A Logical Semantics for Description Logic Programs*

Michael Fink[1] and David Pearce[2]

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

Abstract. We present a new semantics for Description Logic programs [1] (dl-programs) that combine reasoning about ontologies in description logics with non-monotonic rules interpreted under answer set semantics. Our semantics is equivalent to that of [1], but is more logical in style, being based on the logic $\mathbf{QHT}$ of quantified here-and-there that provides a foundation for ordinary logic programs under answer set semantics and removes the need for program reducts. Here we extend the concept of $\mathbf{QHT}$-model to encompass dl-programs. As an application we characterise some logical relations between dl-programs, by mating the idea of $\mathbf{QHT}$-equivalence with the concept of query inseparability taken from description logics.

1 Introduction

Amalgamating description logics and nonmonotonic logic programs in order to combine rule-based reasoning with ontologies is a growing field of research in knowledge representation and reasoning. Its relevance stems from the aim to build powerful AI systems for Semantic Web reasoning, gradually extending the expressiveness and reasoning capabilities of their underlying formal framework. There have been several different proposals for merging description logics and logic programs into a more tightly or a more loosely integrated semantical framework. Among the best known methods are those based on stable model semantics or answer set programming (ASP); see eg., [1–6]. We shall focus on dl-programs [1] which are given as a pair $\mathcal{D} = (\mathcal{T}, \mathcal{P})$, where $\mathcal{T}$ is a description logic (classical) knowledge base, and $\mathcal{P}$ is a set of so-called dl-rules. Intuitively, the intended models are simply models of $\mathcal{P}$. However the rules of $\mathcal{P}$ may contain special expressions, called dl-atoms, that refer to concepts in $\mathcal{T}$. These atoms are evaluated in a candidate model for $\mathcal{P}$ by posing queries to the classical base $\mathcal{T}$.

As for ordinary ASP, the semantics of dl-programs has been defined by means of program reducts of $\mathcal{P}$. However, it is more involved, since the meaning assigned to concepts appearing in dl-atoms via $\mathcal{T}$ has to be taken into account as well, and the interpretations of the two parts of the program are to some extent distinct. While we can understand the $\mathcal{T}$ component roughly in the sense of classical logic, the answer set semantics does not associate any logic to the $\mathcal{P}$ component and thus to the dl-program as a whole. This is clearly an obstacle to studying intertheory relations and modularity properties that are relevant for applications. It is therefore useful to try to reformulate

* Partially supported by the MCICINN projects TIN2006-15455, TIN2009-14562-CO5, and CSD2007-00022, as well as by the WWTF project ICT08-020.

the answer set semantics of dl-programs in a style that is closer to ordinary logical semantics.

Fortunately there is a suitable logical foundation for ASP. Answer sets can be understood as minimal models in an ordinary, monotonic logic: the logic of *here-and-there*. In first-order form this logic, called the *quantified logic of here-and-there*, in symbols $\mathbf{QHT}$, provides a foundation for non-ground answer set programs [7]. In $\mathbf{QHT}$ one can define a notion of minimal model, called *equilibrium model* [8], that exactly corresponds to answer sets. The logic associated with just these minimal models is known as *equilibrium logic*. An important feature of $\mathbf{QHT}$ is that equivalence in this logic is a necessary and sufficient condition for two programs or nonmonotonic theories to be strongly equivalent, meaning that they are inter-substitutable without loss in all contexts [15]. We may call this the strong equivalence property.

If hybrid theories like dl-programs are to become a successful, practical tool in knowledge-based reasoning, we need to study how ontologies and rules can be combined in a modular fashion. 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. There has already 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 [9–11]. On the other hand, in ASP, work on (strong) equivalence relations between programs began already in [12], 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 from propositional programs to theories and programs in first-order logic [13–15]. This is important for the study of dl-programs where first-order languages are needed.

The aim of this paper is to provide a more logical style of semantics for dl-programs by extending the concepts of $\mathbf{QHT}$-model and equilibrium model to embrace dl-rules. This helps to make the semantics simpler and more uniform. As an illustration of its use we consider ways to define and study strong forms of equivalence between programs that may be useful for combining ontologies and rules in a modular fashion. Briefly:

- We formulate different, logical semantics for dl-programs using $\mathbf{QHT}$-models, removing the need for program reducts.
- We combine the idea of $\mathbf{QHT}$-equivalence with the concept of query inseparability and apply the new semantics to characterise different notions of equivalence between dl-programs.
- Besides strong and weak answer set semantics for dl-programs, we define an alternative semantics which precisely captures the semantics of dl-programs realised as HEX programs, ie., under the so-called *FLP-reduct* [16].

In the next section we introduce necessary background, before characterising dl-program semantics by means of $\mathbf{QHT}$ in Section 3. We study equivalence concepts for dl-programs in Section 4, followed by a discussion of extensions to dl-programs under HEX semantics and conclusions (Sections 5 and 6).

2 Preliminaries

Quantified Equilibrium Logic. In this paper we restrict attention to function-free languages with a single negation symbol, '$\neg$', working with a quantified version of the logic *here-and-there*. In other respects we follow the treatment of [17, 7]. We consider first-order languages $\mathcal{L} = \langle C, P \rangle$ built over a set of *constant* symbols, C, and a set of *predicate* symbols, P. The sets of $\mathcal{L}$-terms, ground $\mathcal{L}$-terms, $\mathcal{L}$-formulas, $\mathcal{L}$-sentences and atomic $\mathcal{L}$-sentences are defined in the usual way. If D is a non-empty set of domain constants, 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 *classical* $\mathcal{L}$-structure can be regarded as a tuple $\mathcal{M} = \langle (D, \sigma), I \rangle$ where I is an $\mathcal{L}$-interpretation over D and $\sigma\colon C \cup D \to D$ is a mapping, called the *assignment*, such that $\sigma(d) = d$ for all $d \in D$. If $D = C$ and $\sigma = id$, $\mathcal{M}$ is an *Herbrand structure*.

