Some Equivalence Concepts for Hybrid Theories *

Michael Fink[1] and David Pearce[2]

[1] Vienna University of Technology, Austria, fink@kr.tuwien.ac.at
[2] Universidad Politécnica de Madrid, Spain, david.pearce@upm.es

Abstract. Hybrid theories arise out of the combination of ontologies with non-monotonic rules and are receiving increasing attention in KR. Popular semantics for such theories are those based on answer set programming (ASP). The present paper is a preliminary attempt to combine some concepts of equivalence from the two areas (ontologies and ASP) in order to initiate the study of equivalence and modularity for hybrid theories.

1 Introduction

A growing field of research in knowledge technologies concerns the ways and means of amalgamating description logics and nonmonotonic programs in order to combine rule-based reasoning with ontologies. There have been several different proposals for merging these languages into a more tightly or a more loosely integrated semantical framework. The best known methods are probably those based on stable model semantics or answer set programming (ASP); these have given rise to HEX-programs, dl-programs and *DL+log* or *hybrid knowledge bases*. The idea of hybrid knowledge base was introduced by [21] and further developed in [22, 23, 12]; for dl-programs see eg. [5]. The practical importance of such hybrid systems is also underlined by their current use in major European research projects.[3]

If hybrid systems are to become a successful, practical tool in knowledge based reasoning, it is essential that modularity and related notions are addressed. Knowing for instance in which contexts one hybrid theory can be replaced by another without loss is important for formalising knowledge and for transforming and simplifying theories. For each of the relevant underlying technologies, modularity is an active area of study. For example, there has been a strong interest recently in developing logical treatments of modularity for ontologies reconstructed in description logics (DL). An approach based on conservative extensions and entailment and difference concepts can be found in [10, 11, 9]. In this case a concept of equivalence and relativised equivalence between DL ontologies is obtained as a limiting case when the difference relation is zero.

On the other hand, in nonmonotonic logic programming, in particular in ASP, work on (strong) equivalence relations between programs began already in [13], and the study of variations of this basic concept has formed a very active area of research since, especially as a tool for program transformation and optimisation. Recently focus has turned

* Partially supported by the MCICINNprojects TIN2006-15455-CO3 and CSD2007-00022, WWTF project ICT08-020, and FWF project P20840.

[3] See for instance the ONTORULE integrated FP7 project: http://ontorule-project.eu/.

from propositional programs to theories and programs in first-order logic [8, 14]. This is important for the study of hybrid theories where first order languages are needed.

We use the term *hybrid theory* to refer to any of the standard approaches to combining classical theories and ontologies with rules. In this paper, we focus on two types of hybrid theories: those we call *hybrid knowledge bases (HKBs)* following [21], and those known as *dl-programs* [5], both kinds of theories being interpreted under answer set semantics. While notions of equivalent theory are studied both in ASP and in the area of ontologies, it is clear that additional degrees of freedom arise when we consider logical relations between hybrid theories. In the case of HKBs we define several different equivalence concepts and provide semantical characterisations of them. In the case of dl-programs, semantic integration between the different components is looser, and as a result strong logical relations between theories appear harder to characterise. Nevertheless we take some initial steps towards combining some of the different equivalence concepts that have been defined for ontologies and for logic programs.

2 Review of Quantified Equilibrium Logic and Answer Sets

In this paper we deal with function-free first order languages $\mathcal{L} = \langle C, P\rangle$ built over a set of *constant* symbols, C, and a set of *predicate* symbols, P. We assume a single negation symbol, '$\neg$', together with the usual connectives and quantifiers, $\wedge, \vee, \rightarrow, \exists, \forall$. We shall also assume that $\mathcal{L}$ contains the constants $\top$ and $\bot$ and, where convenient, we regard $\neg\varphi$ as an abbreviation for $\varphi \rightarrow \bot$. In other respects we follow the treatment of [19]. The sets of $\mathcal{L}$-formulas, $\mathcal{L}$-sentences and atomic $\mathcal{L}$-sentences are defined in the usual way. We work in a non-classical logic called *Quantified Here-and-There Logic with static domains and decidable equality*. For reasons of space we give here just a short summary. A complete axiomatisation and more detailed description of this logic can be found in [14] where the logic is denoted by $\mathbf{SQHT}^{=}$. In terms of satisfiability and validity this logic is equivalent to the logic previously introduced in [18]. To simplify notation we drop the labels for static domains and equality and refer to this logic simply as quantified here-and-there, $\mathbf{QHT}$.

The semantics of $\mathbf{QHT}$ is given in terms of intuitionistic Kripke models, see [3], with two notable exceptions. One concerns equality: we regard equality as decidable and as satisfying the axiom $\forall x \forall y((x = y) \vee \neg(x = y))$. Furthermore, we suppose a logic with constant or static domains; in other words within a given Kripke model the same set of individuals populates each world. In addition, $\mathbf{QHT}$ is complete for very simple Kripke models, those possessing just two worlds, sometimes labelled h ("here") and t ("there"), ordered by $h \leq t$.

