi(\mathbf{x})))$; 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, $\Pi_2 \models \forall\mathbf{x}(\psi(\mathbf{x}) \leftrightarrow \tau(\sigma(\psi(\mathbf{x}))))$.
(iii) The mapping F_τ from models of Π_2 into models of Π_1 is an injection.

An interpretation satisfying any of (i)-(iii) of Proposition 5 is said to be *surjective*. Such interpretation preserve the property of being an equilibrium model, in the following sense.

Proposition 6. *Let τ be a surjective interpretation of Π_1 in Π_2. For any model $\mathcal{M}$ of Π_2, if $F_\tau(\mathcal{M})$ is an equilibrium model of Π_1 then $\mathcal{M}$ is an equilibrium model of Π_2.*

If τ is a surjective and a faithful interpretation, then it 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) in Prop. 5, is also bijective. 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 $\mathcal{L}_1$-structures $\mathcal{I}$ to $\mathcal{L}_2$-structures $F_\sigma(\mathcal{I})$ 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 structures (even in the classical case).

4.2 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.

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 $\Pi_2 \cup \overline{\tau} \equiv \Pi_1 \cup \overline{\sigma} \equiv \Pi$.*

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 1. *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 9. *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 $F_\tau(\mathcal{M}) \models_e \Pi_1$.*

Clearly, condition D5 is satisfied and the presence of an inverse interpretation provides the sense in which the correspondence between Π_1 and Π_2 is idempotent. Lastly we consider D6.

Proposition 10. *Let Π_1 and Π_2 be theories in $\mathcal{L}_1$ and $\mathcal{L}_2$ respectively synonymous wrt τ and σ. Let Π a set of $\mathcal{L}_1$-formulas. Then $\Pi_1 \cup \Pi$ is synonymous with $\Pi_2 \cup \tau(\Pi)$ wrt τ and σ.*

5 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 [20]. The classical version of Proposition 6 is essentially contained in [3], though a more detailed statement and proof can be found in [40]. Outside the field of mathematics, the classical theory of interpretability and definitional equivalence was extended and applied to empirical forms of knowledge in [29, 34, 30]; see also [41] 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 depends, see eg. [24–26]. In the context of nonmonotonic logic programming the study of different kinds of equivalence between programs is relatively new (see references in section 1). Until now the case of programs in different languages has only been considered in [36]. There has been some discussion of the role and properties of definitions in ASP in [17, 12],.

6 Concluding Remarks

We have argued that formal approaches to intertheory relations developed for mathematical and scientific knowledge can be applied to systems of logic programming and nonmonotonic reasoning used for practical problem solving and knowledge representation in AI. In particular, we have described how the theory of interpretability and definitional equivalence can be applied in the context of first-order logic programs under answer set semantics and nonmonotonic theoreis in the system of quantified 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 non-classical 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 (*quantified here-and-there*). This allows us to define a robust or modular concept of equivalence across different languages.

Several avenues are left open for future exploration. For example, one might want to study other kinds of interpretability relations, eg where the formula δ_p^τ defining a predicate p may contain additional parameters, or where the semantic mapping F_τ may relate models with different domains. 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 [9, 35].

