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 defined by Ferraris, Lee and Lifschitz. We show that a few existing integration proposals can be uniformly related to the first-order stable model semantics.

1 Introduction

Integrating nonmonotonic rules and ontologies has received much attention, especially in the context of the Semantic Web. A *hybrid knowledge base (hybrid KB)* is a pair $(\mathcal{T}, \mathcal{P})$ where $\mathcal{T}$ is a first-order logic (FOL) knowledge base (typically in a description logic (DL)) and $\mathcal{P}$ is a logic program. The existing integration approaches can be classified into three categories [1]. In the *loose integration* approach (e.g., [1]), $\mathcal{T}$ is viewed as an external source of information with its own semantics that can be accessed by entailment-based query interfaces from $\mathcal{P}$. In the *tight integration with semantic separation* approach (e.g.,[2; 3; 4]), the semantics of logic programs are adapted to allow predicates of $\mathcal{T}$ in the rules, thereby leading to a more tight coupling. On the other hand, a model of the hybrid KB is constructed by the union of a model of $\mathcal{T}$ and a model of $\mathcal{P}$. In the *tight integration under a unifying logic* approach (e.g.,[5; 6]), $\mathcal{T}$ and $\mathcal{P}$ are treated uniformly as they are embedded into a unifying nonmonotonic logic.

Typically, existing integration approaches assume that the underlying signature does not contain function constants of positive arity. We represent the signature by $\langle C, P\rangle$ where C is a set of object constants and P is a set of predicate constants. Formally, a hybrid KB $(\mathcal{T}, \mathcal{P})$ of the signature $\langle C, P_\mathcal{T} \cup P_\mathcal{P}\rangle$ where $P_\mathcal{T}$ and $P_\mathcal{P}$ are disjoint sets of predicate constants, consists of a first-order logic knowledge base $\mathcal{T}$ of signature $\langle C, P_\mathcal{T}\rangle$ and a logic program $\mathcal{P}$ of signature $\langle C, P_\mathcal{T} \cup P_\mathcal{P}\rangle$.

In this paper, we investigate whether the first-order stable model semantics (FOSM) [7], 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 well-known integration proposals from each category, namely, nonmonotonic dl-programs [1] (loose integration), $\mathcal{DL} + log$

J. Delgrande and W. Faber (Eds.): LPNMR 2011, LNAI 6645, pp. 248–253, 2011.

[3] (tight integration with semantic separation), and quantified equilibrium logic based integration [5] (tight integration under a unifying logic), can be related to the first-order stable model semantics.

2 FOSM Based Hybrid KB

We refer the reader to [7] for the definition of the first-order stable model semantics, which applies to any first-order sentence. There the stable models of a first-order sentence F relative to a list $\mathbf{p}$ of predicates are defined as the models of the second-order sentence $\text{SM}[F;\mathbf{p}]$ (in the sense of classical logic). Syntactically, $\text{SM}[F;\mathbf{p}]$ is the formula

$$F \wedge \neg\exists\mathbf{u}((\mathbf{u} < \mathbf{p}) \wedge F^*(\mathbf{u})), \tag{1}$$

where $\mathbf{u}$ is a list of predicate variables corresponding to $\mathbf{p}$, and F^* is defined recursively (See [7] for the details). In general, $\mathbf{p}$ is any list of predicate constants called *intensional* predicates—the predicates that we "intend to characterize" by F. Logic programs are identified as a special class of first-order theories by turning them into their *FOL-representations*. In [7], it is shown that the answer sets of a logic program $\mathcal{P}$ are precisely the Herbrand interpretations that satisfy $\text{SM}[F;\mathbf{p}]$, where F is the FOL-representation of $\mathcal{P}$ and $\mathbf{p}$ is the list of all predicate constants occurring in $\mathcal{P}$. In another special case when $\mathbf{p}$ is empty, $\text{SM}[F;\mathbf{p}]$ is equivalent to F. Consequently, both logic programs and first-order logic formulas can be viewed as special cases of $\text{SM}[F;\mathbf{p}]$ depending on the choice of intensional predicates $\mathbf{p}$. As we show below, the distinction between intensional and non-intensional predicates is useful in characterizing hybrid KBs.

