On Representational Issues About Combinations of Classical Theories with Nonmonotonic Rules*

Jos de Bruijn[1], Thomas Eiter[2], Axel Polleres[1,3], and Hans Tompits[2]

[1] Digital Enterprise Research Institute (DERI), Leopold-Franzens Universität Innsbruck, Technikerstraße 21a, A-6020 Innsbruck, Austria
jos.debruijn@deri.org
[2] Institut für Informationssysteme 184/3, Technische Universität Wien, Favoritenstrasse 9-11, A-1040 Vienna, Austria
{eiter, tompits}@kr.tuwien.ac.at
[3] Universidad Rey Juan Carlos, Campus de Mostoles, DI-236, Calle Tulipan s/n, E-28933 Madrid, Spain
axel.polleres@urjc.es

Abstract. In the context of current efforts around Semantic-Web languages, the combination of classical theories in classical first-order logic (and in particular of ontologies in various description logics) with rule languages rooted in logic programming is receiving considerable attention. Existing approaches such as SWRL, dl-programs, and $\mathcal{DL}$+*log*, differ significantly in the way ontologies interact with (nonmonotonic) rules bases. In this paper, we identify fundamental representational issues which need to be addressed by such combinations and formulate a number of formal principles which help to characterize and classify existing and possible future approaches to the combination of rules and classical theories. We use the formal principles to explicate the underlying assumptions of current approaches. Finally, we propose a number of settings, based on our analysis of the representational issues and the fundamental principles underlying current approaches.

1 Introduction

The question of combining different knowledge-representation formalisms is recently gaining increasing interest in the context of the Semantic-Web initiative. While the W3C recommendation of the OWL Web ontology language [1] has been around for over two years, attention is now shifting towards defining a rule language for the Semantic Web which integrates with OWL. From a formal point of view, OWL (DL) can be seen as a syntactic variant of an expressive description logic [2], viz. $\mathcal{SHOIN}(\mathbf{D})$ [3], which is a decidable subset of classical first-order logic. In this sense, OWL follows the

* The first author was partially supported by the European Commission under projects Knowledge Web (IST-2004-507482), DIP (FP6-507483), and SEKT (IST-2003-506826), as well as by the Wolfgang Pauli Institute, Vienna. The second and the fourth author were partially supported by the Austrian Science Fund (FWF) under project P17212 and by the European Commission under project REWERSE (IST-2003-506779). The third author was partially supported by the CICyT project TIC-2003-9001-C02.

J. Lang, F. Lin, and J. Wang (Eds.): KSEM 2006, LNAI 4092, pp. 1–22, 2006.

tradition of earlier classical ontology languages such as KIF [4] or, more recently, the ISO Common Logic [5] effort.[1]

Declarative rule languages, on the contrary, are usually based on logic-programming methods, adopting a non-classical semantics via minimal Herbrand models. Additionally, such languages often include extensions with nonmonotonic negation [6,7]. The main differences between classical logic and rule-based languages are assumptions concerning an open vs. a closed domain and non-uniqueness vs. uniqueness of names. Combinations of ontologies, or, more generally, first-order (FO) theories, and rule bases need to take these differences into account.

There have recently been several proposals for integrating such classical ontologies (FO theories) and rule bases (e.g., [8,9,10,11,12]). Each of these approaches overcomes the differences between the paradigms in a different way, often without making the underlying assumptions of the semantics of the combination explicit.

In this paper, we study general representational issues when dealing with a combination of classical theories and rule-based languages. In particular, we specify a number of formal principles such a combination must obey, taking the fundamental differences between the classical semantics and the semantics of rule-based languages into account, as well as the different kinds of interaction between them. Furthermore, we propose a number of generic settings for such a combination, which help clarify and classify possible approaches. As formal languages underlying the classical component (ontology) and the rules component of a combined knowledge base we consider here classical first-order logic with equality and disjunctive logic programs under the stable-model semantics [7,13], respectively.

