# Integrating Rules and Ontologies in the First-Order Stable Model Semantics (Preliminary Report)

**Joohyung Lee** and **Ravi Palla**
School of Computing, Informatics and Decision Systems Engineering
Arizona State University
Tempe, AZ, 85287, USA
{joolee, Ravi.Palla}@asu.edu

## Abstract

We present an approach to integrating rules and ontologies on the basis of the first-order stable model semantics proposed by Ferraris, Lee and Lifschitz. We show that some existing integration proposals can be uniformly reformulated in terms of the first-order stable model semantics. The reformulations are simpler than the original proposals in the sense that they do not refer to grounding.

## Introduction

Integrating nonmonotonic rules and ontologies has received much attention, especially in the context of the Semantic Web. A hybrid knowledge base is a pair $(\mathcal{T}, \mathcal{P})$ where $\mathcal{T}$ is a FOL knowledge base (typically in a description logic) of signature $\Sigma_{\mathcal{T}}$ and $\mathcal{P}$ is a logic program of signature $\Sigma_{\mathcal{P}}$. The existing integration approaches can be classified into three categories: *loose integration*, *tight integration with semantic separation* and *tight integration under a unifying logic* (Nazarenko *et al.* 2010). In the *loose integration* approach, $\mathcal{T}$ and $\mathcal{P}$ are viewed as separate, independent components, and they are connected through minimal safe interfaces for exchanging data (usually in the form of ground atoms). Examples in this category include *nonmonotonic dl-programs* (Eiter *et al.* 2008) and the combination of description logics and defeasible logic (Wang *et al.* 2004). In the *tight integration with semantic separation* approach, $\mathcal{T}$ and $\mathcal{P}$ are more tightly integrated, but the predicates in $\Sigma_{\mathcal{T}}$ and $\Sigma_{\mathcal{P}}$ are kept separate. It builds an integrated model $I$ as the union of a model $I_{\mathcal{T}}$ of $\mathcal{T}$ and a model $I_{\mathcal{P}}$ of $\mathcal{P}$ with the same domain. Examples are $r$-hybrid KB (Rosati 2005), $\mathcal{DL}+log$ (Rosati 2006), $g$-hybrid KB (Heymans *et al.* 2008), and $f$-hybrid KB (Feier & Heymans 2009). Finally, in the *tight integration under a unifying logic* approach, $\mathcal{T}$ and $\mathcal{P}$ are treated uniformly by translating them into a uniform logic, and there is no principled separation between $\Sigma_{\mathcal{T}}$ and $\Sigma_{\mathcal{P}}$. Examples are Hybrid MKNF KB (Motik & Rosati 2010), the first-order autoepistemic logic based integration (de Bruijn *et al.* 2007a), and the QEL-based integration (de Bruijn *et al.* 2007b). This approach is attractive since it provides a seamless integration of DLs and logic programs, and information flow is bi-directional. On the other hand, computational complexity is high.

In this paper, we investigate whether the first-order stable model semantics (Ferraris, Lee, & Lifschitz 2011), which naturally extends both first-order logic and logic programs, can serve as a unifying logic for the integration of rules and ontologies. As the first step, we show how some of the existing integration proposals can be reformulated in terms of the first-order stable model semantics. Our reformulations are simpler than the original proposals in the sense that they do not refer to grounding.

We begin with a review of the first-order stable model semantics, and the notion of semi-safety. Then we present a reformulation of $\mathcal{DL}+log$ in the first-order stable model semantics, and show how the small predicate property, that is ensured by semi-safety, can be used to strengthen some complexity results related to $\mathcal{DL}+log$. Then we relate our approach to the approach in (de Bruijn *et al.* 2007b) that uses Quantified Equilibrium Logic as a unifying logic and, as a corollary, show a reformulation of $g$-hybrid knowledge bases. Next we relate our approach to nonmonotonic dl-programs.

## Background

### Review: First-Order Stable Model Semantics (FOSM)

We assume the following set of primitive propositional connectives and quantifiers:

$$\bot \text{ (falsity)},\ \wedge,\ \vee,\ \rightarrow,\ \forall,\ \exists\ .$$

$\neg F$ is an abbreviation for $F \rightarrow \bot$, symbol $\top$ stands for $\bot \rightarrow \bot$, and $F \leftrightarrow G$ stands for $(F \rightarrow G) \wedge (G \rightarrow F)$.

In (Ferraris, Lee, & Lifschitz 2011) the stable models are defined in terms of the SM operator with *intensional* predicates. For any first-order sentence $F$ and any list $\mathbf{p} = (p_1, \dots, p_n)$ of intensional predicates, SM$[F; \mathbf{p}]$ is defined as [1]

$$F \wedge \neg\exists\mathbf{u}((\mathbf{u} < \mathbf{p}) \wedge F^*(\mathbf{u})),$$

where $\mathbf{u}$ is a list of distinct predicate variables $u_1, \dots, u_n$ and $F^*(\mathbf{u})$ is defined recursively:

[1] Here we use the expression SM$[F; \mathbf{p}]$ in place of SM$_{\mathbf{p}}[F]$ used in (Ferraris, Lee, & Lifschitz 2011).

- $p_i(\mathbf{t})^* = u_i(\mathbf{t})$ for any list $\mathbf{t}$ of terms;
- $F^* = F$ for any atomic formula $F$ that does not contain members of $\mathbf{p}$;
- $(F \odot G)^* = F^* \odot G^*$, where $\odot \in \{\wedge, \vee\}$;
- $(F \rightarrow G)^* = (F^* \rightarrow G^*) \wedge (F \rightarrow G)$;
- $(QxF)^* = QxF^*$, where $Q \in \{\forall, \exists\}$.

The models of SM$[F; \mathbf{p}]$ are called the $\mathbf{p}$-*stable* models of $F$. Intuitively, they are the models of $F$ that are "stable" on $\mathbf{p}$. We often write SM$[F]$ instead of SM$[F; \mathbf{p}]$ when $\mathbf{p}$ is the list of all predicate constants occurring in $F$, and call a model of SM$[F]$ simply a *stable* model of $F$. *Answer sets* are defined as a special class of stable models as follows. By $\sigma(F)$ we denote the signature consisting of the object, function and predicate constants occurring in $F$. By $pr(F)$ we denote the list of all predicate constants occurring in $F$. If $F$ contains at least one object constant, an Herbrand interpretation of $\sigma(F)$ that satisfies SM$[F; pr(F)]$ is called an *answer set* of $F$. The answer sets of a logic program $\Pi$ are defined as the answer sets of the FOL-representation of $\Pi$ (i.e., the conjunction of the universal closures of implications corresponding to the rules). For example, the FOL-representation of the program

$$\begin{array}{l} p(a) \\ q(b) \\ r(x) \leftarrow p(x), \mathit{not}\ q(x) \end{array}$$

is

$$p(a) \wedge q(b) \wedge \forall x((p(x) \wedge \neg q(x)) \rightarrow r(x)). \quad (1)$$

It is shown in (Ferraris, Lee, & Lifschitz 2011) that this definition of an answer set, when applied to the syntax of logic programs, is equivalent to the traditional definition of an answer set, which is based on grounding and fixpoint construction, given in (Gelfond & Lifschitz 1988).