Throughout this paper, we assume that a hybrid KB contains finitely many rules in $\mathcal{P}$[1]. We identify a hybrid KB $(\mathcal{T}, \mathcal{P})$ of signature $\langle C, P_{\mathcal{T}} \cup P_{\mathcal{P}}\rangle$ with the second-order sentence $\text{SM}[FO(\mathcal{T}) \wedge FO(\mathcal{P}); P_{\mathcal{P}}]$ of the same signature, where $FO(\mathcal{T})$ ($FO(\mathcal{P})$, respectively) is the first-order logic (FOL) representation of $\mathcal{T}$ ($\mathcal{P}$, respectively).

Example 1. [5, Example 1] Consider a hybrid KB consisting of a first-order logic theory $\mathcal{T}$

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

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

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

[1] This is for simplicity of applying SM. Alternatively we may extend SM to (possibly infinite) sets of formulas.

($AGENT$s are by default $PERSON$s, unless known to be $machine$s, and $DaveB$ is an $AGENT$). Here $P_{\mathcal{T}}$ is $\{PERSON, AGENT, HAS\text{-}MOTHER, ANIMAL\}$, and $P_{\mathcal{P}}$ is $\{machine\}$. Formula $\mathrm{SM}[FO(\mathcal{T}) \wedge FO(\mathcal{P});\ machine]$ entails $PERSON(DaveB)$. Furthermore, it entails each of $\exists y HAS\text{-}MOTHER(DaveB, y)$ and $ANIMAL(DaveB)$.

In fact, this treatment of a hybrid KB is essentially equivalent to the quantified equilibrium logic (QEL) based approach, as stated in Theorem 15 from [5]. The equivalence is also immediate from Lemma 9 from [7], which shows the equivalence between the first-order stable model semantics and QEL. de Bruijn *et al.* [5] show that a few other integration approaches, such as r-hybrid, r^+-hybrid, and g-hybrid KBs, can be embedded into QEL-based hybrid KBs. Consequently, they can also be represented by the first-order stable model semantics.

In the following we relate $\mathcal{DL} + log$ [3] and nonmonotonic dl-programs [1] to the first-order stable model semantics.

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

We refer the reader to [3] for the nonmonotonic semantics of $\mathcal{DL} + log$. A $\mathcal{DL} + log$ knowledge base is $(\mathcal{T}, \mathcal{P})$ where $\mathcal{T}$ is a DL knowledge base of signature $\langle C, P_{\mathcal{T}}\rangle$ and $\mathcal{P}$ is a (disjunctive) Datalog program of signature $\langle C, P_{\mathcal{T}} \cup P_{\mathcal{P}}\rangle$. $\mathcal{DL} + log$ imposes the *standard name assumption*: every interpretation is over the same fixed, countably infinite domain Δ, and in addition the set C of object constants is such that it is in the same one-to-one correspondence with Δ in every interpretation. As a result, for simplicity, we identify Δ with C.

In $\mathcal{DL} + log$, the predicates from $P_{\mathcal{T}}$ are not allowed to occur in the negative body of a rule in $\mathcal{P}$. In order to ensure decidable reasoning, $\mathcal{DL} + log$ imposes two conditions: *Datalog safety* and *weak safety*. The rules of $\mathcal{P}$ are called *Datalog safe* if every variable occurring in a rule also occurs in the positive body of the rule, and they are called *weakly safe* if every variable occurring in the head of a rule also occurs in a Datalog atom in the positive body of the rule.

The nonmonotonic semantics of $\mathcal{DL} + log$ is based on the stable model semantics for disjunctive logic programs. 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 1. *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 [3] are precisely the interpretations of $\langle C, P_{\mathcal{T}} \cup P_{\mathcal{P}}\rangle$ that satisfy* $\mathrm{SM}[FO(\mathcal{T}) \wedge FO(\mathcal{P});\ P_{\mathcal{P}}]$.

Since the reformulation does not refer to grounding, arguably, it provides a simpler account of $\mathcal{DL} + log$ in comparison with the original semantics in [3].