A *here-and-there* $\mathcal{L}$-structure with static domains, or $\mathbf{QHT}(\mathcal{L})$-*structure*, is a tuple $\mathcal{M} = \langle (D, \sigma), I_h, I_t \rangle$ where $\langle (D, \sigma), I_h \rangle$ and $\langle (D, \sigma), I_t \rangle$ are classical $\mathcal{L}$-structures such that $I_h \subseteq I_t$. We can think of a here-and-there structure $\mathcal{M}$ as similar to a first-order classical model, 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 [18], where the worlds are ordered by $h \le t$. 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. Since $h \le t$, whatever is verified at h remains true at t. The satisfaction relation for $\mathcal{M}$ is defined so as to reflect the two different components, so we write $\mathcal{M}, w \models \varphi$ to denote that φ is true in $\mathcal{M}$ with respect to the w component. The recursive definition of the satisfaction relation forces us to consider formulas from $\langle C \cup D, P \rangle$. Evidently we should require that an atomic sentence is true at w just in case it belongs to the w-interpretation. Formally, if $p(t_1, \ldots, t_n) \in At(C \cup D, P)$, r and s are $\mathcal{L}$-terms, and $w \in \{h, t\}$ then

$$\mathcal{M}, w \models p(t_1, \ldots, t_n) \quad \text{iff} \quad p(\sigma(t_1), \ldots, \sigma(t_n)) \in I_w.$$
$$\mathcal{M}, w \models r = s \quad \text{iff} \quad \sigma(r) = \sigma(s)$$

The second clause means that our semantics satisfies the axiom of "decidible equality"

$$\forall x \forall y (x = y \vee x \neq y).$$

Then $\models$ is extended recursively using the usual Kripke truth conditions for $\wedge, \vee, \to, \neg, \forall, \exists$ in intuitionistic logic bearing in mind our assumptions about the two worlds h and t and the single domain D, see eg. [18, 7].

Truth of a sentence in a model is defined as follows: $\mathcal{M} \models \varphi$ iff $\mathcal{M}, w \models \varphi$ for each $w \in \{h, t\}$. 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.

The resulting logic is called *Quantified Here-and-There Logic with static domains and decidable equality*, and denoted in [15] by $\mathbf{SQHT}^{=}$, where a complete axiomaisation can be found. To simplify notation we drop the labels for static domains and equality and refer to this logic simply as quantified here-and-there, $\mathbf{QHT}$. Quantified equilibrium logic, or QEL, is based on a suitable notion of minimal model.

Definition 1. *Among* $\mathbf{QHT}$*-structures over a given language we define the order* $\trianglelefteq$ *by:* $\langle(D,\sigma),H,T\rangle \trianglelefteq \langle(D',\sigma'),H',T'\rangle$ *if* $D = D'$, $\sigma = \sigma'$, $T = T'$ *and* $H \subseteq H'$*. If the subset relation is strict, we write '*$\triangleleft$*'. Let* Γ *be a set of sentences and* $\mathcal{M} = \langle(D,\sigma),H,T\rangle$ *a model of* Γ*.* $\mathcal{M}$ *is said to be an* equilibrium *model of* Γ *if it is minimal under* $\trianglelefteq$ *among models of* Γ*, and it is total.*

Answer sets. We assume the reader is familiar with the usual definitions of answer set based on *Herbrand* models and ground programs, eg. [19]. Two variations of this semantics, the open [20] and generalised open answer set [5] semantics, consider non-ground programs and open domains, thereby relaxing the standard name assumption. In addition, [21] offers a very general concept of stable model for arbitrary first-order formulas, defining the property of being a stable model syntactically via a second-order condition.

The correspondence between QEL and answer set semantics is by now quite well known and has been described in several works (see [22, 17, 7, 21]). By the usual convention, when $\mathcal{P}$ is a logic program with variables we consider the models of its universal closure expressed as a set of logical formulas. It follows that if $\mathcal{P}$ is a logic program (of any form), a total $\mathbf{QHT}$ model $\langle U,T,T\rangle$ of $\mathcal{P}$ is an equilibrium model of $\mathcal{P}$ iff it is a stable model of $\mathcal{P}$ in the sense of [21]. Moreover, two logic programs $\mathcal{P}_1$ and $\mathcal{P}_2$ are *strongly equivalent* iff they coincide on their $\mathbf{QHT}$-models. Placing additional restrictions on $\mathbf{QHT}$ models, we obtain a correspondence to other notions of answer set such as those based on a standard name assumption.

DL-programs. Turning to dl-programs [1, 23], we start without restricting the syntax of the classical part or the knowledge base that is combined with logic program rules; later on we shall consider some concepts and properties that apply to dl-programs based on (particular) description logics (for a background and corresponding notation used cf. [24]). In other words, we consider arbitrary function-free first-order theories that are combined with dl-rules and, for the moment, we allow for arbitrary formulas as queries in dl-atoms. Moreover, disjunction is allowed in rule heads, while we require that the classical theory and the logic program share a single set of constants.

More formally, 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

$$DL[S_1 op_1 p_1, \ldots, S_m op_m p_m; Q](t_1, \ldots, t_n), \tag{1}$$

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

$$b_1 \wedge \ldots \wedge b_m \wedge \neg b_{m+1} \wedge \ldots \wedge \neg b_n \rightarrow h_1 \vee \ldots \vee h_l \tag{2}$$

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. The expression $h_1 \vee \ldots \vee h_l$ is abbreviated by $Hd(r)$. 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.