The following proposition tells us that first-order logic formulas and logic programs can be viewed as special cases of SM$[F; \mathbf{p}]$.

**Proposition 1**

*(a) If $F$ is a first-order sentence, then the models of $F$ (in the sense of classical logic) are precisely the models of* SM$[F; \emptyset]$.

*(b) If $F$ is the FOL representation of a logic program $\Pi$, then the answer sets of $\Pi$ in the sense of (Gelfond & Lifschitz 1988) are precisely the Herbrand interpretations of $\sigma(F)$ that satisfy* SM$[F; pr(F)]$.

Part (a) is easy to see; Part (b) is Theorem 1 from (Ferraris, Lee, & Lifschitz 2011) .

## Semi-Safety and Small Predicate Property

We assume that the signature contains no function constants of positive arity. To every quantifier-free formula $F$, we assign a set RV$_{\mathbf{p}}(F)$ of *restricted variables relative to* $\mathbf{p}$ as follows.

- For an atomic formula $F$ (including equality and $\bot$),
  - if $F$ is an equality between two variables, or is an atom whose predicate constant is not in $\mathbf{p}$, then RV$_{\mathbf{p}}(F) = \emptyset$;
  - otherwise, RV$_{\mathbf{p}}(F)$ is the set of all variables occurring in $F$;
- RV$_{\mathbf{p}}(G \wedge H) =$ RV$_{\mathbf{p}}(G) \cup$ RV$_{\mathbf{p}}(H)$;
- RV$_{\mathbf{p}}(G \vee H) =$ RV$_{\mathbf{p}}(G) \cap RV_{\mathbf{p}}(H)$;
- RV$_{\mathbf{p}}(G \rightarrow H) = \emptyset$.

We say that a variable $x$ is $\mathbf{p}$-*restricted* in a quantifier-free formula $F$ if $x \in$ RV$_{\mathbf{p}}(F)$.

Recall that the occurrence of one formula in another is called *positive* if the number of implications containing that occurrence in the antecedent is even, and *negative* otherwise. We say that an occurrence of a predicate constant, a variable, or any other subexpression in a formula $F$ is *strictly positive* if that occurrence is not in the antecedent of any implication.

Consider a sentence $F$ in prenex form:

$$Q_1x_1 \cdots Q_nx_nM \quad (2)$$

(each $Q_i$ is $\forall$ or $\exists$; $x_1, \dots, x_n$ are distinct variables; the matrix $M$ is quantifier-free). It is shown that every first-order formula under the stable model semantics can be turned into prenex form (Lee & Palla 2007, Theorem 2). We say that $F$ is *semi-safe relative to* $\mathbf{p}$ if every strictly positive occurrence of every variable $x_i$ in $M$ belongs to a subformula $G \rightarrow H$ where $x_i$ is $\mathbf{p}$-restricted in $G$.

Proposition 2 below shows that all stable models of a semi-safe sentence have the small predicate property: the relation represented by any of its predicate constants $p_i$ can hold for a tuple of arguments only if each member of the tuple is represented by an object constant occurring in $F$. For any finite set $\mathbf{c}$ of object constants, $in_{\mathbf{c}}(x)$ stands for the formula

$$\bigvee_{c \in \mathbf{c}} x = c.$$

*The small predicate property relative to* $\mathbf{p}$, denoted by $SPP^{\mathbf{p}}_{\mathbf{c}}$, is the conjunction of the sentences

$$\forall v_1, \dots, v_n \Big( p(v_1, \dots, v_n) \rightarrow \bigwedge_{i=1,\dots,n} in_{\mathbf{c}}(v_i) \Big)$$

for all predicate constants $p$ in $\mathbf{p}$, where $v_1, \dots, v_n$ are distinct variables.

The following proposition is an extension of Proposition 1 from (Lee, Lifschitz, & Palla 2009), where $\mathbf{p}$ was limited to the list of predicate constants occurring in $F$. It says that every stable model of a semi-safe sentence satisfies the small predicate property.[2] By $c(F)$ we denote the set of all object constants occurring in $F$.

**Proposition 2** *For any semi-safe sentence $F$ relative to $\mathbf{p}$, formula* SM$[F; \mathbf{p}]$ *entails $SPP^{\mathbf{p}}_{c(F)}$.*

In (Lee, Lifschitz, & Palla 2008; Lee, Lifschitz, & Palla 2009), the definition of *safety* is extended to arbitrary sentences by restricting semi-safe sentences to satisfy a further condition. It is shown there that a safe sentence and the result of its grounding have the same stable models.

[2] This extension is mentioned in (Bartholomew & Lee 2010).

## FOSM Based Integration

DL knowledge bases can be viewed as theories in first-order logic. Given a DL knowledge base $\mathcal{T}$ of signature $\Sigma_{\mathcal{T}}$ and a logic program $\mathcal{P}$ of signature $\Sigma_{\mathcal{P}}$, our approach is to identify the models of the hybrid knowledge base $(\mathcal{T}, \mathcal{P})$ with the interpretations of signature $\Sigma_{\mathcal{T}} \cup \Sigma_{\mathcal{P}}$ (in the sense of classical logic) that satisfy $\text{SM}[\mathit{FO}(\mathcal{T}) \wedge \mathit{FO}(\mathcal{P}); \mathbf{p}]$, where $\mathit{FO}(\mathcal{T})$ and $\mathit{FO}(\mathcal{P})$ are the first-order representations of $\mathcal{T}$ and $\mathcal{P}$ respectively, and $\mathbf{p}$ is a list of intensional predicates. We assume that $\mathcal{T}$ and $\mathcal{P}$ are finite, and so are the predicate constants in $\Sigma_{\mathcal{P}}$. Typically, existing integration approaches assume that the signatures do not contain function constants of positive arity, and $\Sigma_{\mathcal{T}}$ and $\Sigma_{\mathcal{P}}$ share the same set of object constants, but have disjoint sets of predicate constants; typically $\mathbf{p}$ is the list of all predicate constants in $\Sigma_{\mathcal{P}}$.