We use the following notation. If D is a non-empty set, we denote by $At(D, P)$ the set of ground atomic sentences of $\langle D, P\rangle$. By an $\mathcal{L}$-interpretation I over a set D we mean a subset of $At(D, P)$. A $\mathbf{QHT}(\mathcal{L})$*-structure* can therefore be regarded as a tuple $\mathcal{M} = \langle (D, \sigma), I_h, I_t\rangle$. where I_h, I_t are $\mathcal{L}$-interpretations over D such that $I_h \subseteq I_t$ and $\sigma\colon C \cup D \rightarrow D$ is a mapping, called the *assignment*, such that $\sigma(d) = d$ for all $d \in D$. Evidently, $\langle (D, \sigma), I_h\rangle$ and $\langle (D, \sigma), I_t\rangle$ are classical $\mathcal{L}$-structures. Given an interpretation we let $\sigma|_C$ denote the restriction of the assignment σ to constants in C.

For a $\mathbf{QHT}(\mathcal{L})$-structure $\mathcal{M} = \langle(D, \sigma), I_h, I_t\rangle$ and $\mathcal{L}' \subset \mathcal{L}$, we denote the restriction of $\mathcal{M}$ to the sublanguage $\mathcal{L}'$ by $\mathcal{M}|_{\mathcal{L}'} = \langle(D, \sigma|_{\mathcal{L}'}), I_h|_{\mathcal{L}'}, I_t|_{\mathcal{L}'}\rangle$.

The truth relation for $\mathbf{QHT}$ is denoted by '$\models$' and truth of a sentence in a model is defined as follows: $\mathcal{M} \models \varphi$ iff $\mathcal{M}, w \models \varphi$ for each $w \in \{h, t\}$. A sentence φ is a consequence of a set of sentences Γ, denoted $\Gamma \models \varphi$, if every model of Γ is a model of φ. In a model $\mathcal{M}$ we also use the symbols H and T, possibly with subscripts, to denote the interpretations I_h and I_t respectively; so, an $\mathcal{L}$-structure may be written in the form $\langle U, H, T\rangle$, where $U = (D, \sigma)$. A structure $\langle U, H, T\rangle$ is called *total* if $H = T$, whence it is equivalent to a classical structure.

An answer semantics for arbitrary first-order formulas can be defined using the quantified variant of equilibrium logic [15, 16] that we denote by $\mathbf{QEL}$. As in the propositional case, this is based on a suitable notion of minimal model as follows.

Definition 1 ([17, 18]). *Let Γ be a set of $\mathcal{L}$-sentences. An* equilibrium model *or* answer set *of Γ is a total model $\mathcal{M} = \langle(D, \sigma), T, T\rangle$ of Γ such that there is no model of Γ of the form $\langle(D, \sigma), H, T\rangle$ where H is a proper subset of T.*

2.1 Hybrid Knowledge Bases

Hybrid knowledge bases combine classical theories with nonmonotonic rules interpreted under answer set semantics. Formally, a *hybrid knowledge base* $\mathcal{K} = (\mathcal{T}, \mathcal{P})$ over the function-free language $\mathcal{L} = \langle C, P_{\mathcal{T}} \cup P_{\mathcal{P}}\rangle$ consists of a classical first-order theory $\mathcal{T}$ (also called the *structural* part of $\mathcal{K}$) over the language $\mathcal{L}_{\mathcal{T}} = \langle C, P_{\mathcal{T}}\rangle$ and a program $\mathcal{P}$ (also called *rules* part of $\mathcal{K}$) over the language $\mathcal{L}$, where $P_{\mathcal{T}} \cap P_{\mathcal{P}} = \emptyset$, i.e. $\mathcal{T}$ and $\mathcal{P}$ share a single set of constants, and the predicate symbols allowed to be used in $\mathcal{P}$ are a superset of the predicate symbols in $\mathcal{L}_{\mathcal{T}}$. Intuitively, the predicates in $\mathcal{L}_{\mathcal{T}}$ are interpreted classically, whereas the predicates in $\mathcal{L}_{\mathcal{P}}$ are interpreted nonmonotonically under a generalised form of answer set semantics. With $\mathcal{L}_{\mathcal{P}} = \langle C, P_{\mathcal{P}}\rangle$ we denote the restricted language of $\mathcal{P}$.

There are several distinct variants of hybrid knowledge bases [21–23, 12]. Their differences depend mainly on what kind of nonmonotonic rules are allowed in $\mathcal{P}$ and what safety conditions are imposed in order to ensure decidability. However the semantics of these different variants is quite similar in each case. Essentially classical models of $\mathcal{T}$ are used as candidate interpretations for the $\mathcal{L}_{\mathcal{T}}$ predicates appearing in $\mathcal{P}$. Using these models, $\mathcal{P}$ is then preprocessed by reduction to a program without $\mathcal{L}_{\mathcal{T}}$ predicates, converted into ground form and its answer sets computed. A uniform stability condition on the two components, classical and nonmonotonic, defines the semantics or intended models.

Equilibrium models define an equivalent semantics for hybrid knowledge bases in a simpler and more direct fashion. Given a hybrid KB $\mathcal{K} = (\mathcal{T}, \mathcal{P})$ we call $\mathcal{T} \cup \mathcal{P} \cup st(\mathcal{T})$ the *stable closure* of $\mathcal{K}$, where $st(\mathcal{T}) = \{\forall x(p(x) \vee \neg p(x)) \colon p \in \mathcal{L}_{\mathcal{T}}\}$.