In view of the relationship between the two formalisms in Proposition 1, we observe that the condition of weak safety imposed in $\mathcal{DL} + log$ coincides with the condition of *semi-safety* from [8] that applies to $FO(\mathcal{T}) \wedge FO(\mathcal{P})$ when we take

$P_{\mathcal{P}}$ as intensional predicates[2]. Using the results on semi-safety presented in [8], below we show that the requirement of Datalog safety can be dropped without affecting the decidability of reasoning in $\mathcal{DL}+log$.

Proposition 2. *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 $\mathcal{P}$, Proposition 2 tells us that the decidability results (Theorems 11 and 12 from [3]) and the complexity results (Theorem 13 from [3]) with respect to the nonmonotonic semantics of $\mathcal{DL}+log$ 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.

4 Relating to Nonmonotonic dl-Programs by Eiter *et al.*

A nonmonotonic *dl-program* [1] 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 nondisjunctive rules that can contain queries to $\mathcal{T}$ in the form of "dl-atoms." A *dl-atom* is of the form

$$DL[S_1 op_1 p_1, \dots, S_m op_m p_m;\ Q](\mathbf{t}) \quad (m \geq 0) \tag{2}$$

where $S_i \in P_{\mathcal{T}}$, $p_i \in P_{\mathcal{P}}$, and $op_i \in \{\oplus, \odot, \ominus\}$; $Q(\mathbf{t})$ is a *dl-query* [1].

The semantics of dl-programs is defined by extending the answer set semantics to generalized programs. For 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 (2) *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 $\ominus$,

The satisfaction relation is extended to allow propositional connectives in the usual way.

Eiter *et al.* [1] define two semantics of dl-programs, which are based on different definitions of a reduct. In defining *weak* answer sets, the reduct is obtained from the given program by eliminating all dl-atoms (similar to the way that the

[2] The definition of semi-safety (called "argument-restricted" in that paper) is more general. That definition applies to any prenex formula even allowing function constants of positive arity.

negative literals in the body are eliminated in forming the reduct). In defining *strong* answer sets, the reduct is obtained from the given program by eliminating all nonmonotonic dl-atoms, but leaving monotonic dl-atoms. Below we show that each semantics can be characterized by our approach by extending F^* to handle dl-atoms in different ways.

For this, we define *dl-formulas* of 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 $\langle C, P_\mathcal{P}\rangle$[3]. Note that any generalized normal logic program can be viewed as a dl-formula: $FO(\mathcal{P})$ can be extended to a generalized normal logic program $\mathcal{P}$ in a straightforward way. Let F be a ground dl-formula[4]. We define F^{w*} the same as F^* except for a new clause for a dl-atom:

$$DL[S_1 op_1 p_1, \ldots, S_m op_m p_m; Q](\mathbf{c})^{w*}(\mathbf{u}) = DL[S_1 op_1 p_1, \ldots, S_m op_m p_m; Q](\mathbf{c}).$$

$\mathrm{SM}^w[F]$ is defined the same as formula (1) except that F^{w*} is used in place of F^*. The following proposition shows how weak answer sets can be characterized by this extension. The notion $FO(\mathcal{P})$ is straightforwardly extended to a generalized normal logic program by treating dl-atoms like standard atoms.

Proposition 3. *For any dl-program $(\mathcal{T}, \mathcal{P})$ such that $\mathcal{P}$ is ground, the weak answer sets of $(\mathcal{T}, \mathcal{P})$ are precisely the Herbrand interpretations of signature $\langle C, P_\mathcal{P}\rangle$ that satisfy $\mathrm{SM}^w[FO(\mathcal{P});\ P_\mathcal{P}]$ relative to $\mathcal{T}$.*

In order to capture strong answer sets, we define F^{s*} the same as F^* except for a new clause for a dl-atom:

$$DL[S_1 op_1 p_1, \ldots, S_m op_m p_m; Q](\mathbf{c})^{s*}(\mathbf{u}) = DL[S_1 op_1 u_1, \ldots, S_m op_m u_m; Q](\mathbf{c})$$

($u_1, \ldots, u_m$ are the elements of $\mathbf{u}$ that correspond to $p_1, \ldots, p_m$) if this dl-atom is monotonic; otherwise

$$DL[S_1 op_1 p_1, \ldots, S_m op_m p_m; Q](\mathbf{c})^{s*}(\mathbf{u}) = DL[S_1 op_1 p_1, \ldots, S_m op_m p_m; Q](\mathbf{c}).$$

$\mathrm{SM}^s[F]$ is defined the same as formula (1) except that F^{s*} is used in place of F^*. The following proposition shows how strong answer sets can be characterized by this extension.