Example 1. Consider the following vocabulary dealing with wine: constants frb and ldm are used for '*Freixenet Brut*' and '*Lambrusco di Modena*', respectively; the classical predicates $W(x)$, $R(x)$, $S(x)$, and $L(x)$, represent the concepts of *White Wine*, *Red Wine*, *Sparkling Wine*, and *Lambrusco*; $w(x)$, $r(x)$, $s(x)$, and $l(x)$ are rule predicates intended to reason about the above concepts in rules; an additional rule predicate $sc(x)$ encodes whether a wine is *served cold*.

Now, let $(\mathcal{T}, \mathcal{P})$ be the following dl-program over this vocabulary.

$\mathcal{T}$	$\mathcal{P}$	
$L \sqsubseteq R \sqcap S$	$l(ldm)$	$sc(X) \vee \neg sc(X)$
$\neg W \sqcap \neg R \sqsubseteq \bot$	$s(frb)$	$l(X) \rightarrow \neg sc(X)$
$R \sqcap W \sqsubseteq \bot$	$DL[S \uplus s, L \uplus l; R](X) \rightarrow r(X)$	$\neg r(X) \rightarrow w(X)$

Intuitivey, the dl-rule says: in $\mathcal{T}$ add to S the contents of s and add to L the contents of l; if $R(X)$ now follows (in the enlarged $\mathcal{T}$), then $r(X)$. $\Box$

Turning to the formal 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, and let $\models_c$ denote classical entailment.

An 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 (1) under $\mathcal{T}$ if $\mathcal{T} \cup \bigcup_{i=1}^{m} A_i(I) \models_c Q(t_1, \ldots, t_n)$, 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 $b_i \in \{b_{m+1}, \ldots, b_n\}$ 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 $\mathcal{D}$.

3 Logical Semantics

We reformulate the semantics for dl-programs in a style that is closer to ordinary logical semantics and in particular to the logic $\mathbf{QHT}$. This makes it easier to characterise logical properties of dl-programs and relations between them.

Dl-atoms and rules are defined as above in (1), (2). We use the usual semantics for $\mathbf{QHT}$, so the truth conditions for ordinary atoms, conjunctions, disjunctions, negation and implications in a model $\mathcal{M} = \langle U, H, T\rangle$ are the same as before. For dl-atoms we define three semantics, the last two of which correspond to weak and strong answer sets respectively. Informally these semantics work as follows. The truth of a dl-atom (1) is checked as before by inspecting whether the query Q follows classically from a certain extension of the theory $\mathcal{T}$. The difference is that, as a base model for computing the A_i, as well as for defining the truth of a dl-atom, we now use a $\mathbf{QHT}$ model instead of a classical Herbrand model. This allows a more uniform treatment of the different operators. We begin with a semantics that corresponds to a variation of strong answer sets.

Definition 2 (models of dl-atoms). *Let α be a ground dl-atom of the form (1) and let $\mathcal{M} = \langle U, H, T\rangle$ be a $\mathbf{QHT}$ structure. Then, $\mathcal{M}$ is a* model *of α under $\mathcal{T}$ iff $\mathcal{M}, w \models \alpha$ for $w = h, t$; where $\mathcal{M}, w \models \alpha$ iff $\mathcal{T} \cup \bigcup_{i=1}^{m} A_i(w) \models_c Q(t_1, \ldots, t_n)$, where*

- *$A_i(w) = \{S_i(\boldsymbol{e}) \mid \mathcal{M}, w \models p_i(\boldsymbol{e})\}$, for $op_i = \uplus$,*
- *$A_i(w) = \{\neg S_i(\boldsymbol{e}) \mid \mathcal{M}, w \models p_i(\boldsymbol{e})\}$, for $op_i = \cupdot$,*
- *$A_i(w) = \{\neg S_i(\boldsymbol{e}) \mid \mathcal{M}, w \models \neg p_i(\boldsymbol{e})\}$, for $op_i = \Cap$,*

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

Definition 3 (weak models of dl-atoms). *Let α be a ground dl-atom of the form (1) and let $\mathcal{M} = \langle U, H, T\rangle$ be a $\mathbf{QHT}$ structure. Then we say that $\mathcal{M}$ is a* weak model *of α under $\mathcal{T}$ iff $\mathcal{M}, w \models \alpha$ for $w = h, t$; where $\mathcal{M}, t \models \alpha$ is defined as in the semantics of Definition 2 and $\mathcal{M}, h \models \alpha \Leftrightarrow \mathcal{M}, t \models \alpha$.*

Observe that now (U, T) need not be an Herbrand model. Notice that in the first semantics operators are evaluated at both worlds h and t in the model, while in the second, weak semantics they are essentially evaluated only at t which then determines the value at h.

Finally we introduce a variant of the first semantics that corresponds to strong answer sets. For this we need to distinguish between atoms known to be monotonic and others. As before we use the symbols $DL^{+}(\mathcal{P})$ and $DL^{?}(\mathcal{P})$ for these. Let us adopt the convention that all atoms containing an occurrence of the operator $op_i = \Cap$ belong to $DL^{?}(\mathcal{P})$, while all others are in $DL^{+}(\mathcal{P})$.