Definition 2. *Let $\mathcal{K} = (\mathcal{T}, \mathcal{P})$ be a hybrid knowledge base. $\mathcal{M} = \langle U, T, T\rangle$ is said to be an* equilibrium model *of $\mathcal{K} = (\mathcal{T}, \mathcal{P})$ if it is an equilibrium model of the stable closure of $\mathcal{K}$, ie. of $\mathcal{T} \cup \mathcal{P} \cup st(\mathcal{T})$.*

The appropriateness of this definition is established in [1] where it is shown that when $\mathcal{K} = (\mathcal{T}, \mathcal{P})$ is a g-hybrid knowledge base in the sense of [12] or an r$^+$-hybrid knowledge base in the sense of [22], Definition 2 yields an equivalent semantics. Other semantic variants from [21, 23] can also be modelled easily in this framework.

3 Some equivalence concepts for hybrid knowledge bases

As usual, we say that two first-order theories Π_1 and Π_2 are *classically equivalent*, in symbols $\Pi_1 \equiv_c \Pi_2$, if and only if their classical models coincide. They are called *answer-set equivalent*, symbolically $\Pi_1 \equiv_a \Pi_2$, if and only if they have the same equilibrium models, i.e., answer sets.

The study of strong equivalence for logic programs and nonmonotonic theories was initiated in [13]. It has since become an important tool in ASP as a basis for program transformation and optimisation. In equilibrium logic two (first-order) theories Π_1 and Π_2 are *strongly equivalent*, $\Pi_1 \equiv_s \Pi_2$, if and only if for any theory Π, $\Pi_1 \cup \Pi$ and $\Pi_2 \cup \Pi$ have the same equilibrium models [14, 19]. Under this definition we have:

Theorem 1 ([14, 19]). *Two (first-order) theories Π_1 and Π_2 are strongly equivalent if and only if they are equivalent in* $\mathbf{QHT}$.

Let us say that two HKBs, $\mathcal{K}_1$ and $\mathcal{K}_2$, are *equivalent*, in symbols $\mathcal{K}_1 \equiv \mathcal{K}_2$ iff they have the same equilibrium models. Following this we can introduce three different concepts of strong equivalence. We make the restriction that these concepts apply to knowledge bases sharing the same structural language; this assumption will be left implicit throughout.[4] First, we say that $\mathcal{K}_1$ and $\mathcal{K}_2$ are strongly equivalent, if for any $\mathcal{K} = (\mathcal{T}, \mathcal{P})$, $\mathcal{K}_1 \cup \mathcal{K} \equiv \mathcal{K}_2 \cup \mathcal{K}$, where union is defined in the obvious way as the unions of the structural and the program parts. Two special cases of strong equivalence arise if we restrict attention to just one of the components. Let us say that $\mathcal{K}_1$ and $\mathcal{K}_2$ are $\mathcal{P}$-equivalent if for all $\mathcal{P}$, $\mathcal{K}_1 \cup \mathcal{P} \equiv \mathcal{K}_2 \cup \mathcal{P}$, and $\mathcal{T}$-equivalent if for $\mathcal{T}$, $\mathcal{K}_1 \cup \mathcal{T} \equiv \mathcal{K}_2 \cup \mathcal{T}$. Evidently, since we assume that $\mathcal{K}_1$ and $\mathcal{K}_2$ share the same structural language, this will also be true of their extensions by new KBs, theories or programs. Our main goal is to give logical characterisations of $\mathcal{P}$-equivalence and $\mathcal{T}$-equivalence.

3.1 $\mathcal{P}$-equivalence

Let us first analyse these concepts for the case that no syntactic restrictions are imposed on the theories and programs in $\mathcal{K}_1$, $\mathcal{K}_2$ and their extensions. Evidently, if two knowledge bases are strongly equivalent, they are also $\mathcal{P}$-equivalent and $\mathcal{T}$-equivalent. In the former case we can strengthen this observation.

Proposition 1. *Two hybrid KBs. $\mathcal{K}_1$ and $\mathcal{K}_2$, are strongly equivalent if and only if they are $\mathcal{P}$-equivalent.*

[4] Intuitively the reason for this is that the structural language $\mathcal{L}_{\mathcal{T}}$ associated with a hybrid knowledge base $\mathcal{K} = (\mathcal{T}, \mathcal{P})$ is part of its identity or 'meaning'. Precisely the predicates in $\mathcal{L}_{\mathcal{T}}$ are the ones treated classically. Without the assumption of a common structural language, the natural properties expressed in Corollary 1 (a) and (1) below would no longer hold.

Evidently this holds because we impose no restrictions on the syntax of the nonmonotonic part. Another straightforward consequence is that two strongly equivalent logic programs cannot be separated by adding any structural knowledge component.

Proposition 2. *Suppose $\mathcal{P}_1$ and $\mathcal{P}_2$ are strongly equivalent programs. Then for any $\mathcal{T}$, $(\mathcal{T}, \mathcal{P}_1)$ and $(\mathcal{T}, \mathcal{P}_2)$ are equivalent HKBs. Hence they are also strongly equivalent.*

Until further notice, let us suppose that $\mathcal{K}_1 = (\mathcal{T}_1, \mathcal{P}_1)$ and $\mathcal{K}_2 = (\mathcal{T}_2, \mathcal{P}_2)$ are hybrid KBs sharing a common structural language $\mathcal{L}$.