We stress that we do not consider *extensions* of a classical formalism with nonmonotonic features such as default logic [14], autoepistemic logic [15], or circumscription [16,17], but start our observations based on existing approaches which *combine standard semantics* for the ontology and rules components.

2 Preliminaries

We start with a brief review of the basic elements of classical first-order logic with equality and disjunctive logic programs under the stable-model semantics. As we will see in the next section, both formalisms generalize those considered in the major approaches to combining rules and ontologies.

2.1 First-Order Logic

A first-order language $\mathcal{L}$ consists of all formulas over a signature $\Sigma = (\mathcal{F}, \mathcal{P})$, where $\mathcal{F}$ and $\mathcal{P}$ are countable sets of *function* and *predicate symbols*, respectively, and a countably infinite set $\mathcal{V}$ of *variable symbols*. Each $f \in \mathcal{F}$ and each $p \in \mathcal{P}$ has an associated *arity* $n \geq 0$; 0-ary function symbols are also called *constants*. *Terms* of $\mathcal{L}$ are either constants, variables, or constructed terms of form $f(t_1, .., t_n)$, where f is an n-ary function symbol and $t_1, ..., t_n$ are terms. An *atomic formula* is either a predicate $p(t_1, ..., t_n)$,

[1] Although Common Logic is syntactically of higher-order type, most part of it is actually first-order.

with p being an n-ary predicate symbol, or $t_1 = t_2$, where $t_1, ..., t_n$ are terms in $\mathcal{L}$. Variable-free terms (or atomic formulas) are called *ground*. A ground term is also referred to as a *name*.

Complex formulas are constructed in the usual way using the connectives $\neg$, $\wedge$, $\vee$, and $\supset$, the quantifiers $\exists$ and $\forall$ and the auxiliary symbols "(" and ")." A variable occurrence is called *free* if it does not occur in the scope of a quantifier. A formula is *open* if it has free variables, *closed* otherwise. Closed formulas are also called *sentences* of $\mathcal{L}$. By $\forall\phi$ and $\exists\phi$ we denote the universal and existential closure of a formula ϕ, respectively.

An *interpretation* of a language $\mathcal{L}$ is a tuple $\mathcal{I} = \langle U, \cdot^I \rangle$, where U is a nonempty set (called *domain*) and $\cdot^I$ is a mapping which assigns a function $f^I : U^n \to U$ to every n-ary function symbol $f \in \mathcal{F}$ and a relation $p^I \subseteq U^n$ to every n-ary predicate symbol $p \in \mathcal{P}$.

A *variable assignment* B for an interpretation $\mathcal{I}$ is a mapping which assigns an element $x^B \in U$ to every variable $x \in \mathcal{V}$. A variable assignment B' is an x-*variant* of B if $y^B = y^{B'}$ for every variable $y \in \mathcal{V}$ such that $y \neq x$. A *variable substitution* β is a set of form $\{x_1/t_1, ..., x_k/t_k\}$, where $x_1, ..., x_k \in \mathcal{V}$ are distinct variables and $t_1, ..., t_k$ are names of $\mathcal{L}$. A variable substition is *total* if it contains x/n for every variable $x \in \mathcal{V}$.[2] Given a variable assignment B and substitution β, if $\beta = \{x/t \mid x \in \mathcal{V}, t^{\mathcal{I}} = x^B$, for some name $t\}$, then β is *associated with* B.

The *application* of a variable substitution β to some term, formula, or theory is defined as follows: for a variable x, $x\beta = t$, if β contains some x/t, and $x\beta = x$ otherwise; for a formula $\phi(x_1, ..., x_n)$, where $x_1, ..., x_n$ are the free variables of ϕ, $\phi(x_1, ..., x_n)\beta = \phi(x_1\beta, ..., x_n\beta)$; for a set $\Phi = \{\phi_1, ..., \phi_n\}$ of formulas, $\Phi\beta = \{\phi_1\beta, ..., \phi_n\beta\}$.