**Example 1** *(de Bruijn* et al. *2007b, Example 1) Consider a hybrid knowledge base consisting of a classical theory $\mathcal{T}$ :*

$$\forall x(\mathit{PERSON}(x) \rightarrow (\mathit{AGENT}(x) \wedge (\exists y \mathit{HAS\text{-}MOTHER}(x, y))))$$
$$\forall x((\exists y \mathit{HAS\text{-}MOTHER}(x, y)) \rightarrow \mathit{ANIMAL}(x))$$

*which says that every PERSON is an AGENT and has some (unknown) mother, and everyone who has a mother is an ANIMAL, and a nonmonotonic logic program $\mathcal{P}$:*

$$\mathit{PERSON}(x) \leftarrow \mathit{AGENT}(x), \mathit{not}\ machine(x)$$
$$\mathit{AGENT}(DaveB)$$

*which says that AGENTs are by default PERSONs, unless known to be machines, and DaveB is an AGENT.* $\text{SM}[\mathit{FO}(\mathcal{T}) \wedge \mathit{FO}(\mathcal{P});\ \mathit{machine}]$ *entails* $\mathit{Person}(DaveB)$. *Furthermore, it entails each of* $\exists y \mathit{HAS\text{-}Mother}(DaveB, y)$ *and* $\mathit{ANIMAL}(DaveB)$.

## Relating to $\mathcal{DL} + log$ by Rosati

In $\mathcal{DL} + log$, predicate constants are partitioned into *DL predicates* $P_{\mathcal{T}}$ and *Datalog predicates* $P_{\mathcal{P}}$. DL predicates are further partitioned into *concept names* and *role names*. Additionally, $\mathcal{DL} + log$ assumes a countably infinite set of object constants, denoted by $C$.

A $\mathcal{DL} + log$ knowledge base is denoted by $(\mathcal{T}, \mathcal{P})$, where $\mathcal{T}$ is a DL knowledge base of signature $\langle C, P_{\mathcal{T}} \rangle$ and $\mathcal{P}$ is a Datalog program of signature $\langle C, P_{\mathcal{T}} \cup P_{\mathcal{P}} \rangle$ consisting of rules $R$ of the form

$$\begin{aligned} p_1(X_1)\ ;\ \dots\ ;\ p_n(X_n) \leftarrow & \\ & r_1(Y_1), \dots, r_m(Y_m), s_1(Z_1), \dots, s_k(Z_k), \\ & \mathit{not}\ u_1(W_1), \dots, \mathit{not}\ u_h(W_h) \end{aligned} \tag{3}$$

$(n \geq 0, m \geq 0, k \geq 0, h \geq 0)$ where $X_i$, $Y_i$, $Z_i$, $W_i$ are lists of object variables and object constants, and

- each $p_i$ is either a DL predicate or a Datalog predicate;
- each $r_i$, $u_i$ is a Datalog predicate;
- each $s_i$ is a DL predicate;
- (*Datalog safety*) every variable occurring in $R$ must also occur in at least one of the atoms $r_1(Y_1), \dots, r_m(Y_m), s_1(Z_1), \dots, s_k(Z_k)$;
- (*Weak safety*) every variable occurring in the head must also occur in at least one of the atoms $r_1(Y_1), \dots, r_m(Y_m)$.

Rosati (2006) presents two semantics of $\mathcal{DL} + log$ KB: the monotonic and the nonmonotonic semantics. The monotonic semantics of $\mathcal{DL} + log$ is given by simply viewing $\mathcal{T}$ and $\mathcal{P}$ as theories in first-order logic: given a $\mathcal{DL} + log$ knowledge base $(\mathcal{T}, \mathcal{P})$ of signature $\langle C, P_{\mathcal{T}} \cup P_{\mathcal{P}} \rangle$, an interpretation $I$ is a *monotonic model* of $(\mathcal{T}, \mathcal{P})$ if $I$ satisfies $\mathit{FO}(\mathcal{T}) \wedge \mathit{FO}(\mathcal{P})$. In view of Proposition 1 (a), this semantics can be expressed by $\text{SM}[\mathit{FO}(\mathcal{T}) \wedge \mathit{FO}(\mathcal{P}); \emptyset]$.

The nonmonotonic semantics of $\mathcal{DL} + log$ is based on the stable model semantics for disjunctive logic programs. The notation $gr(\mathcal{P}, C)$ represents the ground program obtained by replacing every variable in every rule of $\mathcal{P}$ with every object constant in $C$.

Given $gr(\mathcal{P}, C)$ and an interpretation $I$ of signature $\langle C, P_{\mathcal{T}} \rangle$, the *projection* of $gr(\mathcal{P}, C)$ with respect to $I$, denoted by $\Pi(gr(\mathcal{P}, C), I)$, is obtained as follows. For every rule $R \in gr(\mathcal{P}, C)$,

- delete $R$ if $I \models r(\mathbf{t})$ for some head atom $r(\mathbf{t})$ such that $r \in P_{\mathcal{T}}$;
- delete every atom $r(\mathbf{t})$ in the head such that $r \in P_{\mathcal{T}}$ and $I \not\models r(\mathbf{t})$;
- delete $R$ if $I \not\models r(\mathbf{t})$ for some atom $r(\mathbf{t})$ in the body such that $r \in P_{\mathcal{T}}$;
- delete every atom $r(\mathbf{t})$ in the body such that $r \in P_{\mathcal{T}}$ and $I \models r(\mathbf{t})$.

The $\mathcal{DL} + log$ approach imposes the *standard name assumption*: every interpretation is over the same fixed, countably infinite, domain $\Delta$, and in addition, the set $C$ of object constants is such that it is in the same one-to-one correspondence with $\Delta$ in every interpretation. As a result, for simplicity, we assume that the domain with respect to every interpretation is $C$.

An interpretation $I$ (in the sense of classical logic) of a signature $\sigma$ can be represented as a pair $\langle I^f, X \rangle$, where $I^f$ is the restriction of $I$ to function constants (including object constants) from $\sigma$, and $X$ is the set of atoms, formed using predicate constants from $\sigma$ and the names of elements of $|I|$, which are satisfied by $I$.