Proposition 3 ([2]). *$\mathcal{K}_1$ and $\mathcal{K}_2$ are strongly equivalent if and only if $\mathcal{T}_1 \cup st(\mathcal{T}_1) \cup \mathcal{P}_1$ and $\mathcal{T}_2 \cup st(\mathcal{T}_2) \cup \mathcal{P}_2$ are logically equivalent in $\mathbf{QHT}$.*

We mention some conditions to test for strong equivalence and non-equivalence.

Corollary 1 ([2]). *(a) $\mathcal{K}_1$ and $\mathcal{K}_2$ are strongly equivalent if $\mathcal{T}_1$ and $\mathcal{T}_2$ are classically equivalent and $\mathcal{P}_1$ and $\mathcal{P}_2$ are equivalent in $\mathbf{QHT}$. (b) $\mathcal{K}_1$ and $\mathcal{K}_2$ are not strongly equivalent if $\mathcal{T}_1 \cup \mathcal{P}_1$ and $\mathcal{T}_2 \cup \mathcal{P}_2$ are not equivalent in classical logic.*

Special cases of strong equivalence arise when hybrid KBs are based on the same classical theory, say, or share the same rule base. That is, $(\mathcal{T}, \mathcal{P}_1)$ and $(\mathcal{T}, \mathcal{P}_2)$ are strongly equivalent if $\mathcal{P}_1$ and $\mathcal{P}_2$ are $\mathbf{QHT}^s_=$-equivalent. Analogously:

$(\mathcal{T}_1, \mathcal{P})$ and $(\mathcal{T}_2, \mathcal{P})$ are strongly equivalent if $\mathcal{T}_1$ and $\mathcal{T}_2$ are classically equivalent. (1)

3.2 $\mathcal{T}$-equivalence

To study $\mathcal{T}$-equivalence we make use of the notion of uniform equivalence, which has been characterised for first-order theories in [8]. Let us say that an $\mathcal{L}$- sentence φ is *factual* if it contains no occurrences of implication, other than possibly in the form $\alpha \rightarrow \bot$. Then, two theories Π_1 and Π_2 are called *uniformly equivalent*, symbolically $\Pi_1 \equiv_u \Pi_2$, if and only if for any factual theory Π $\Pi_1 \cup \Pi$ and $\Pi_2 \cup \Pi$ have the same equilibrium models.

Uniform equivalence for theories is characterised in terms of countermodels: An $\mathcal{L}$-structure $\mathcal{M} = \langle U, H, T \rangle$ is called an *$\mathcal{L}$-countermodel* of a theory Π if and only if $\mathcal{M} \not\models \Pi$; it is called *total* if $H = T$. Furthermore, a total $\mathcal{L}$-structure $\mathcal{M} = \langle U, T, T \rangle$ is called *total-closed* in a set S of $\mathcal{L}$-structures if $\langle U, H, T \rangle \in S$ for every $H \subseteq T$. An $\mathcal{L}$-structure $\mathcal{M} = \langle U, H, T \rangle$ is *there-closed* in a set S of $\mathcal{L}$-structures if $\langle U, H', T \rangle \in S$ for every $H \subseteq H' \subset T$, respectively for $H = H' = T$.

Total models and non-total countermodels are joined to form so-called equivalence structures. An $\mathcal{L}$-countermodel $\langle U, H, T \rangle$ of a theory Π is called a *here $\mathcal{L}$-countermodel* of Π if $\langle U, T \rangle$ is a classical model of Π. An $\mathcal{L}$-structure $\langle U, H, T \rangle$ is an *equivalence $\mathcal{L}$-structure* of a theory Π, if it is a total model of Π or a here $\mathcal{L}$-countermodel.

By means of equivalence structures, the relevant notions of equivalence can be characterised uniformly. Given a theory Π over $\mathcal{L}$, let $E_s(\Pi)$ denote the set of its equivalence $\mathcal{L}$-structures, and let $E_c(\Pi)$ be the total $\mathcal{L}$-structures in $E_s(\Pi)$. Furthermore, $E_a(\Pi)$ is the set of total-closed $\mathcal{L}$-structures in $E_s(\Pi)$, and $E_u(\Pi)$ is the set of there-closed $\mathcal{L}$-structures in $E_s(\Pi)$.

Lemma 1 (Corollary 2 in [8]). *Let Π_1 and Π_2 be first-order theories over $\mathcal{L}$, then $\Pi_1 \equiv_e \Pi_2$ iff $E_e(\Pi_1) = E_e(\Pi_2)$ for $e \in \{c, a, s, u\}$.*

As for strong equivalence, we aim at a characterisation of $\mathcal{T}$-equivalence between HKBs by reference to their stable closures. A simple first observation is the following:

Lemma 2. *Let $\mathcal{K} = (\mathcal{T}, \mathcal{P})$ be a HKB, and let $\mathcal{M} = \langle U, H, T\rangle$ be an $\mathcal{L}$-structure. Then, $\mathcal{M} \models st(\mathcal{T})$ if and only if $H|_{\mathcal{L}_\mathcal{T}} = T|_{\mathcal{L}_\mathcal{T}}$.*

Due to stable closure, it is sufficient to consider $\mathcal{L}$-structures such that $H|_{\mathcal{L}_\mathcal{T}} = T|_{\mathcal{L}_\mathcal{T}}$. In order to decide whether such $\mathcal{L}$-structures are models of the structural part together with the stable closure, it is sufficient to consider the respective total model.