Note that each assignment may have, depending on the interpretation, several associated variable substitutions.

Example 1. Consider a language $\mathcal{L}$ with constants $\mathcal{F} = \{a, b, c\}$, and an interpretation $\mathcal{I} = \langle U, \cdot^I \rangle$ with $U = \{k, l, m\}$ and such that $a^I = k$, $b^I = l$, and $c^I = l$. The variable assignment B is defined as follows: $x^B = k$, $y^B = l$, and $z^B = m$. B has two associated variable substitutions, $\beta_1 = \{x/a, y/b\}$ and $\beta_2 = \{x/a, y/c\}$, but no total associated variable substitution since m is an unnamed individual. $\Box$

Given an interpretation $\mathcal{I} = \langle U, \cdot^I \rangle$, a variable assignment B, and a term t of $\mathcal{L}$, $t^{I,B}$ is defined as follows: $x^{I,B} = x^B$, for a variable x, and $t^{I,B} = f^I(t_1^{I,B}, ..., t_n^{I,B})$, for $t = f(t_1, ..., t_n)$. An individual $k \in U$ which is represented by at least one name t in the language, i.e., such that $t^I = k$, is called a *named* individual, otherwise *unnamed*.

An interpretation $\mathcal{I} = \langle U, \cdot^I \rangle$ *satisfies* an atomic formula $p(t_1, ..., t_n)$ *relative* to a variable assignment B, denoted $\mathcal{I}, B \models p(t_1, ..., t_n)$, if $(t_1^{I,B}, ..., t_n^{I,B}) \in p^I$. Furthermore, $\mathcal{I}, B \models t_1 = t_2$ iff $t_1^{I,B} = t_2^{I,B}$. This is extended to arbitrary formulas as usual. In particular, we have that $\mathcal{I}, B \models \forall x\phi_1$ (resp., $\mathcal{I}, B \models \exists x\phi_1$) iff for every (resp., for some) B' which is an x-variant of $B, \mathcal{I}, B' \models \phi_1$ holds.

An interpretation $\mathcal{I}$ is a *model* of ϕ, denoted $\mathcal{I} \models \phi$, if $\mathcal{I}, B \models \phi$, for every variable assignment B. This definition is straighforwardly extended to the case of first-order

[2] Note that our notion of a variable substitution is slightly different from the usual one, since we only allow substitution of variables with *names* rather than with arbitrary terms.

theories. Given a theory Φ and a formula ϕ over $\mathcal{L}$, Φ *entails* ϕ, denoted $\Phi \models \phi$, iff, for all interpretations $\mathcal{I}$ in $\mathcal{L}$ such that $\mathcal{I} \models \Phi$, $\mathcal{I} \models \phi$ holds.

2.2 Logic Programs

A *disjunctive logic program* P consists of rules of form

$$h_1 \mid \ldots \mid h_l \leftarrow b_1, \ldots, b_m, \textit{not}\ b_{m+1}, \ldots \textit{not}\ b_n,$$

where $h_1, \ldots, h_l, b_1, \ldots, b_n$ are atomic formulas. $H(r) = \{h_1, ..., h_l\}$ is the set of *head atoms* of r, $B^+(r) = \{b_1, ..., b_m\}$ is the set of *positive body atoms* of r, and $B^-(r) = \{b_{m+1}, ..., b_n\}$ is the set of *negative body atoms* of r. If $l = 1$, then r is a *normal rule*. If every rule in $r \in P$ is normal, then P is normal. If $B^-(r) = \emptyset$, then r is *positive*. If every rule $r \in P$ is positive, then P is positive.