Given a $\mathcal{DL} + log$ knowledge base $(\mathcal{T}, \mathcal{P})$ of signature $\langle C, P_{\mathcal{T}} \cup P_{\mathcal{P}} \rangle$, an interpretation $I$ is a *nonmonotonic model* of $(\mathcal{T}, \mathcal{P})$ if

- $I|_C$ (the restriction of $I$ on $C$) is an identity function that maps every constant in $C$ to itself;
- $\langle I|_C, I|_{P_{\mathcal{T}}} \rangle$ satisfies $\mathcal{T}$;
- $\langle I|_C, I|_{P_{\mathcal{P}}} \rangle$, identified with a set of ground atoms, is an answer set of $\Pi(gr(\mathcal{P}, C), \langle I|_C, I|_{P_{\mathcal{T}}} \rangle)$.

The following proposition shows how the nonmonotonic semantics of $\mathcal{DL} + log$ can be reformulated in terms of the first-order stable model semantics.

**Proposition 3** *For any $\mathcal{DL} + log$ knowledge base $(\mathcal{T}, \mathcal{P})$, under the standard name assumption, the nonmonotonic models of $(\mathcal{T}, \mathcal{P})$ according to (Rosati 2006) are precisely the interpretations of $\langle C, P_{\mathcal{T}} \cup P_{\mathcal{P}} \rangle$ that satisfy*

$$\text{SM}[\mathit{FO}(\mathcal{T}) \wedge \mathit{FO}(\mathcal{P});\ P_{\mathcal{P}}].$$

Since the reformulation does not refer to grounding, arguably, it provides a simpler account of $\mathcal{DL} + log$.

## On the Safety Conditions imposed in $\mathcal{DL} + log$

Recall that $\mathcal{DL} + log$ imposes weak safety (every variable occurring in the head of a rule also occurs in a Datalog atom in the positive body) and Datalog safety (every variable occurring in a rule also occurs in the positive body), which, even when combined, yields a condition that is weaker than DL-safety (Motik, Sattler, & Studer 2005), where every variable occurring in a rule is also required to occur in a datalog atom in the positive body.

Note that in Proposition 3 we identify Datalog predicates with intensional predicates, and DL predicates with non-intensional predicates. Clearly, under this view, the definition of semi-safety presented earlier coincides with the definition of weak-safety for programs whose rules have the form (3). In fact, by using the property ensured by semi-safety, below we show that weak safety is a sufficient condition for guaranteeing decidability of reasoning with $\mathcal{DL} + log$.

**Proposition 4** *Let $\mathcal{K} = (\mathcal{T}, \mathcal{P})$ be a $\mathcal{DL} + log$ knowledge base such that $\mathcal{P}$ is weakly safe but is not necessarily datalog safe. Let $\mathcal{P}'$ be the program obtained from $\mathcal{P}$ by removing in every rule, all the negative datalog literals that contain a variable that occurs only in the negative body. Then $\mathcal{K}$ is equivalent (under the nonmonotonic semantics) to the $\mathcal{DL} + log$ knowledge base $(\mathcal{T}, \mathcal{P}')$.*

Since the complexity of the transformation required to obtain $\mathcal{P}'$ is polynomial in the size of $\mathcal{P}$, the decidability results (Theorems 11 and 12 from (Rosati 2006)) and the complexity results (Theorem 13 from (Rosati 2006)) with respect to the nonmonotonic semantics can be straightforwardly carried over to $\mathcal{DL} + log$ knowledge bases $(\mathcal{T}, \mathcal{P})$ where $\mathcal{P}$ is weakly safe but not necessarily datalog safe. In other words, in terms of decidability and complexity results mentioned above, the requirement of datalog safety can be dropped.

The proof of Proposition 4 uses the following lemma, which is a slight extension of Theorem 3 from (Ferraris, Lee, & Lifschitz 2011).

**Lemma 1** *If $G$ does not contain any strictly positive occurrences of predicates in $\mathbf{p}$, then* $\mathrm{SM}[F \wedge G; \mathbf{p}]$ *is equivalent to* $\mathrm{SM}[F; \mathbf{p}] \wedge G$.

**Proof of Proposition 4** If $\mathcal{P}$ is weakly safe and datalog safe, then $\mathcal{P}'$ is the same as $\mathcal{P}$. Assume that $\mathcal{P}$ is weakly safe but not datalog safe. Then there is a rule that contains some variable $y$ that occurs only in a negative datalog literal. $\mathcal{P}'$ is obtained from $\mathcal{P}$ by removing all negative datalog literals that contain such a variable $y$. By Lemma 1, since $\mathit{FO}(\mathcal{T})$ contains no predicate from $P_{\mathcal{P}}$,

$$\mathrm{SM}[\mathit{FO}(\mathcal{T}) \wedge \mathit{FO}(\mathcal{P}); P_{\mathcal{P}}]$$

is equivalent to

$$\mathit{FO}(\mathcal{T}) \wedge \mathrm{SM}[\mathit{FO}(\mathcal{P}); P_{\mathcal{P}}].$$

Similarly

$$\mathrm{SM}[\mathit{FO}(\mathcal{T}) \wedge \mathit{FO}(\mathcal{P}'); P_{\mathcal{P}}]$$

is equivalent to

$$\mathit{FO}(\mathcal{T}) \wedge \mathrm{SM}[\mathit{FO}(\mathcal{P}'); P_{\mathcal{P}}].$$

So it is sufficient to show that $\mathrm{SM}[\mathit{FO}(\mathcal{P}); P_{\mathcal{P}}]$ is equivalent to $\mathrm{SM}[\mathit{FO}(\mathcal{P}'); P_{\mathcal{P}}]$. Since $\mathcal{P}$ is semi-safe relative to $P_{\mathcal{P}}$, $\mathcal{P}'$ is also semi-safe relative to $P_{\mathcal{P}}$. By Proposition 2 above and Theorem 9 from (Ferraris, Lee, & Lifschitz 2011), it is sufficient to show that under the assumption $\mathit{SPP}^{P_{\mathcal{P}}}_{c(\mathcal{P})}$, where $c(\mathcal{P})$ is the set of object constants occurring in $\mathcal{P}$,

$$(\mathbf{q} \le P_{\mathcal{P}}) \rightarrow (\mathit{FO}(\mathcal{P})^*(\mathbf{q}) \leftrightarrow \mathit{FO}(\mathcal{P}')^*(\mathbf{q})) \qquad (4)$$

is logically valid. Given a rule, let $F(\mathbf{y})$ be the conjunction of negative datalog literals that contain a variable occurring only in a negative datalog literal, and $\mathbf{y}$ is the list of all such variables. Formula (4) is logically valid, since $(\exists \mathbf{y} F(\mathbf{y}))^*(\mathbf{q})$ is equivalent to $\exists \mathbf{y} F(\mathbf{y})$ under the assumption $\mathbf{q} \le P_{\mathcal{P}}$, and $\exists \mathbf{y} F(\mathbf{y})$ is equivalent to $\top$ under the assumption $\mathit{SPP}^{P_{\mathcal{P}}}_{c(\mathcal{P})}$ (the extents of predicates in $P_{\mathcal{P}}$ are finite) and the standard name assumption (there are infinitely many objects in the domain). ∎