Lemma 3. *Let $\mathcal{K} = (\mathcal{T}, \mathcal{P})$ be a HKB, and let $\mathcal{M} = \langle U, H, T\rangle$ be an $\mathcal{L}$-structure, such that $H|_{\mathcal{L}_\mathcal{T}} = T|_{\mathcal{L}_\mathcal{T}}$. Then, $\mathcal{M} \models \mathcal{T} \cup st(\mathcal{T})$ if and only if $\langle U, T, T\rangle \models \mathcal{T} \cup st(\mathcal{T})$.*

Due to the classical interpretation of the structural theory in hybrid KBs, $\mathcal{T}$-equivalence amounts to a particular form of uniform equivalence between the respective stable closures, similar to relativised notions of equivalence. Rather than relativising to a sublanguage, due to the stable closure, it is sufficient to consider $\mathcal{L}$-structures such that H and T coincide on atoms over predicates from $\mathcal{L}$ appearing in H.

Definition 3. *Given a HKB, $\mathcal{K} = (\mathcal{T}, \mathcal{P})$, let $E_u^{\mathcal{L}}(\mathcal{K}) = \{\langle U, H, T\rangle \mid \langle U, H, T\rangle \in E_u(\mathcal{T} \cup \mathcal{P} \cup st(\mathcal{T})), H = T|_{\mathcal{L}_H}\}$, where $\mathcal{L}_H = (C, P_H)$ and $P_H = \{p \mid p(t_1, \ldots, t_n) \in H\}$.*

Then, $\mathcal{T}$-equivalence between HKBs can be characterised as follows.

Proposition 4. *Two HKBs, $\mathcal{K}_1$ and $\mathcal{K}_2$, are $\mathcal{T}$-equivalent if and only if $E_u^{\mathcal{L}}(\mathcal{K}_1) = E_u^{\mathcal{L}}(\mathcal{K}_2)$.*

4 Some remarks on dl-programs

We consider dl-programs [5, 6], without restricting the classical part, which is combined with logic program rules, to be a description logic. Rather we consider arbitrary function-free first-order theories and allow for arbitrary formulas as queries. Moreover, disjunction is allowed in rule heads, whereas, like for hybrid knowledge bases, we require that the classical theory and the logic program share a single set of constants.

Again let $\mathcal{L}_\mathcal{T} = \langle C, P_\mathcal{T}\rangle$ and $\mathcal{L}_\mathcal{P} = \langle C, P_\mathcal{P}\rangle$ be function-free first-order languages, such that $P_\mathcal{T} \cap P_\mathcal{P} = \emptyset$. Symbols in $P_\mathcal{T}$, respectively in $P_\mathcal{P}$, are called *classical predicates* and *rule predicates*, respectively. A *dl-atom* is of the form

$$DL[S_1 op_1 p_1, \ldots, S_m op_m p_m; Q](t_1, \ldots, t_n), \tag{2}$$

where $S_i \in P_\mathcal{T}$ and $p_i \in P_\mathcal{P}$ are k-ary predicate symbols, $op_i \in \{\uplus, \cup\!\!\!\!-, \cap\!\!\!\!-\}$, Q is an n-ary classical predicate or a formula in $\mathcal{L}_\mathcal{T}$ with n free variables, and $t_1, \ldots, t_n$ are terms. A *dl-rule* is like a logic program rule of the form

$$b_1 \wedge \ldots \wedge b_m \wedge \neg b_{m+1} \ldots \neg \wedge \neg b_n \rightarrow h_1 \vee \ldots \vee h_l$$

with the restriction that head atoms $h_1, \ldots, h_l$ are equality-free atoms of $\mathcal{L}_\mathcal{P}$, and body atoms $b_1, \ldots, b_n$ are either atoms of $\mathcal{L}_\mathcal{P}$ or dl-atoms. The positive body $\{b_1, \ldots, b_m\}$ and the negative body $\{b_{m+1}, \ldots, b_n\}$ of a dl-rule r are denoted by $B^+(r)$ and $B^-(r)$, respectively. A *dl-program* over $\mathcal{L} = \langle C, P_\mathcal{T} \cup P_\mathcal{P} \rangle$ is a pair $\mathcal{D} = (\mathcal{T}, \mathcal{P})$, where $\mathcal{T}$ is a finite first-order theory over $\mathcal{L}_\mathcal{T}$ and $\mathcal{P}$ is a set of dl-rules.

Turning to the semantics of dl-programs, let us denote the set of dl-atoms in a rule r, respectively in a set of rules $\mathcal{P}$, by $DL(r)$ and $DL(\mathcal{P})$, respectively.

A Herbrand structure $\mathcal{M} = \langle U, I \rangle$ (with $U = (D, \sigma)$) is a *model* of a literal l under $\mathcal{T}$ if $l \in I$. It is a model of a ground dl-atom of the form (2) under $\mathcal{T}$ if $\mathcal{T} \cup \bigcup_{i=1}^{m} A_i(I) \models Q(t_1, \ldots, t_n)$, where

- $A_i(I) = \{S_i(\boldsymbol{e}) \mid p_i(\boldsymbol{e}) \in I\}$, for $op_i = \uplus$,
- $A_i(I) = \{\neg S_i(\boldsymbol{e}) \mid p_i(\boldsymbol{e}) \in I\}$, for $op_i = \cup\!\!\!\!-$,
- $A_i(I) = \{\neg S_i(\boldsymbol{e}) \mid p_i(\boldsymbol{e}) \notin I\}$, for $op_i = \Cap$,

and $\boldsymbol{e} = e_1, \ldots, e_n$ are ground terms.