Let Σ_P denote a first-order signature which is a superset of the function, predicate, and variable symbols which occur in P and let $\mathcal{L}_P$ denote the first-order language based on Σ_P. The *Herbrand universe* U_H of $\mathcal{L}_P$ is the set of all ground terms over Σ_P. The *Herbrand base* B_H of $\mathcal{L}_P$ is the set of all atomic formulas which can be formed using the predicate symbols of Σ_P and the terms in U_H. A *Herbrand interpretation* M is a subset of B_H. With a little abuse of notation, we can view M equivalently as a first-order interpretation $\langle U_H, \cdot^I \rangle$, where $\cdot^I$ is such that $\langle t_1, ..., t_n \rangle \in p^I$ iff $p(t_1, ..., t_n) \in M$, for an n-ary predicate symbol p and ground terms $t_1, ..., t_n$. Depending on the context, we view M either as a set of atoms of $\mathcal{L}_P$ or as a first-order interpretation of $\mathcal{L}_P$.

The *grounding* of a logic program P, denoted $gr(P)$, is the union of all possible ground instantiations of P, obtained by replacing each variable in r with a term in U_H, for each rule $r \in P$.

Let P be a positive logic program. A Herbrand interpretation M of P is a *model* of P if, for every rule $r \in gr(P)$, $B^+(r) \subseteq M$ implies $H(r) \cap M \neq \emptyset$. A Herbrand model M of a logic program P is *minimal* iff for every model M' such that $M' \subseteq M$, $M' = M$. Every positive normal logic program has a single minimal Herbrand model, which is the intersection of all Herbrand models.

Following Gelfond and Lifschitz [7], the *reduct* of a logic program P with respect to an interpretation M, denoted P^M, is obtained from $gr(P)$ by deleting (i) each rule with a literal $\textit{not}\ b$ in its body with $b \in M$, and (ii) all negative body literals in the remaining rules. If M is a minimal Herbrand model of the reduct P^M, then M is a *stable model* of P.

Example 2. Consider the following program P:

$$p(a); \quad p(b); \quad q(X) \mid r(X) \leftarrow p(X), \textit{not}\ s(X),$$

together with the interpretation $M_1 = \{p(a), p(b), q(a), r(a)\}$. The reduct $P^{M_1} = \{p(a); p(b); q(a) \mid r(a) \leftarrow p(a), \textit{not}\ s(a); q(b) \mid r(b) \leftarrow p(b), \textit{not}\ s(b)\}$ has the minimal model M_1, thus M_1 is a stable model of P. The other stable models of P are $M_2 = \{p(a), p(b), q(a), r(b)\}$, $M_3 = \{p(a), p(b), q(b), r(a)\}$, and $M_4 = \{p(a), p(b), q(b), r(b)\}$. $\square$

A disjunctive logic program P is *consistent* if it has a stable model. Furthermore, P *cautiously entails* a ground atomic formula α if $\alpha \in M$ for every stable model M of P. As well, P *bravely entails* a ground atomic formula α if $\alpha \in M$ for some stable model M of P.

The stable-model semantics [7], also referred to as the *answer-set semantics*, coincides with the minimal Herbrand-model semantics [18] for positive programs, with the perfect-model semantics [19], the well-founded semantics [6] for locally stratified programs, and with the well-founded semantics in case the well-founded model is total [7,6].

3 Current Approaches for Combining Knowledge Bases

We are concerned in this paper with knowledge bases which combine classical first-order logic and rules. A combined knowledge base $\mathcal{KB} = \langle \Phi, P \rangle$ consists of

- a first-order theory (the *classical component*) Φ, which is a set of formulas in some first-order language $\mathcal{L}_\Phi$ with signature Σ_Φ, and
- a disjunctive logic program (the *rules component*) P with signature Σ_P.

The combined signature of $\mathcal{KB}$, denoted $\Sigma_{\mathcal{KB}}$, is the union of Σ_Φ and Σ_P.