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. *Logic and Structure.* Springer, 1983..
7. D. van Dalen. Intuitionistic logic. In *Handbook of Philosophical Logic, Volume III: Alternatives in Classical Logic*, Kluwer, Dordrecht, 1986.
8. 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.
9. 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.
10. T. Eiter, P. Trazler & S. Woltran. An Implementation for Recognizing Rule Replacements in Non-Ground Answer Set Programs. in M. Fischer *at al* (eds), *Jelia 2006*, LNAI 4160, Springer, 2006, 477-480.
11. H. Enderton. *A Mathematical Introduction to Logic* Academic Press, 1972.
12. S. Erdogem and V. Lifschitz. Definitions in Answer Set Programming in V. Lifschitz and I. Niemelä (eds), *Proceedings LPNMR 2004*, Springer LNAI 2923, 2004.
13. K. Gödel. Zum intuitionistischen aussagenkalkül. *Anzeiger der Akademie der Wissenschaften Wien, mathematisch, naturwissenschaftliche Klasse*, 69:65–66, 1932.
14. A. Heyting. Die formalen regeln der intuitionistischen logik. *Sitzungsberichte der Preussischen Akademie der Wissenschaften, Physikalisch-mathematische Klasse*, pages 42–56, 1930.
15. 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.
16. Paolo Ferraris, Joohyung Lee, and Vladimir Lifschitz. A new perspective on stable models. IJCAI 2007.
17. P. Ferraris and V. Lifschitz. Weight Constraints as Nested Expressions *Theory and Practice of Logic Programming*, to appear, 2004.
18. D. Gabbay and L. Maksimova. *Interpolation and Definability: Modal and Intuitionistic Logics.* Ser. Oxford Logic Guides:46, Oxford University Press, Oxford, 2005
19. 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.
20. S. Kanger. Equivalent Theories. *Theoria*, 38 (1972), 1-6.
21. 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.
22. V. Lifschitz, D. Pearce, and A. Valverde. Strongly equivalent logic programs. *ACM Transactions on Computational Logic*, 2(4):526–541, 2001.
23. V. Lifschitz, D. Pearce, and A. Valverde. A Characterization of Strong Equivalence for Logic Programs with Variables *Proceedings LPNMR 07*, Springer, 2007, to appear.
24. L. Maksimova. Interpolation in superintuitionistic predicate logics with equality. *Algebra and Logic*, 36:543-561, 1997.
25. L.Maksimova. Interpolation in superintuitionistic and modal predicate logics with equality. In: M.Kracht, M. de Rijke, H.Wansing and M.Zakharyaschev, editors. *Advances in Modal Logic, Volume I*, pages 133-141. CSLI Publications, Stanford, 1998.
26. L. Maksimova. Intuitionistic logic and implicit definability. *Annals of Pure and Applied Logic* 105(1-3): 83-102, 2000

27. E. Oikarinen and T. Janhunen. Verifying the Equivalence Logic Programs in the Disjunctive Case. In V. Lifschitz and I. Niemelä (eds), *Logic Programming and Nonmonotonic Reasoning*, LPNMR 2004, Springer LNAI 2923, 2004.
28. H. Ono. Model extension theorem and Craig's interpolation theorem for intermediate predicate logics. *Reports on Mathematical Logic* 15 (1983), 41-58.
29. D. Pearce. Some Relations between Empirical Systems. *Epistemologia* 4 (1981), 363-380.
30. D. Pearce. *Roads to Commensurability* Kluwer (Synthese Library Vol. 187),1987.
31. D. Pearce. A new logical characterization of stable models and answer sets. In *Non-Monotonic Extensions of Logic Programming, NMELP 96*, Springer LNCS 1216, 1997, pages 57–70.
32. D. Pearce. Equilibrium Logic. *Annals of Mathematics and AI*, 2006.
33. 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.
34. D. Pearce and V. Rantala. New Foundations for Metascience. *Synthese*, 1985.
35. D. Pearce and A. Valverde. Uniform Equivalence for Equilibrium Logic and Logic Programs. In V. Lifschitz and I. Niemelä (eds) *Logic Programming and Nonmonotonic Reasoning*, LPNMR 2004, Springer LNAI 2923, 2004.
36. D. Pearce and A. Valverde. Synonymous Theories in Answer Set Programming and Equilibrium Logic. In Proceedings ECAI 2004, 2004.
37. David Pearce and Agustin Valverde. Towards a first order equilibrium logic for nonmonotonic reasoning. In *Proceedings of European Conference on Logics in Artificial Intelligence (JELIA)*, pages 147–160, 2004.
38. David Pearce and Agustin Valverde. A first order nonmonotonic extension of constructive logic. *Studia Logica*, 80:323–348, 2005.
39. David Pearce and Agustin Valverde. Quantified Equilibrium Logic and the First Order Logic of Here-and-There. Technical Report, Univ. Rey Juan Carlos, 2006,
40. C. Pinter Properties Preserved under Definitional Equivalence and Interpretations. In *Zeitschr. f. math. Logik und Grundlagen d. Math.*, 24: 481–488, 1978.
41. V. Rantala. *Explanatory Translation*. Kluwer (Synthese Library Vol 312), 2002.
42. P. Simons, I. Niemelä, and T. Soininen. Extending and implementing the stable model semantics. *Artificial Intellingence*, 138(1–2):181–234, 2002.
43. 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.
44. 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.
45. Stefan Woltran. *LPNMR 2007*, to appear.