As usual, $\mathcal{M}$ is a *model* of a ground dl-rule r under $\mathcal{T}$ if $\mathcal{M}$ is a model of some $h_i \in \{h_1, \ldots, h_l\}$ under $\mathcal{T}$, whenever $\mathcal{M}$ is a model of all $b_i \in \{b_1, \ldots, b_m\}$ under $\mathcal{T}$ and it is no model of any$c_i \in \{c_1, \ldots, c_n\}$ under $\mathcal{T}$. $\mathcal{M}$ is a model of a dl-program $\mathcal{D} = (\mathcal{T}, \mathcal{P})$ if $\mathcal{M}$ is a model of every $r \in gr_U(\mathcal{P})$ under $\mathcal{T}$.

Furthermore, given a dl-program $\mathcal{D} = (\mathcal{T}, \mathcal{P})$, the *weak dl-transform* of $\mathcal{P}$ relative to $\mathcal{T}$ and a model $\mathcal{M}$ of $\mathcal{P}$, denoted $w\mathcal{P}_\mathcal{T}^\mathcal{M}$, is the logic program obtained from $gr_U(\mathcal{P})$ by deleting

- each $r \in gr_U(\mathcal{P})$ such that either $\mathcal{M}$ is not a model of some $\alpha \in B^+(r) \cap DL(r)$, or a model of some $\alpha \in B^-(r)$, and
- all literals in $B^-(r) \cup (B^+(r) \cap DL(r))$ from each remaining $r \in gr_U(\mathcal{P})$.

If $\mathcal{M}$ is an answer set of the logic program $w\mathcal{P}_\mathcal{T}^\mathcal{M}$, then $\mathcal{M}$ is a *weak answer set* of $\mathcal{D}$.

Now assume that, $\mathcal{D} = (\mathcal{T}, \mathcal{P})$ has an associated set of ground dl-atoms $DL^+(\mathcal{P})$ known to be monotonic, and for any ground rule r, let $DL^?(r) = DL(r) \setminus DL^+(\mathcal{P})$. The *strong dl-transform* of $\mathcal{P}$ relative to $\mathcal{T}$ and a model $\mathcal{M}$ of $\mathcal{P}$, denoted $s\mathcal{P}_\mathcal{T}^\mathcal{M}$, is the logic program obtained from $gr_U(\mathcal{P})$ as before replacing $DL(r)$ by $DL^?(r)$.

If $\mathcal{M}$ is the least model of $(\mathcal{T}, s\mathcal{P}_\mathcal{T}^\mathcal{M})$, then $\mathcal{M}$ is a *strong answer set* of $\mathcal{D}$.

4.1 Equivalence for dl-programs

The study of strong equivalence concepts for dl-programs appears harder than in the case of HKBs. The main problem is that we do not yet have a unifying semantics like equilibrium logic. A good feature of dl-programs is that most (though not quite all) concepts of inseparability, equivalence and modularity for DL-ontologies are defined in syntactic (proof theoretic) terms. A difficulty of dl-programs, however, is the semantics of dl-atoms which depends on the ontology but also on the current candidate model. Nevertheless, we can start to make some simple observations.

A first point to note is that if two ordinary answer set programs are strongly equivalent they cannot be separated by additional dl-rules.

Proposition 5. *Let $\Pi_1.\Pi_2$ be two strongly equivalent logic programs. Let r be any dl-rule and let $(\mathcal{T},\mathcal{P}),(\mathcal{T},\mathcal{R})$ be dl-programs where $\mathcal{P} = \Pi_1 \cup \{r\}$ and $\mathcal{R} = \Pi_2 \cup \{r\}$. Then $(\mathcal{T},\mathcal{P})$ and $(\mathcal{T},\mathcal{R})$ are equivalent under the weak and the strong answer set semantics for dl-programs.*

The property continues to hold in the first order case and evidently for any set of additional dl-rules. Let us now consider an equivalence concept drawn from the area of ontologies reconstructed in description logics (DL). We assume the reader is familiar with the standard notions of TBox and ABox. In the papers [10, 11, 9] on modular ontologies there are several slightly different terminologies and notations. However, basically these works consider an ontology to be represented by a TBox, while a knowledge base is a combination of a TBox together with an ABox. We state here is a definition from [9]. To simplify notation we assume that some DL is given, while Σ is a language or vocabulary. Let $\mathcal{T}_1$ and $\mathcal{T}_2$ be TBoxes.

Definition 4. *The* Σ-query difference *between $\mathcal{T}_1$ and $\mathcal{T}_2$, in symbols* $\mathrm{Diff}_\Sigma(\mathcal{T}_1,\mathcal{T}_2)$*, is the set of pairs $(\mathcal{A}, q(x))$ where $\mathcal{A}$ is an ABox and $q(x) \in \Sigma$ is a query such that $(\mathcal{T}_1,\mathcal{A}) \not\models q(\mathbf{a})$ and $(\mathcal{T}_2,\mathcal{A}) \models q(\mathbf{a})$, for some tuple $\mathbf{a}$ of object names from $\mathcal{A}$. We say that $\mathcal{T}_1$ Σ-query entails $\mathcal{T}_2$ if* $\mathrm{Diff}_\Sigma(\mathcal{T}_1,\mathcal{T}_2) = \emptyset$.