Definition 4 (strong models of dl-atoms). *Let α be a ground dl-atom of the form (1) and let $\mathcal{M} = \langle U, H, T\rangle$ be a $\mathbf{QHT}$ structure. Then we say that $\mathcal{M}$ is a* strong model *of α under $\mathcal{T}$ iff $\mathcal{M}, w \models \alpha$ for $w = h, t$; where for all atoms α, $\mathcal{M}, t \models \alpha$ is defined as in the semantics of Definition 2, while $\mathcal{M}, h \models \alpha$ is defined as in the semantics of Definition 2 if $\alpha \in DL^{+}(\mathcal{P})$, and as in Definition 3, ie. by $\mathcal{M}, h \models \alpha \Leftrightarrow \mathcal{M}, t \models \alpha$, otherwise.*

A dl-rule r is true in a model $\mathcal{M}$ under $\mathcal{T}$, in symbols $\mathcal{M} \models_{\mathcal{T}} r$, if the rule is satisfied according to the usual **QHT** semantics. We may suppress the subscript $\mathcal{T}$ if the context is clear. The following property is important but easy to verify.

Proposition 1 (persistence). *For any model $\mathcal{M}$ and rule r, $\mathcal{M}, h \models r \Rightarrow \mathcal{M}, t \models r$, for each of the semantics.*

The notions of model (resp. weak and strong model) and equilibrium model (resp. weak, strong equilibrium model) are now defined in the obvious way.

Definition 5. *A* **QHT** *structure $\mathcal{M} = \langle U, H, T\rangle$ is a* model *(resp.* weak model, strong model*) of a dl-program $\mathcal{D} = (\mathcal{T}, \mathcal{P})$ if $\mathcal{M} \models_{\mathcal{T}} r$ for each $r \in \mathcal{P}$ under the semantics (resp. the weak, the strong semantics) for dl-atoms. It is said to be an* equilibrium model *(resp.* weak, strong equilibrium model*) of $\mathcal{D}$ if $H = T$ and $\mathcal{M}$ is a minimal model (resp. weak, strong model) of $\mathcal{P}$ under $\mathcal{T}$ wrt $\trianglelefteq$, ie. there is no model of $\mathcal{D}$ (resp. weak model, strong model of $\mathcal{D}$) of the form $\langle U, H', T\rangle$ where H' is a proper subset of H.*

For reasons of space we do not give a detailed proof of our main theorem, Proposition 2 below, which established the correctness of our semantics. However the proof is based on the following two lemmas which are fairly routine. We formulate for the case of strong models; similar properties hold for weak models.

Lemma 1. *Let $\mathcal{M} = \langle U, H, T\rangle$ be a* **QHT** *strong Herbrand model of $\mathcal{P}$ under $\mathcal{T}$. Then $\langle H, T\rangle \models s\mathcal{P}^{\mathcal{M}}_{\mathcal{T}}$.*

Lemma 2. *Let $\mathcal{M} = \langle U, H, T\rangle$ be a* **QHT** *strong Herbrand model of $\mathcal{P}$ under $\mathcal{T}$. Then $\mathcal{M}$ is a minimal model of $\mathcal{P}$ under $\mathcal{T}$ wrt $\trianglelefteq$ if and only if $\langle U, H\rangle$ is a minimal model of $(\mathcal{T}, s\mathcal{P}^{\mathcal{M}}_{\mathcal{T}})$.*

From these properties we can derive:

Proposition 2. *A total Herbrand* **QHT** *structure $\mathcal{M} = \langle U, T, T\rangle$ is a weak (resp. strong) equilibrium model of a dl-program $\mathcal{D} = (\mathcal{T}, \mathcal{P})$ iff $\langle U, T\rangle$ is a weak (resp. strong) answer set of $\mathcal{D}$.*

Example 2. Reconsider $\mathcal{D} = (\mathcal{T}, \mathcal{P})$ from Example 1 with universe $U = (\{frb, ldm\}, id)$. The structures $\mathcal{M} = \langle U, T, T\rangle$ and $\mathcal{M} = \langle U, T', T'\rangle$, where $T = \{s(frb), l(ldm), w(frb), r(ldm)\}$ and $T' = T \cup \{sc(frb)\}$, are weak and strong equlibrium models of $\mathcal{D}$ (note that the only dl-atom is monotone, and that, for every $\{l(ldm)\} \subseteq H \subset T'$, the dl-atom is true for ldm). They are also weak, as well as strong, answer sets of $\mathcal{D}$. □

Although the alternative semantics is therefore equivalent to the original one, there are several features worth emphasising. First, since we have removed the need for reducts, we can extend the semantics to more general types of rules and formulas just using the usual truth conditions for **QHT** models.[3] Secondly, although we shall consider here just the usual dl-programs with Herbrand models, our semantics is not limited to this and we could in principle consider non-Herbrand interpretations, as in the case

[3] In principle we could extend the syntax of rules r to any formula providing that Proposition 1 continues to hold.

of hybrid knowledge bases. Thirdly, we now have a more homogeneous and logical semantics that may help us derive logical properties of dl-programs.

Finally, an advantage of the first semantics is that, by using $\mathbf{QHT}$ structures, we do not have to distinguish semantically between monotone and possibly non-monotone operators. All operators are treated similarly. The difference between models and weak models is merely that the former evaluate dl-atoms by looking only at the t-world. Notice that although we apply the words 'weak' and 'strong' to models, these labels are really used to reflect the difference between weak and strong equilibrium models or answer sets. For example, while every strong equilibrium model is also a weak one, not every strong model (or ordinary model) need be a weak one. Observe that if all dl-atoms containing $op_i = \Cap$ are 'pure', in the sense that they do not contain occurrences of $\uplus$ or $\cup\!\!\!\!-$, then the first semantics and the strong semantics coincide.