Several kinds of interactions between FO theories (or ontologies) and rules require a separation between predicates "belonging to" the FO theory component and predicates "belonging to" the rules component. We refer to predicate symbols in Σ_Φ as *classical predicates* and predicates in Σ_P as *rules predicates*. Unless mentioned otherwise, the sets of classical and rules predicates are assumed to be disjoint. *Classical atoms* are atomic formulas with a classical predicate and *rules atoms* are atomic formulas with a rules predicate. All of the approaches mentioned in this paper allow classical predicates to occur in logic programs, but do not allow rules predicates to occur in the FO theory.

In the remainder of this section we give a short survey of the most prominent approaches to combining FO theories and rules.

SWRL and Subsets. SWRL [20] is an extension of OWL DL, which corresponds to the description logic $\mathcal{SHOIN}(\mathbf{D})$, with function-free Horn-like rules.[3] SWRL allows conjunctions of atomic concepts and roles (unary and binary predicates), as well complex concept descriptions in the heads and bodies of rules. We assume here that rules in a SWRL knowledge base are positive Horn formulas. This is no real limitation, since complex concept descriptions may be replaced with new concepts which are defined equivalently to the complex descriptions in the FO theory, and rules with a conjunction of atoms in the head may be split into several rules.

A SWRL knowledge base $\mathcal{KB} = \langle \Phi, P \rangle$ can be seen as consisting of an FO theory Φ (a $\mathcal{SHOIN}(\mathbf{D})$ ontology), and a rules component P, which in turn consists of a set of positive, normal rules where atoms may be either unary, binary or (in)equality predicates. An interpretation $\mathcal{I}$ satisfies $\mathcal{KB}$ iff $\mathcal{I} \models \Phi \cup P$, where $\models$ is the classical first-order satisfaction relation. The ontology and the rules are thus interpreted as a single first-order theory.

[3] SWRL allows classical negation through the OWL DL axioms, but not in rules.

Notice that SWRL does not distinguish between description logic (DL) predicates and rule predicates. There is full interaction between the DL component and the rules component. As was shown in the seminal work about CARIN [21], an unlimited interaction between Horn rules and DLs leads to undecidability of key inference tasks, which also holds for the restricted form of rules allowed in SWRL. In order to recover decidability, one could either reduce the expressiveness of the DL or of the rules component (cf. [22] for a short survey on a number of restrictions which recover decidability; these restrictions reach from only allowing the expressive intersection of DLs and Horn rules [23] to leaving full syntactic freedom for the DL, but restricting Horn rules to so-called *DL-safe rules* [12] or tree-shaped rules [24]).

A drawback of SWRL from a representational point-of-view is that it does not allow the integration of nonmonotonic logic programs with ontologies. The approaches mentioned in the remainder of this section do allow the consideration of nonmonotonic rules in a combined knowledge base.

$\mathcal{DL}$+log and Its Predecessors. $\mathcal{AL}$-*log* [25] is an approach to integrating the description logic $\mathcal{ALC}$ with positive (non-disjunctive) datalog. This approach was extended to the case of disjunctive datalog with negation under the stable-model semantics in [26] and further generalized to the case of arbitrary classical ontology languages in [8]. The latest successor in this chain is $\mathcal{DL}$+*log*, which allows a tighter integration of rules and ontologies than the earlier approaches. In this short survey, we will restrict ourselves to $\mathcal{DL}$+*log*.

The integration of rules and ontologies in a $\mathcal{DL}$+*log* knowledge base $\mathcal{KB} = \langle \Phi, P \rangle$ roughly works as follows. The classical predicates are interpreted in a classical interpretation $\mathcal{I}$. The reduct of the program P with respect to $\mathcal{I}$ "evaluates" all classical atoms according to their truth value in $\mathcal{I}$. The resulting program, denoted $P_{\mathcal{I}}$, does not contain any classical predicates. This program is evaluated using the stable-model semantics as usual. For each model of the classical component, there may be zero, one, or multiple stable models M of the rules component. Models of the combined knowledge base $\mathcal{KB}$ are then of the form $\mathcal{I} \cup M$ for each model $\mathcal{I}$ of Φ and stable model M. One consequence of this definition is that if there is no stable model M for $\mathcal{I}$, then there is no combined model $\mathcal{I} \cup M$. In this way, the logic program can restrict the set of classical models, which is a form of interaction from the rules to the FO theory.