By obvious extension we can say that $\mathcal{T}_1$ and $\mathcal{T}_2$ are *Σ-query equivalent* if each Σ-query entails the other. So they are equivalent for all ABoxes and Σ-queries. Let us turn to dl-programs and let us suppose for the moment that their classical part comprises an ontology or TBox, so a dl-program has the form $(\mathcal{T}, P)$ for some TBox, $\mathcal{T}$. Now the way in which a ground dl-atom is evaluated in an Herbrand interpretation I is similar to the effect of adding an ABox $\mathcal{A}$ to $\mathcal{T}$ and then checking whether a ground query $q(\mathbf{a})$ follows from $(\mathcal{T},\mathcal{A})$. This yields the following property.

Proposition 6. *Suppose that $\mathcal{T}_1$ and $\mathcal{T}_2$ are Σ-query equivalent TBoxes. Let P be any set of dl-rules all of whose dl-atoms are in Σ. Then the dl-programs $(\mathcal{T}_1, P)$ and $(\mathcal{T}_2, P)$ are equivalent under the weak and under the strong answer set semantics.*

We make the assumption above that the same syntactic class of queries is allowed in each case of TBoxes and dl-programs, for example arbitrary queries, conjunctive queries or some intermediate class.

Let us also ignore for the moment the two different answer set semantics. Let us simply say that $(\mathcal{T}_1, P)$ and $(\mathcal{T}_2, P)$ are equivalent if they have the same answer sets, and let us say they are *strongly $\mathcal{T}$-equivalent* relative to a vocabulary Σ if, for any Σ-theory $\mathcal{T}$, $(\mathcal{T}_1 \cup \mathcal{T}, P)$ and $(\mathcal{T}_2 \cup \mathcal{T}, P)$ are equivalent. We can relate this to the concept of *strong* query entailment from [9]. The strong *Σ-query difference* between $\mathcal{T}_1$ and $\mathcal{T}_2$, in symbols $\mathrm{sDiff}_\Sigma(\mathcal{T}_1,\mathcal{T}_2)$, is the set of triples $(\mathcal{T},\mathcal{A}, q(x))$ such that $\mathcal{T}$ is a Σ-TBox and $(\mathcal{A}, q(x)) \in \mathrm{Diff}_\Sigma(\mathcal{T}_1,\mathcal{T}_2)$. Then $\mathcal{T}_1$ *strongly* Σ-query entails $\mathcal{T}_2$ if $\mathrm{sDiff}_\Sigma(\mathcal{T}_1,\mathcal{T}_2) = \emptyset$, and $\mathcal{T}_1$ and $\mathcal{T}_2$ are *strongly Σ-equivalent* if each strongly Σ entails the other.

We can make the following observation.

Proposition 7. *Suppose that $\mathcal{T}_1$ and $\mathcal{T}_2$ are strongly Σ-query equivalent TBoxes. Let P be any set of dl-rules all of whose dl-atoms are in Σ. Then the dl-programs $(\mathcal{T}_1, P)$ and $(\mathcal{T}_2, P)$ are strongly $\mathcal{T}$-equivalent relative to Σ.*

An interesting result of [9] is that in some DLs, such as *DL-Lite*$_{bool}$, query and strong query equivalence coincide and are equivalent to the notion of strong concept equivalence (also defined there). In that case we would have the consequence that if $\mathcal{T}_1$ $\mathcal{T}_2$ are query equivalent then the dl-programs $(\mathcal{T}_1, P)$ and $(\mathcal{T}_2, P)$ are strongly $\mathcal{T}$-equivalent. [9] shows how to characterise properties like Σ-query entailment and equivalence, but as expected they are very strong conditions.

Evidently we can now also start to put Propositions 5 and 6 together. Let us say that a dl-rule is proper if it contains a dl-atom, ordinary otherwise. If $(\mathcal{T}, P)$ is a dl-program, let's denote by P^p the set of proper rules of P and P^o the remaining, ordinary rules. Now consider dl-programs that contain the same proper rules but where other components may vary. Furthermore, suppose we are dealing with a DL such as *DL-Lite*$_{bool}$ with the properties established in [9].

Corollary 2. *Let $(\mathcal{T}_1, P_1)$ and $(\mathcal{T}_2, P_2)$ be two dl-programs based on the same set of proper rules.; ie. $P_1^p = P_2^p$. If $\mathcal{T}_1$ and $\mathcal{T}_2$ are Σ-query equivalent and P_1^o is strongly equivalent to P_2^o, then $(\mathcal{T}_1, P_1)$ and $(\mathcal{T}_2, P_2)$ are strongly equivalent with respect to Σ in the sense that for any Σ-theory $\mathcal{T}$ and set of rules P, $(\mathcal{T}_1 \cup \mathcal{T}, P_1 \cup P)$ is answer set equivalent to $(\mathcal{T}_2 \cup \mathcal{T}, P_2 \cup P)$.*

Though rather straightforward, this kind of observation may nevertheless be useful. It says that if we start with two equivalent ontologies and pair them singly together with one or other of two (strongly) equivalent logic programs, then whatever dl-programs we construct on top using a given (single) set of proper dl-rules, we will always obtain equivalent dl-programs.