4 Equivalence Concepts

To illustrate the use of the new semantics, we introduce and study some concepts of equivalence between dl-programs. We can consider different equivalence relations between dl-programs according to how the different components, $\mathcal{T}$ and $\mathcal{P}$, are allowed to vary. If $\mathcal{D} = (\mathcal{T}, \mathcal{P})$ is a dl-program, $\mathcal{T}$' is a classical theory and $\mathcal{P}$' is a set of dl-rules, then $\mathcal{D} \cup \mathcal{T}'$ stands for the program $(\mathcal{T} \cup \mathcal{T}', \mathcal{P})$ and $\mathcal{D} \cup \mathcal{P}'$ stands for the program $(\mathcal{T}, \mathcal{P} \cup \mathcal{P}')$.

Definition 6 (Equivalence for dl-programs). *Two dl-programs $\mathcal{D}_1$ and $\mathcal{D}_2$ are said to be* equivalent *if they have the same equilibrium models, they are $\mathcal{T}$-*equivalent *if $\mathcal{D}_1 \cup \mathcal{T}$ and $\mathcal{D}_2 \cup \mathcal{T}$ are equivalent for any $\mathcal{T}$, they are $\mathcal{P}$-*equivalent *if $\mathcal{D}_1 \cup \mathcal{P}$ and $\mathcal{D}_2 \cup \mathcal{P}$ are equivalent for any $\mathcal{P}$, and they are* strongly equivalent *if $\mathcal{D}_1 \cup \mathcal{T} \cup \mathcal{P}$ and $\mathcal{D}_2 \cup \mathcal{T} \cup \mathcal{P}$ are equivalent for any $\mathcal{T}$ and $\mathcal{P}$.*

Having the same equilibrium models is to be understood under any of the given semantics. However, unless our results are specific to one semantics, we don't further specify which one. We also say that $\mathcal{D}_1$ and $\mathcal{D}_2$ are $\mathbf{QHT}$-equivalent if they have the same $\mathbf{QHT}$ models (in any of the given senses). Lastly, it is useful to introduce relativised versions of these concepts. Thus, if Σ is a signature or vocabulary and $\mathcal{P}$ is a set of dl-rules, we say that $\mathcal{P}$ is a set of Σ-dl-rules if all classical predicates appearing in any dl-atom are from Σ.

Definition 7 (Σ-equivalence for dl-programs). *Given a signature Σ, two dl-programs $\mathcal{D}_1$ and $\mathcal{D}_2$ are said to be Σ-$\mathcal{T}$-*equivalent *if $\mathcal{D}_1 \cup \mathcal{T}$ and $\mathcal{D}_2 \cup \mathcal{T}$ are equivalent for any theory $\mathcal{T}$ in Σ, they are Σ-$\mathcal{P}$-*equivalent *if $\mathcal{D}_1 \cup \mathcal{P}$ and $\mathcal{D}_2 \cup \mathcal{P}$ are equivalent for any set of Σ-dl-rules $\mathcal{P}$, and they are* strongly Σ-equivalent *if $\mathcal{D}_1 \cup \mathcal{T} \cup \mathcal{P}$ and $\mathcal{D}_2 \cup \mathcal{T} \cup \mathcal{P}$ are equivalent for any $\mathcal{T}$ and $\mathcal{P}$, such that $\mathcal{T}$ in Σ and $\mathcal{P}$ is a set of Σ-dl-rules.*

A first, simple observation is that if two ordinary answer set programs are strongly equivalent they cannot be separated by additional dl-rules.

Proposition 3. *Let Π_1, Π_2 be two strongly equivalent logic programs. Let $\mathcal{R}$ be any set of dl-rules and let $(\mathcal{T}, \mathcal{P}_1), (\mathcal{T}, \mathcal{P}_2)$ be dl-programs where $\mathcal{P}_1 = \Pi_1 \cup \{\mathcal{R}\}$ and $\mathcal{P}_2 = \Pi_2 \cup \{\mathcal{R}\}$. Then $(\mathcal{T}, \mathcal{P}_1)$ and $(\mathcal{T}, \mathcal{P}_2)$ are equivalent under all the given semantics for dl-programs.*

This simple observation can be generalised. Notice that we keep $\mathcal{T}$ fixed in each case since otherwise a given rule $r \in \mathcal{R}$ could have a completely different interpretation in one of the extended dl-programs than it does in the other.

Proposition 4. *Two dl-programs, $(\mathcal{T}, \mathcal{P}_1)$ and $(\mathcal{T}, \mathcal{P}_2)$, are $\mathcal{P}$-equivalent (under a given semantics) if and only if they are $\mathbf{QHT}$-equivalent (under the same semantics).*

Proof (Sketch). For the 'if' direction the argument is the same as for Proposition 3: if $(\mathcal{T}, \mathcal{P}_1)$ and $(\mathcal{T}, \mathcal{P}_2)$ have the same $\mathbf{QHT}$ models, then, whatever set of dl-rules $\mathcal{R}$ that is added to them will yield the same set of $\mathbf{QHT}$ models in each case, and hence the same equilibrium models. For the 'only if' direction we can use the proofs of strong equivalence theorems found in [15]. The only additional property we need to check for the case of dl-rules is that if $\mathcal{M} = \langle U, H, T \rangle$ is a $\mathbf{QHT}$ model of a program $\mathcal{P}$ under $\mathcal{T}$, then $\mathcal{M} = \langle U, T, T \rangle$ is also a $\mathbf{QHT}$ model of $\mathcal{P}$ under $\mathcal{T}$. But this is guaranteed by the persistence property stated in Proposition 1. □

We now turn to the case of a varying knowledge base. To deal with the situation where $\mathcal{T}$ is allowed to vary, we 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 (see eg. the following references). In the papers [9–11] 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 a definition from [11, 25]. To simplify notation we assume that some DL is given, while Σ is a vocabulary or signature.[4] Let $\mathcal{T}_1$ and $\mathcal{T}_2$ be TBoxes.