## Relating to QEL with hybrid Rules

We refer the reader to Section A.5.1 from (Ferraris, Lee, & Lifschitz 2011) for a review of quantified equilibrium logic (QEL). By $\mathit{Choice}(\mathbf{p})$ we denote the conjunction of "choice formulas" $\forall \mathbf{x}(p(\mathbf{x}) \vee \neg p(\mathbf{x}))$ for all predicate constants $p$ in $\mathbf{p}$ where $\mathbf{x}$ is a list of distinct object variables whose length is the same as the arity of $p$. According to the QEL-based integration approach in (de Bruijn *et al.* 2007b), an HT-interpretation $\mathcal{I}$ of signature $\langle C, P_{\mathcal{T}} \cup P_{\mathcal{P}} \rangle$ is a model of the hybrid knowledge $\mathcal{K} = (\mathcal{T}, \mathcal{P})$ iff it is an equilibrium model of $\mathit{FO}(\mathcal{T}) \wedge \mathit{FO}(\mathcal{P}) \wedge \mathit{Choice}(P_{\mathcal{T}})$. Formula $\mathit{FO}(\mathcal{T}) \wedge \mathit{FO}(\mathcal{P}) \wedge \mathit{Choice}(P_{\mathcal{T}})$ is called the *stable closure* of $\mathcal{K}$. The following proposition shows the relationship between the QEL-based approach and our approach.

**Proposition 5** *For any hybrid knowledge base $\mathcal{K} = (\mathcal{T}, \mathcal{P})$ of signature $\langle C, P_{\mathcal{T}} \cup P_{\mathcal{P}} \rangle$, an HT-interpretation $\langle J, X, X \rangle$ of the same signature is an equilibrium model of $\mathcal{K}$ in the sense of (de Bruijn* et al. *2007b) iff $\langle J, X \rangle$ satisfies*

$$\mathrm{SM}[\mathit{FO}(\mathcal{T}) \wedge \mathit{FO}(\mathcal{P});\ P_{\mathcal{P}}].$$

The proof of the proposition is immediate from Lemma 9 from (Ferraris, Lee, & Lifschitz 2011), which establishes the relationship between equilibrium logic and $\mathrm{SM}[F; \mathbf{p}]$ for the special case when $\mathbf{p}$ is the list of all predicate constants in the signature, and the following proposition, which tells us that the set of intensional predicates can be increased by using choice formulas. Recall that by $\mathit{pr}(F)$ we denote the list of all predicate constants occurring in $F$. By $\mathit{False}(\mathbf{p})$ we denote the conjunction of $\forall \mathbf{x} \neg p(\mathbf{x})$ for all predicate constants $p$ in $\mathbf{p}$.

**Proposition 6** *Formula*

$$\mathrm{SM}[F; \mathbf{p}] \leftrightarrow \mathrm{SM}[F \wedge \mathit{Choice}(\mathit{pr}(F) \backslash \mathbf{p}) \wedge \mathit{False}(\mathbf{p} \backslash \mathit{pr}(F))]$$

*is logically valid.*

## Relating to $g$-hybrid KBs

de Bruijin *et al.* (2007b) relate the QEL approach to $r$-hybrid (Rosati 2005) and $g$-hybrid knowledge bases (Heymans *et* *al.* 2008). As a corollary of Proposition 5 in this paper, we can thus relate our approach to $r$-hybrid and $g$-hybrid knowledge bases. Here we present the result only with respect to $g$-hybrid knowledge bases since we already covered the relationship to $\mathcal{DL} + \mathit{log}$, which is an extension of $r$-hybrid knowledge bases.

$g$-hybrid knowledge bases are based on the *open answer set programming* (Heymans, Nieuwenborgh, & Vermeir 2005) approach. More specifically, a $g$-hybrid knowledge base is a pair $(\mathcal{T}, \mathcal{P})$, where $\mathcal{T}$ is a DL knowledge base of signature $\langle C, P_{\mathcal{T}} \rangle$ and $\mathcal{P}$ is a *guarded* program of signature $\langle C, P_{\mathcal{T}} \cup P_{\mathcal{P}} \rangle$ such that $P_{\mathcal{T}} \cap P_{\mathcal{P}} = \emptyset$. A program is said to be *guarded* if, for all rules $R$ that are not of the form

$$p(\mathbf{t}) \vee \mathit{not}\, p(\mathbf{t}) \leftarrow\ , \tag{5}$$

there exists an atom $A$ in the positive body (known as the *guard*) such that all the variables occurring in $R$ also occur in $A$. This implies that, in addition to such safe rules $R$, guarded programs also allow unsafe choice rules of the form (5). Also, guarded programs allow negation in the head but with the restriction that there can be at most one non-negated atom in the head.

Given an interpretation $I$ of signature $\langle C, P_{\mathcal{T}} \cup P_{\mathcal{P}} \rangle$, program $\mathcal{P}_I$ is defined as the ground program obtained from $\mathcal{P}$ by first replacing every occurrence of $c$ from $C$ in it with $c^I$ and then grounding the resulting program with respect to the universe of $I$. Interpretation $I$ is a model of the $g$-hybrid knowledge base $(\mathcal{T}, \mathcal{P})$ if

- the restriction of $I$ to $\langle C, P_{\mathcal{T}} \rangle$ is a model of $\mathcal{T}$, and
- the restriction of $I$ to $\langle C, P_{\mathcal{P}} \rangle$, viewed as a set of ground atoms, is an answer set of $\Pi(\mathcal{P}_I, I)$.[3]

The following proposition is a corollary to Theorem 2 of (de Bruijn *et al.* 2007b) and Proposition 5.