Proposition 4. *For any dl-program $(\mathcal{T}, \mathcal{P})$ such that $\mathcal{P}$ is ground, the strong answer sets of $(\mathcal{T}, \mathcal{P})$ are precisely the Herbrand interpretations of signature $\langle C, P_\mathcal{P}\rangle$ that satisfy $\mathrm{SM}^s[FO(\mathcal{P});\ P_\mathcal{P}]$ relative to $\mathcal{T}$.*

The QEL based approach was extended to cover dl-programs in [10]. In that paper, the authors capture the weak (strong, respectively) semantics of dl-programs by defining weak (strong, respectively) QHT models of dl-atoms. The two variants of F^* above are syntactic counterparts of these definitions of QHT models.

[3] The extension is similar to the extension of first-order formulas to allow aggregate expressions as given in [9].

[4] We require F to be ground because strong answer set semantics distinguishes if a ground dl-atom is monotonic or nonmonotonic.

5 Conclusion

Since the first-order stable model semantics is a generalization of the traditional stable model semantics [11] to first-order formulas, it enables a rather simple and straightforward integration of logic programs and first-order logic KB. Recent work on the first-order stable model semantics helps us in studying the semantic properties and computational aspects of the hybrid KBs. For example, as discussed, the concept of semi-safety in 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 with $\mathcal{DL} + log$. Also, as discussed in [5], the notion of strong equivalence can be applied to provide the notion of equivalence between hybrid KBs.

Acknowledgements. We are grateful to David Pearce for pointers to earlier work and anonymous referees for their useful comments. The authors were partially supported by the National Science Foundation under Grants IIS-0916116 and by the IARPA SCIL program.

References

1. Eiter, T., Ianni, G., Lukasiewicz, T., Schindlauer, R., Tompits, H.: Combining answer set programming with description logics for the semantic web. Artificial Intelligence 172(12-13), 1495–1539 (2008)
2. Rosati, R.: On the decidability and complexity of integrating ontologies and rules. J. Web Sem. 3(1), 61–73 (2005)
3. Rosati, R.: DL+log: Tight integration of description logics and disjunctive datalog. In: Proceedings of International Conference on Principles of Knowledge Representation and Reasoning (KR), pp. 68–78 (2006)
4. Heymans, S., de Bruijn, J., Predoiu, L., Feier, C., Nieuwenborgh, D.V.: Guarded hybrid knowledge bases. TPLP 8(3), 411–429 (2008)
5. de Bruijn, J., Pearce, D., Polleres, A., Valverde, A.: A semantical framework for hybrid knowledge bases. Knowl. Inf. Syst. 25(1), 81–104 (2010)
6. Motik, B., Rosati, R.: Reconciling description logics and rules. J. ACM 57(5) (2010)
7. Ferraris, P., Lee, J., Lifschitz, V.: Stable models and circumscription. Artificial Intelligence 175, 236–263 (2011)
8. Bartholomew, M., Lee, J.: 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), pp. 477–485 (2010)
9. Lee, J., Meng, Y.: On reductive semantics of aggregates in answer set programming. In: Erdem, E., Lin, F., Schaub, T. (eds.) LPNMR 2009. LNCS, vol. 5753, pp. 182–195. Springer, Heidelberg (2009)
10. Fink, M., Pearce, D.: A logical semantics for description logic programs. In: Janhunen, T., Niemelä, I. (eds.) JELIA 2010. LNCS, vol. 6341, pp. 156–168. Springer, Heidelberg (2010)
11. Gelfond, M., Lifschitz, V.: The stable model semantics for logic programming. In: Kowalski, R., Bowen, K. (eds.) Proceedings of International Logic Programming Conference and Symposium, pp. 1070–1080. MIT Press, Cambridge (1988)