Definition 8. *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 $(\mathcal{T}_1, \mathcal{A}) \not\models_c Q(\mathbf{a})$ and $(\mathcal{T}_2, \mathcal{A}) \models_c Q(\mathbf{a})$, for some tuple $\mathbf{a}$ of object names from $\mathcal{A}$. We say that $\mathcal{T}_1$ Σ-query entails $\mathcal{T}_2$ if $\mathrm{Diff}_\Sigma(\mathcal{T}_1, \mathcal{T}_2) = \emptyset$. Furthermore we say that $\mathcal{T}_1$ and $\mathcal{T}_2$ are* Σ-query inseparable *if each Σ-query entails the other.*

In other words, query inseparability means equivalence 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 $(\mathcal{T}, \mathcal{P})$ for some TBox, $\mathcal{T}$. Now the way in which a ground dl-atom is evaluated in an Herbrand interpretation $\mathcal{M}$ 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. From now on we make the assumption 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.[5]

Proposition 5. *Suppose that $\mathcal{T}_1$ and $\mathcal{T}_2$ are Σ-query inseparable TBoxes, and let P be any set of Σ-dl-rules. Then the dl-programs $(\mathcal{T}_1, P)$ and $(\mathcal{T}_2, P)$ are equivalent.*

[4] For [11, 25] the signature does not include constant symbols.

[5] In general the concept of query inseparability depends not only on the vocabulary Σ but also on the given query language or syntax; different ones have been considered in the literature. To save space we leave this variable implicit and merely suppose that the query language operating over $\mathcal{T}$ in the DL is the same one that is used for evaluating dl-atoms in the dl-program.

In order to explore the notion of Σ-$\mathcal{T}$-equivalence we can make use of the concept of *strong* query entailment from [11]. 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 $(\mathcal{A}, q(x)) \in \mathrm{Diff}_\Sigma(\mathcal{T}_1 \cup \mathcal{T}, \mathcal{T}_2 \cup \mathcal{T})$. 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-query *inseparable* if each strongly Σ-query entails the other. Moreover, we say that $\mathcal{T}_1$ and $\mathcal{T}_2$ are *strongly query inseparable* if they are Σ-query inseparable for any Σ. This leads to the following observation by an obvious extension to the argument for Proposition 5.

Proposition 6. *Suppose that $\mathcal{T}_1$ and $\mathcal{T}_2$ are strongly Σ-query inseparable TBoxes, and let P be any set of Σ-dl-rules. Then $(\mathcal{T}_1, P)$ and $(\mathcal{T}_2, P)$ are Σ-$\mathcal{T}$-equivalent.*

An interesting result of [11] is that in some DLs, such as *DL-Lite*$_{bool}$, query and strong query inseparability coincide and are equivalent to the notion of strong concept inseparability (also defined there). In that case we would have the consequence that if $\mathcal{T}_1$ $\mathcal{T}_2$ are Σ-query inseparable then the dl-programs $(\mathcal{T}_1, \mathcal{P})$ and $(\mathcal{T}_2, \mathcal{P})$ are Σ-$\mathcal{T}$-equivalent.

By combining ideas from Propositions 4 and 5 we can obtain some sufficient conditions for Σ-$\mathcal{P}$-equivalence under varying TBoxes.

Proposition 7. *Let $\mathcal{D}_1 = (\mathcal{T}_1, \mathcal{P}_1)$ and $\mathcal{D}_2 = (\mathcal{T}_2, \mathcal{P}_2)$ be $\mathbf{QHT}$-equivalent dl-programs where $\mathcal{T}_1$ and $\mathcal{T}_2$ are Σ-query inseparable TBoxes. Then $\mathcal{D}_1$ and $\mathcal{D}_2$ are Σ-$\mathcal{P}$-equivalent.*

Analogous to Proposition 6, we obtain a 'strong' version of Proposition 7 by replacing Σ-query inseparability by strong Σ-query inseparability.

Proposition 8. *Let $\mathcal{D}_1 = (\mathcal{T}_1, \mathcal{P}_1)$ and $\mathcal{D}_2 = (\mathcal{T}_2, \mathcal{P}_2)$ be $\mathbf{QHT}$-equivalent dl-programs where $\mathcal{T}_1$ and $\mathcal{T}_2$ are strongly Σ-query inseparable TBoxes. Then the dl-programs $\mathcal{D}_1$ and $\mathcal{D}_2$ are strongly Σ-equivalent.*

The following corollary that drops reference to Σ is straightforward. It also generalises Proposition 4.

Corollary 1. *Suppose $\mathcal{T}_1$ and $\mathcal{T}_2$ are strongly query inseparable. Then $(\mathcal{T}_1, \mathcal{P}_1)$ and $(\mathcal{T}_2, \mathcal{P}_2)$ are strongly equivalent iff they are $\mathbf{QHT}$-equivalent.*

Further generalisations may be possible by applying the concept of relativised program equivalence, but we leave this for future work.

To illustrate the above concepts, let us consider a simple example.