**Proposition 7** *For any $g$-hybrid knowledge base $\mathcal{K} = (\mathcal{T}, \mathcal{P})$, an interpretation $I$ of signature $\langle C, P_{\mathcal{T}} \cup P_{\mathcal{P}} \rangle$ is a model of $\mathcal{K}$ in the sense of (de Bruijn* et al. *2007b) iff $I$ is a model of*

$$\mathrm{SM}[\mathit{FO}(\mathcal{T}) \wedge \mathit{FO}(\mathcal{P});\ P_{\mathcal{P}}].$$

## Relating to Nonmonotonic dl-programs by Eiter *et al.*

### Review: Nonmonotonic dl-programs

We first review the syntax and the semantics of dl-programs. For simplicity, we do not allow strong negation. A nonmonotonic *dl-program* (Eiter *et al.* 2008) is a pair $(\mathcal{T}, \mathcal{P})$, where $\mathcal{T}$ is a DL knowledge base of signature $\langle C, P_{\mathcal{T}} \rangle$ and $\mathcal{P}$ is a *generalized* normal logic program of signature $\langle C, P_{\mathcal{P}} \rangle$ such that $P_{\mathcal{T}} \cap P_{\mathcal{P}} = \emptyset$. A generalized normal logic program is a set of *dl-rules* that can contain queries to $\mathcal{T}$ in their bodies, in the form of *dl-atoms*.

A *dl-atom* is of the form

$$DL[S_1 op_1 p_1, \ldots, S_m op_m p_m;\ Q](\mathbf{t}) \quad (m \geq 0) \tag{6}$$

where each $S_i$ is a concept, a role or a special symbol $\theta \in \{=, \neq\}$, symbol $p_i$ is a unary predicate constant if $S_i$ is a concept and a binary predicate constant otherwise and $op_i \in \{\oplus, \odot, \ominus\}$; $Q(\mathbf{t})$ is a dl-query (Eiter *et al.* 2008).

A *dl-rule* is of the form

$$a \leftarrow b_1, \ldots, b_k, \mathit{not}\ b_{k+1}, \ldots, \mathit{not}\ b_m \tag{7}$$

where $a$ is an atom and each $b_i$ is either an atom, equality, or a dl-atom. We identify rule (7) with

$$a \leftarrow B, N \tag{8}$$

where $B$ is $b_1, \ldots, b_k$ and $N$ is $\mathit{not}\ b_{k+1}, \ldots, \mathit{not}\ b_m$.

The semantics of dl-programs is defined by extending the answer set semantics to generalized programs. In order to do this, the definition of satisfaction is extended to ground dl-atoms. An Herbrand interpretation $I$ *satisfies* a ground atom $A$ *relative to* $\mathcal{T}$ if $I$ satisfies $A$. An Herbrand interpretation $I$ *satisfies* a ground dl-atom (6) *relative to* $\mathcal{T}$ if $\mathcal{T} \cup \bigcup_{i=1}^{m} A_i(I)$ entails $Q(\mathbf{t})$, where $A_i(I)$ is

- $\{S_i(\mathbf{e}) \mid p_i(\mathbf{e}) \in I\}$ if $op_i$ is $\oplus$,
- $\{\neg S_i(\mathbf{e}) \mid p_i(\mathbf{e}) \in I\}$ if $op_i$ is $\odot$,
- $\{\neg S_i(\mathbf{e}) \mid p_i(\mathbf{e}) \notin I\}$ if $op_i$ is $\ominus$,

and $\mathbf{t}$ is any list of ground terms. The satisfaction relation is extended to allow connectives in the usual way.

Given a dl-program $(\mathcal{T}, \mathcal{P})$, the *weak dl-transform* of $\mathcal{P}$ relative to $\mathcal{T}$ and an Herbrand interpretation $I$ of $\langle C, P_{\mathcal{P}} \rangle$, denoted by $w\mathcal{P}^I_{\mathcal{T}}$, is the logic program obtained from $gr(\mathcal{P}, C)$ by deleting

- each rule (8) in $gr(\mathcal{P}, C)$ such that
  - $I \not\models_{\mathcal{T}} b$ for some dl-atom $b$ in $B$, or
  - $I \models_{\mathcal{T}} b$ for some literal *not* $b$ in $N$;
- from each remaining dl-rule (8), all the dl-atoms in $B$ and all the literals in $N$.

$I$ is a *weak answer set* of $(\mathcal{T}, \mathcal{P})$ if $I$ is the minimal model of $w\mathcal{P}^I_{\mathcal{T}}$.

By $DL^?_{\mathcal{P}}$ we denote the set of dl-atoms in $gr(\mathcal{P}, C)$ that are not known to be monotonic. The *strong dl-transform* of $\mathcal{P}$ relative to $\mathcal{T}$ and $I$, denoted by $s\mathcal{P}^I_{\mathcal{T}}$, is the logic program obtained from $gr(\mathcal{P}, C)$ by deleting

- each rule (8) in $gr(\mathcal{P}, C)$ such that
  - $I \not\models_{\mathcal{T}} b$ for some dl-atom $b$ in $B \cap DL^?_{\mathcal{P}}$ or
  - $I \models_{\mathcal{T}} b$ for some *not* $b$ in $N$;
- from each remaining dl-rule (8), all the dl-atoms in $B \cap DL^?_{\mathcal{P}}$ and all the literals in $N$.

$I$ is a *strong answer set* of $(\mathcal{T}, \mathcal{P})$ if $I$ is the minimal model of $s\mathcal{P}^I_{\mathcal{T}}$.

### Relating to Nonmonotonic dl-Programs

In order to relate our approach to the semantics of dl-programs, we extend the SM operator to formulas that allow dl-atoms. The treatment is similar to the extension of SM to formulas that allow aggregates, as given in (Lee & Meng 2009; Ferraris & Lifschitz 2010).

[3] The definition of projection $\Pi$ given earlier is straightforwardly extended to cover a rule like (5) that allows *not* in the head.

We define *dl-formulas* of the signature $\langle C, P_\mathcal{T} \cup P_\mathcal{P}\rangle$ as an extension of first-order formulas by treating dl-atoms as a base case in addition to standard atomic formulas formed from $\langle C, P_\mathcal{P}\rangle$. Note that any generalized normal logic program can be viewed as a dl-formula. We extend the notion *FO*$(\mathcal{P})$ to a generalized normal logic program $\mathcal{P}$ in a straightforward way. Also the SM operator is extended to dl-formulas by extending the recursive definition $F^*$ by adding the clause

$$\begin{array}{l} DL[S_1 op_1 p_1, \dots, S_m op_m p_m; Q](\mathbf{t})^*(\mathbf{u}) = \\ \qquad DL[S_1 op_1 p'_1, \dots, S_m op_m p'_m; Q](\mathbf{t}) \wedge \\ \qquad DL[S_1 op_1 p_1, \dots, S_m op_m p_m; Q](\mathbf{t}) \end{array}$$

where symbol $p'_i$ is $u_i$ if $p_i$ is intensional and $p_i$ otherwise.

The following proposition shows how strong answer sets can be characterized by this extension of SM.