5 Conclusion

We have provided some initial results characterising concepts of equivalence for hybrid theories. In particular, we first considered hybrid knowledge bases and generalised the concept of strong equivalence for nonmonotonic theories to this setting. Restricting extensions to just one part of the HKBs, either the (classical) theory part or the rule part, the special cases of $\mathcal{T}$-equivalence and $\mathcal{P}$-equivalence are obtained, for which we gave precise model-theoretic characterisations. As a second hybrid formalism, we have considered dl-programs which provide a loose coupling by means of an inference interface and thus lack a unifying semantics. In this setting, the characterisation of strong equivalence is less obvious. However, we have given some preliminary results using the concept of (strong) Σ-query equivalence for DL TBoxes.

Our main objective was to initiate the study of equivalence and modularity for hybrid theories. Our results reveal that characterisations of logical relations between hybrid theories in general are not straightforward compositions of respective characterisations on the components of the theory, even in case of a unifying semantics; the particular interplay of the components has to be taken into account. Consequently, our results are not intended to be exhaustive and, besides more precise characterisations of strong equivalence concepts for dl-programs, many issues remain for further research, including the investigation of more specific and relativised notions of equivalence, the study of computational properties and algorithmic aspects, a detailed analysis of the computational complexity and syntactic fragments, to mention just a few.

References

1. de Bruijn, J., Pearce, D., Polleres, A., Valverde, A. Quantified Equilibrium Logic and Hybrid Rules. In: Marchiori, M. *et al* (eds), RR2007, LNCS vol 4524, pp. 58-72, Springer, Heidelberg (2007)
2. de Bruijn, J., Pearce, D., Polleres, A., Valverde, A. A semantical framework for hybrid knowledge bases. in *Knowledge and Information Systems*, forthcoming.
3. van Dalen, D. *Logic and Structure*. Springer, 1983.
4. Eiter, T., Fink, M., Tompits, H., Woltran, S.: Strong and Uniform Equivalence in Answer-Set Programming: Characterizations and Complexity Results for the Non-Ground Case. *Proc. of AAAI 2005*, pp. 695-700
5. Eiter,T, Lukasiewicz, T, Schindlauer, R, Tompits, H. Combining Answer Set Programming with Description Logics for the Semantic Web. *KR 2004*.
6. Eiter,T, Ianni, G., Lukasiewicz, T, Schindlauer, R, Tompits, H. Combining Answer Set Programming with Description Logics for the Semantic Web. *INFSYS Research Report* 1843-07-04, TU Wien, 2007.
7. Ferraris, P., Lee, J., Lifschitz, V.: A New Perspective on Stable Models. In: Veloso, M. (ed) 20th International Joint Conference on Artificial Intelligence, IJCAI 2007, pp. 372-379.
8. Fink, M. Equivalences in answer set programming by countermodels in the logic of here-and-there. ICLP 2008, LNCS 5366, pp. 99-113.
9. Kontchakov, R, Wolter, F and Zakharyaschev, M. Can you tell the difference between *DL-Lite* ontologies? *Proceedings KR 2008*, 2008.
10. Lutz, C., and Wolter, F. Conservative extensions in the lightweight description logic EL. In *Proc. CADE 07*, 84?99.
11. Lutz, C.; Walther, D.; and Wolter, F. 2007. Conservative extensions in expressive description logics. In *Proc. of IJCAI 07*, 453–458.
12. Heymans, S., Predoiu, L., Feier, C., de Bruijn, J., van Nieuwenborgh, D.: G-hybrid Knowledge Bases. In: Applications of Logic Programming in the Semantic Web and Semantic Web Services, ALPSWS 2006, CEUR Workshop Proceedings, vol 196 (2006)
13. Lifschitz, V., Pearce, D., Valverde, A. Strongly equivalent logic programs. *ACM Transactions on Computational Logic*, 2:526–541, 2001.
14. Lifschitz, V., Pearce, D., Valverde, A. A Characterization of Strong Equivalence for Logic Programs with Variables. In *Proc. of LPNMR 2007*, Springer LNAI 4483, pp. 188-200.
15. Pearce, D. A New Logical Characterisation of Stable Models and Answer Sets. In *Proc. of NMELP 96*, 1997.
16. Pearce, D. Equilibrium logic. *Ann. Math. & and Art. Intelligence*, 47:3-41, 2006.
17. Pearce, D., Valverde, A. Towards a first order equilibrium logic for nonmonotonic reasoning. In *Proc. of JELIA 2004*, Springer LNAI 3229, pp. 147-160.
18. Pearce, D., Valverde, A. A First-Order Nonmonotonic Extension of Constructive Logic. *Studia Logica* 80:321-346, 2005.
19. Pearce, D., Valverde, A. Quantified Equilibrium Logic. *Tech. report, Univ. Rey Juan Carlos*, 2006. $http : //www.matap.uma.es/investigacion/tr/ma06_02.pdf$.
20. Pearce, D., Valverde, A. Quantified Equilibrium Logic and Foundations for Answer Set Programs. *Proc. ICLP 08*, Springer, LNCS, 2008.
21. Rosati, R. On the Decidability and Complexity of Integrating Ontologies and Rules. *Journal of Web Semantics*, 3(1), 2005.
22. Rosati, R. Semantic and Computational Advantages of the Safe Integration of Ontologies and Rules. *PPSWR 2005*.
23. Rosati,R. $\mathcal{DL} + log$: Tight Integration of Description Logics and Disjunctive Datalog. *KR 2006*.