Example 3. Using the vocabulary of Example 1, let $\mathcal{D}_1 = (\mathcal{T}_1, \mathcal{P}_1)$ and $\mathcal{D}_2 = (\mathcal{T}_2, \mathcal{P}_2)$ be dl-programs given by:

$\mathcal{T}_1$	$\mathcal{T}_2$	$\mathcal{P}'$
$L \sqsubseteq R \sqcap S$	$L \sqsubseteq R \sqcap S$	$l(ldm)$
$\neg W \sqcap \neg R \sqsubseteq \bot$	$\neg W \sqcap \neg R \sqsubseteq \bot$	$s(frb)$
$R \sqcap W \sqsubseteq \bot$	$R \sqsubseteq \neg W$	$sc(X) \vee \neg sc(X)$
	$W \sqsubseteq \neg R$	$l(x) \rightarrow \neg sc(x)$

and $\mathcal{P}_1 = \mathcal{P}' \cup \{r_{1_1}, r_{2_1}, r_3\}$, $\mathcal{P}_2 = \mathcal{P}' \cup \{r_{1_2}, r_{2_2}, r_3\}$, where $r_{1_1} = w(X) \rightarrow sc(X)$, $r_{1_2} = w(X) \wedge \neg sc(X) \rightarrow \bot$, $r_3 = DL[S \uplus s, L \uplus l; S](X) \wedge \neg r(X) \rightarrow w(X)$, $r_{2_1} = DL[S \uplus s, L \uplus l; R](X) \rightarrow r(X)$, $r_{2_2} = DL[S \uplus s, L \uplus l; \neg W](X) \rightarrow r(X)$.

Suppose that $\mathcal{T}_1$ and $\mathcal{T}_2$ are TBoxes in *DL-Lite*$_{bool}$. If Σ is the classical language as given in Example 1, then $\mathcal{T}_1$ and $\mathcal{T}_2$ are Σ-query equivalent and therefore strongly Σ-query equivalent by the results of [11]. Moreover, $\mathcal{D}_1$ and $\mathcal{D}_2$ are $\mathbf{QHT}$-equivalent. To see the latter, first observe that the ordinary rules in each of the programs are strongly equivalent. Secondly, the dl-atom $DL[S \uplus s, L \uplus l; R](X)$ has the same models under $\mathcal{T}_1$ as the dl-atom $DL[S \uplus s, L \uplus l; \neg W](X)$ has under $\mathcal{T}_2$, because in both theories the concepts R and $\neg W$ are equivalent. Therefore, $\mathcal{D}_1$ and $\mathcal{D}_2$ are strongly equivalent. □

5 HEX Programs

Another type of hybrid theory, called HEX program, was introduced in [6]. This combines answer set programs with higher-order atoms and external atoms. In particular, the external atoms can refer, as in dl-programs, to concepts belonging to a classical knowledge base or ontology. In such a case one can compare the semantics of the HEX program with that of the corresponding dl-program. Although both are based on answer sets, the two semantics are only partially in agreement. Specifically, as shown in [6], they agree on programs all of whose external atoms (dl-atoms) contain only monotone operators. Then, the answer sets of the HEX program coincide with the strong answer sets of the dl-program.

The study of equivalence concepts for HEX programs in general is beyond the scope of this work. However, we can easily deal with the case where such programs contain external atoms having precisely the form of dl-atoms (monotonic or otherwise). For in this case the HEX semantics is in agreement with our first, alternative semantics for dl-programs, given in Definition 2. Without giving a detailed account of HEX programs, we indicate briefly why this is so.

Formally, external atoms in HEX programs have their own special notation and semantics. However, since dl-atoms can easily be simulated in HEX programs, for the purposes of our comparison let us keep the usual notation as for dl-programs. In that case, a HEX program is just a disjunctive logic program $\mathcal{P}$ containing rules of form (2) whose bodies can contain dl-atoms of form (1). The interpretation of such rules is similar to that of dl-programs except that a different form of program reduct is used. In [6] this is called *FLP-reduct* following the first use of this notion in [16].

Assume that we are given such a HEX program $\mathcal{P}$ along with some knowledge base $\mathcal{T}$ with respect to which the external atoms are evaluated (in what follows we shall leave the $\mathcal{T}$ component as implicit). Then the truth of an external atom of form (1) in a classical Herbrand model $\mathcal{M}$ is defined as for dl-programs in Section 4 above. Ground rules are also satisfied in $\mathcal{M}$ in the same way. Given $\mathcal{P}$ and a classical Herbrand model $\mathcal{M} = \langle U, T \rangle$, the reduct of $\mathcal{P}$ wrt. $\mathcal{M}$, denoted by $\mathcal{P}^{\mathcal{M}}$, is the set of all $r \in gr_U(\mathcal{P})$ such that $\mathcal{M} \models B(r)$. Then $\mathcal{M}$ is said to be an *answer set* of $\mathcal{P}$ iff it is a minimal model of $\mathcal{P}^{\mathcal{M}}$.

Proposition 9. *Let $\mathcal{P}$ be a HEX program as above with external atoms in the form of dl-atoms. A $\mathbf{QHT}$ Herbrand structure $\langle U, T, T \rangle$ is an equilibrium model of $\mathcal{P}$ under the semantics of Definition 2 if and only if $\langle U, T \rangle$ is an answer set of $\mathcal{P}$.*

All our observations and results about equivalences of dl-programs hold for any of the three semantics given. By Proposition 9 they carry over to HEX programs with external access to a TBox $\mathcal{T}$.

6 Conclusion

The logic $\mathbf{QHT}$ of quantified here-and-there provides a foundation for the answer set semantics of logic programs, and sharing the same $\mathbf{QHT}$-models is a necessary and sufficient condition for two programs or theories to be strongly equivalent. In this paper we have shown how the concept of $\mathbf{QHT}$-model can be extended to embrace also dl-programs interpreted under answer set semantics, removing the need for reducts and allowing a more logical style of semantics. Slight variations in the concept of $\mathbf{QHT}$-model give rise to the weak and the strong answer set semantics as well as to a variation based on HEX programs.

As an application of the new semantics we considered some strong forms of equivalence between dl-programs as a first step towards the modular combination of ontologies and rules. Since a dl-program is a pair $(\mathcal{T}, \mathcal{P})$, strong forms of equivalence are obtained by considering theory extensions, which can be relativised to either the $\mathcal{T}$ component or the $\mathcal{P}$ component.