**Proposition 8** *For any dl-program $(\mathcal{T}, \mathcal{P})$ such that every occurrence of $\ominus$ is in the scope of negation, the strong answer sets of $(\mathcal{T}, \mathcal{P})$ are precisely the Herbrand interpretations of $\langle C, P_\mathcal{P}\rangle$ that satisfy* SM$[FO(\mathcal{P});\ P_\mathcal{P}]$ *relative to $\mathcal{T}$.*

The proof uses the following lemma, which extends Lemma 6 from (Ferraris, Lee, & Lifschitz 2011) to dl-formulas. In the following, $X$ and $Y$ are Herbrand interpretations of $\langle C, P_\mathcal{P}\rangle$ such that $Y$ is a subset of $X$ (we identify an Herbrand interpretation with the set of ground atoms that are true in it), $\mathbf{q}$ is a list of new predicate constants of the same length as $P_\mathcal{P}$, and $Y^{P_\mathcal{P}}_\mathbf{q}$ is obtained from $Y$ by replacing every predicate constant in $P_\mathcal{P}$ with the corresponding predicate constant in $\mathbf{q}$.

**Lemma 2** *For any dl-formula $F$, $X \cup Y^{P_\mathcal{P}}_\mathbf{q} \models_\mathcal{T} (\neg F)^*(\mathbf{q})$ iff $X \models_\mathcal{T} \neg F$.*

**Proof of Proposition 8** Without loss of generality, let us assume that $\mathcal{P}$ is a variable-free program obtained by grounding. It is clear that $X \models_\mathcal{T} \mathcal{P}$ iff $X \models_\mathcal{T} s\mathcal{P}^X_\mathcal{T}$. If $X \not\models_\mathcal{T} \mathcal{P}$, then $X$ is not a strong answer set of $\langle \mathcal{T}, \mathcal{P}\rangle$, and $X$ does not satisfy SM$[FO(\mathcal{P});\ P_\mathcal{P}]$ relative to $\mathcal{T}$.

Assume $X \models_\mathcal{T} \mathcal{P}$. It is sufficient to prove that, for any rule (8) in $\mathcal{P}$, and any Herbrand interpretation $Y$ that is a subset of $X$,

$$Y \models_\mathcal{T} s(a \leftarrow B, N)^X_\mathcal{T}$$

iff

$$X \cup Y^{P_\mathcal{P}}_\mathbf{q} \models_\mathcal{T} (B \wedge N)^*(\mathbf{q}) \rightarrow a^*(\mathbf{q}).$$

*Case 1:* $s(a \leftarrow B, N)^X_\mathcal{T}$ is empty. Clearly, $Y \models_\mathcal{T} s(a \leftarrow B, N)^X_\mathcal{T}$. Since no dl-atom in $B$ mentions $\ominus$, $B$ contains no dl-atoms from $DL^?_\mathcal{P}$, and it follows that $X \models_\mathcal{T} b$ for some *not* $b$ in $N$. Consequently, by Lemma 2, it follows that $X \cup Y^{P_\mathcal{P}}_\mathbf{q} \not\models_\mathcal{T} (\neg b)^*(\mathbf{q})$, so that $X \cup Y^{P_\mathcal{P}}_\mathbf{q} \models_\mathcal{T} (B \wedge N)^*(\mathbf{q}) \rightarrow a^*(\mathbf{q})$.

*Case 2:* $s(a \leftarrow B, N)^X_\mathcal{T}$ is not empty. Since no dl-atom in $B$ mentions $\ominus$, $B$ contains no dl-atoms from $DL^?_\mathcal{P}$, and it follows that $s(a \leftarrow B, N)^X_\mathcal{T}$ is $a \leftarrow B$. Also it follows that $X \not\models_\mathcal{T} b$ for every *not* $b$ in $N$, so that by Lemma 2, we get $X \cup Y^{P_\mathcal{P}}_\mathbf{q} \models_\mathcal{T} N^*(\mathbf{q})$. Since $a$ is an atom and $B$ is a set of atoms and monotonic dl-atoms, it is clear that $Y \models_\mathcal{T} a \leftarrow B$ iff $X \cup Y^{P_\mathcal{P}}_\mathbf{q} \models_\mathcal{T} B^*(\mathbf{q}) \rightarrow a^*(\mathbf{q})$. ∎

The syntactic condition about $\ominus$ in Proposition 8 ensures that all dl-atoms in the positive bodies of the ground program are monotonic. The proposition does not hold if the condition is dropped. For example, consider the dl-program $(\mathcal{T}, \mathcal{P})$ where the axioms in $\mathcal{T}$ (written as first-order formulas) are

$$\begin{array}{c} Q(b), \\ \forall x(\neg S(x) \rightarrow Q(x)) \end{array}$$

and $\mathcal{P}$ is

$$p(x) \leftarrow DL[S \ominus p; Q](x).$$

With $C = \{a, b\}$, note that $DL[S \ominus p; Q](b)$ is monotonic but $DL[S \ominus p; Q](a)$ is not; $(\mathcal{T}, \mathcal{P})$ has no strong answer sets, but $\{p(a), p(b)\}$ is an Herbrand model of SM$[FO(\mathcal{P}); p]$.

In the case of weak answer set semantics, the condition is not required, but instead we need to prepend $\neg\neg$ to all dl-atoms.

**Proposition 9** *For any dl-program $(\mathcal{T}, \mathcal{P})$, the weak answer sets of $(\mathcal{T}, \mathcal{P})$ are precisely the Herbrand interpretations of signature $\langle C, P_\mathcal{P}\rangle$ that satisfy* SM$[FO(\mathcal{P})^{\neg};\ P_\mathcal{P}]$ *relative to $\mathcal{T}$, where $FO(\mathcal{P})^{\neg}$ is obtained from $FO(\mathcal{P})$ by prepending $\neg\neg$ to all occurrences of dl-atoms.*

**Proof**. Similar to the proof of Proposition 8, using the fact that, by Lemma 2, for any dl-atom $A$, $X \cup Y^{P_\mathcal{P}}_\mathbf{q} \models_\mathcal{T} (\neg\neg A)^*(\mathbf{q})$ iff $X \models_\mathcal{T} \neg\neg A$. ∎

Note that our treatment of dl-programs uses the extension of $F^*$ to cover dl-atoms. It is an open question whether dl-atoms can be "unfolded" into first-order formulas, which would make this extension unnecessary. It is worthwhile to note that an encoding of dl-atoms in terms of first-order formulas similar to the one in MKNF (Motik & Rosati 2010, Definition 7.5) does not apply here. For example, consider $\mathcal{K} = (\mathcal{T}, \mathcal{P})$ such that $\mathcal{T}$ is empty, and $\mathcal{P}$ is the following:

$$\begin{array}{rcl} p(a) & \leftarrow & DL[Q \oplus p; R](a) \\ & \leftarrow & \textit{not}\ p(a). \end{array} \tag{9}$$

Since $DL[Q \oplus p; R](a)$ cannot be satisfied by any set of atoms formed using $\langle a, p/1\rangle$, $\mathcal{P}$ has neither strong nor weak answer sets. On the other hand, without going into details, if we represent $DL[Q \oplus p; R](a)$ by $\forall x(p(x) \rightarrow Q(x)) \rightarrow R(a))$, the "FOL-representation" of (9) is

$$F = \big((\forall x(p(x) \rightarrow Q(x)) \rightarrow R(a)) \rightarrow p(a)\big) \wedge \neg\neg p(a)$$

and SM$[F; p]$ is satisfiable.