As in ordinary answer set programmming, the property that two theories have the same $\mathbf{QHT}$ models is again significant. This property is a necessary and sufficient condition for the $\mathcal{P}$-equivalence of dl-programs, if they are based on the same classical theory $\mathcal{T}$ or on possibly different but query inseparable TBoxes.

For the other main kind of equivalence, where the $\mathcal{T}$ component may vary, the situation is as follows. For dl-programs based on TBoxes we can use the idea of Σ-query inseparability to characterise forms of $\mathcal{T}$-equivalence for dl-programs based on the same $\mathcal{P}$ component or on $\mathbf{QHT}$-equivalent programs. The concept of Σ-query inseparability has been studied for description logics such as DL-Lite and $\mathcal{EL}$ and model-theoretic characterisations are available and in some cases implemented [11, 26, 9, 25].

One direction of future work is to study modularity issues and equivalence concepts and their properties for dl-programs based on specific DLs such as $\mathcal{EL}$. Such properties may include algorithmic aspects and an analysis of computational complexity. Another direction of work is the study of more specific and relativised notions of equivalence between programs.

References

1. Eiter, T., Lukasiewicz, T., Schindlauer, R., Tompits, H.: Combining answer set programming with description logics for the semantic web. In KR, AAAI Press (2004) 141–151
2. Rosati, R.: Dl+log: Tight integration of description logics and disjunctive datalog. In KR, AAAI Press (2006) 68–78
3. Rosati, R.: On the decidability and complexity of integrating ontologies and rules. J. Web Sem. **3**(1) (2005) 61–73
4. Rosati, R.: Semantic and computational advantages of the safe integration of ontologies and rules. In PPSWR. Volume 3703 of LNCS, Springer (2005) 50–64

5. Heymans, S., de Bruijn, J., Predoiu, L., Feier, C., Nieuwenborgh, D.V.: Guarded hybrid knowledge bases. TPLP **8**(3) (2008) 411–429
6. Eiter, T., Ianni, G., Schindlauer, R., Tompits, H.: A uniform integration of higher-order reasoning and external evaluations in answer-set programming. In IJCAI, (2005) 90–96
7. Pearce, D., Valverde, A.: Quantified equilibrium logic and foundations for answer set programs. In ICLP. Volume 5366 of LNCS, Springer (2008) 546–560
8. Pearce, D.: Equilibrium logic. AMAI **47**(1-2) (2006) 3–41
9. Lutz, C., Wolter, F.: Conservative extensions in the lightweight description logic $\mathcal{EL}$. In CADE. Volume 4603 of LNCS, Springer (2007) 84–99
10. Lutz, C., Walther, D., Wolter, F.: Conservative extensions in expressive description logics. In IJCAI, (2007) 453–458
11. Kontchakov, R., Wolter, F., Zakharyaschev, M.: Can you tell the difference between dl-lite ontologies? In KR, AAAI Press (2008) 285–295
12. Lifschitz, V., Pearce, D., Valverde, A.: Strongly equivalent logic programs. ACM TOCL **2**(4) (2001) 526–541
13. Fink, M.: A general framework for equivalences in answer-set programming by countermodels in the logic of here-and-there. TPLP (2010) forthcoming.
14. Fink, M.: Equivalences in answer-set programming by countermodels in the logic of here-and-there. In ICLP. Volume 5366 of LNCS, Springer (2008) 99–113
15. Lifschitz, V., Pearce, D., Valverde, A.: A characterization of strong equivalence for logic programs with variables. In LPNMR. Volume 4483 of LNCS, Springer (2007) 188–200
16. Faber, W., Leone, N., Pfeifer, G.: Recursive aggregates in disjunctive logic programs: Semantics and complexity. In JELIA. Volume 3229 of LNCS Springer (2004) 200–212
17. Pearce, D., Valverde, A.: Quantified equilibrium logic an the first order logic of here-and-there. Technical Report MA-06-02, Univ. Rey Juan Carlos (2006)
18. van Dalen, D.: Logic and Structure, 4th Ed., Springer, 2004.
19. Baral, C.: Knowledge Representation, Reasoning, and Declarative Problem Solving. Cambridge University Press, (2003)
20. Heymans, S., Nieuwenborgh, D.V., Vermeir, D.: Open answer set programming with guarded programs. ACM TOCL **9**(4) (2008)
21. Ferraris, P., Lee, J., Lifschitz, V.: A new perspective on stable models. In IJCAI, (2007) 372–379
22. Pearce, D., Valverde, A.: A first order nonmonotonic extension of constructive logic. Studia Logica **80**(2-3) (2005) 321–346
23. Eiter, T., Ianni, G., Lukasiewicz, T., Schindlauer, R., Tompits, H.: Combining answer set programming with description logics for the Semantic Web. AIJ **172**(12-13) (2008) 1495–1539
24. Baader, F., Calvanese, D., McGuinness, D., Nardi, D., Patel-Schneider, P., eds.: The Description Logic Handbook: Theory, Implementation and Applications. Cambridge University Press, (2003)
25. Lutz, C., Wolter, F.: Deciding inseparability and conservative extensions in the description logic el. J. Symb. Comput. **45**(2) (2010) 194–228
26. Kontchakov, R., Wolter, F., Zakharyaschev, M.: A logic-based framework for ontology comparison and module extraction in dl-lite. (2010) submitted.
27. Fink, M., Pearce, D.: Some equivalence concepts for hybrid theories. In CAEPIA, Spanish Association for Artifical Intelligence (2009) 327–336