## Conclusion

In this paper, we investigate whether the stable model semantics from (Ferraris, Lee, & Lifschitz 2011), which distinguishes between intensional and extensional predicates, is a natural formalism to integrate rules and ontologies. Since the first-order stable model semantics is a generalization of the traditional stable model semantics (Gelfond & Lifschitz 1988) to first-order formulas, it enables a rather simple and straightforward integration of ASP rules and ontologies. Recent work on the first-order stable model semantics helps us in studying the semantic properties and computational aspects of the hybrid knowledge bases. For example, the splitting theorem presented in (Ferraris *et al.* 2009) is used in proving Proposition 3 that shows the relationship between $\mathcal{DL} + log$ and our approach. Also, as already discussed, the concept of semi-safety studied under the first-order stable model semantics coincides with the concept of weak safety in $\mathcal{DL} + log$ and the results on semi-safety can be used to show that weak safety is a sufficient condition for ensuring the decidability of reasoning in $\mathcal{DL} + log$. Further, as discussed in (de Bruijn *et al.* 2007b), the notion of strong equivalence can be applied to provide the notion of equivalence between hybrid knowledge bases.

As part of the future work, we plan to investigate the relationship between hybrid MKNF knowledge bases (Motik & Rosati 2010) and our approach.

## Acknowledgements

We are grateful to anonymous referees for their useful comments on this paper. The authors were partially supported by the National Science Foundation under Grants IIS-0916116 and by the IARPA SCIL program.

## References

Bartholomew, M., and Lee, J. 2010. A decidable class of groundable formulas in the general theory of stable models. In *Proceedings of International Conference on Principles of Knowledge Representation and Reasoning (KR)*, 477–485.

de Bruijn, J.; Eiter, T.; Polleres, A.; and Tompits, H. 2007a. Embedding non-ground logic programs into autoepistemic logic for knowledge-base combination. In *Proceedings of International Joint Conference on Artificial Intelligence (IJCAI)*, 304–309.

de Bruijn, J.; Pearce, D.; Polleres, A.; and Valverde, A. 2007b. Quantified equilibrium logic and hybrid rules. In *RR*, 58–72.

Eiter, T.; Ianni, G.; Lukasiewicz, T.; Schindlauer, R.; and Tompits, H. 2008. Combining answer set programming with description logics for the semantic web. *Artificial Intelligence* 172(12-13):1495–1539.

Feier, C., and Heymans, S. 2009. Hybrid reasoning with forest logic programs. In *ESWC*, 338–352.

Ferraris, P., and Lifschitz, V. 2010. The stable model semantics for first-order formulas with aggregates[4]. In *Proceedings of International Workshop on Nonmonotonic Reasoning (NMR)*.

Ferraris, P.; Lee, J.; Lifschitz, V.; and Palla, R. 2009. Symmetric splitting in the general theory of stable models. In *Proceedings of International Joint Conference on Artificial Intelligence (IJCAI)*, 797–803.

Ferraris, P.; Lee, J.; and Lifschitz, V. 2011. Stable models and circumscription. *Artificial Intelligence* 175:236–263.

Gelfond, M., and Lifschitz, V. 1988. The stable model semantics for logic programming. In Kowalski, R., and Bowen, K., eds., *Proceedings of International Logic Programming Conference and Symposium*, 1070–1080. MIT Press.

Heymans, S.; de Bruijn, J.; Predoiu, L.; Feier, C.; and Nieuwenborgh, D. V. 2008. Guarded hybrid knowledge bases. *TPLP* 8(3):411–429.

Heymans, S.; Nieuwenborgh, D. V.; and Vermeir, D. 2005. Guarded open answer set programming. In *Procedings of International Conference on Logic Programming and Nonmonotonic Reasoning (LPNMR)*, 92–104.

Lee, J., and Meng, Y. 2009. On reductive semantics of aggregates in answer set programming. In *Procedings of International Conference on Logic Programming and Nonmonotonic Reasoning (LPNMR)*, 182–195.

Lee, J., and Palla, R. 2007. Yet another proof of the strong equivalence between propositional theories and logic programs. In *Working Notes of the Workshop on Correspondence and Equivalence for Nonmonotonic Theories*.

Lee, J.; Lifschitz, V.; and Palla, R. 2008. Safe formulas in the general theory of stable models (preliminary report). In *Proceedings of International Conference on Logic Programming (ICLP)*, 672–676.

Lee, J.; Lifschitz, V.; and Palla, R. 2009. Safe formulas in the general theory of stable models[5]. Unpublished Draft.

Motik, B., and Rosati, R. 2010. Reconciling description logics and rules. *J. ACM* 57(5).

Motik, B.; Sattler, U.; and Studer, R. 2005. Query answering for owl-dl with rules. *J. Web Sem.* 3(1):41–60.

Nazarenko, A.; Polo, L.; Eiter, T.; de Bruijn, J.; Schwichtenberg, A.; and Heymans, S. 2010. How to integrate ontologies and rules? Tutorial, 24th Conference on Artificial Intelligence (AAAI 2010).

Rosati, R. 2005. On the decidability and complexity of integrating ontologies and rules. *J. Web Sem.* 3(1):61–73.

Rosati, R. 2006. DL+log: Tight integration of description logics and disjunctive datalog. In *Proceedings of International Conference on Principles of Knowledge Representation and Reasoning (KR)*, 68–78.

Wang, K.; Billington, D.; Blee, J.; and Antoniou, G. 2004. Combining description logic and defeasible logic for the semantic web. In *RuleML*, 170–181.

[4] http://userweb.cs.utexas.edu/users/vl/papers/smaf.pdf .

[5] http://peace.eas.asu.edu/joolee/papers/safety.pdf .