A ground atom is a consequence of the combined knowledge base iff it is true in every combined model.

In order to use the standard definitions of stable models, $\mathcal{DL}$+*log* imposes the standard-names assumption, which assumes a one-to-one correspondence between names in the language and individuals in the domain of each interpretation. Another restriction is that classical predicates are not allowed to occur negatively in rule bodies. Furthermore, $\mathcal{DL}$+*log* defines the weak DL-safeness restriction on variables in rules in order to retain decidability of reasoning. Each variable which occurs in the head of a rule must occur in a positive rules atom in the body. This ensures that only conclusions are drawn about individuals in the Herbrand universe. The "weak" in "weak safeness" refers to the fact that there may be variables in classical atoms in the body of a rule which do not occur in any atom in the head. This allows to express conjunctive queries over a DL knowledge base in the body of a rule, while still keeping the combined formalism decidable.

As for the various variants of safeness restrictions mentioned so far, one may argue that these restrictions are really limiting, because variables can to a large extent only range over constants which occur in the rules component. However, it is often argued that one could easily add a predicate to the rules component and add a fact $O(a)$ for each constant a which occurs in the classical component. One could then add $O(x)$ to the body of each rule for each unsafe variable x, as proposed for instance in [12].

dl-Programs. In contrast to the $\mathcal{DL}$+*log* approach, the rules in a dl-program [10] do not interact with the FO theory based on single models, but rather using a clean interface which allows the exchange of ground atoms. This approach relies also on the stable-models semantics, but there is a more strict separation between the classical component and the rules component.

The interaction between the classical component and the rules component is through special query predicates in the bodies of rules, called *dl-atoms*. Allowed queries are *concept membership*, *role membership*, and *concept inclusion*. The approach allows a bidirectional flow of information: dl-atoms allow to "extend" the extensions of unary and binary rules predicates in the DL knowledge base, to be taken into account for the query to be answered.

As is the case for $\mathcal{DL}$+*log*, dl-programs distinguish between classical predicates and rules predicates; in dl-programs, the distinction between DL predicates and rules predicates is made implicitly—the only places where classical predicates occur in rules are the dl-atoms.

The semantics of dl-programs is defined with respect to ground logic programs. However, unlike for usual logic programs, the grounding of dl-programs is not computed with respect to the Herbrand universe of the logic program, but with respect to some arbitrary signature Σ, which might be the combined signature of the classical component and the rule component. The extended Herbrand base of a dl-program consists of all the atoms which can be constructed using the predicate and constant symbols in the signature Σ. An interpretation M is a subset of the extended Herbrand base. A ground dl-atom can be viewed as a set S^M of facts together with a ground query $Q(\mathbf{c})$, where Q is a (possibly negated) unary or binary predicate and $\mathbf{c}$ is a constant or a binary tuple of constants, respectively. A dl-atom is true in M with respect to a FO theory Φ iff

$$\Phi \cup S^M \models Q(\mathbf{c}).$$

Truth of regular atoms in the program is determined in the usual way, i.e., a ground atom α is true in M iff $\alpha \in M$. DL atoms can be removed from the ground program based on their truth value in M with respect to Φ: rules with a dl-atom in the body which is false in M with respect to Φ are removed from the program and the dl-atoms in the bodies of the remaining rules are removed. The stable-model semantics for the resulting normal program is then defined as usual.

4 Representational Issues of Combined Knowledge Bases

As we have seen in the previous section, the semantics of a combined knowledge base is defined differently for the different approaches. It is not immediately clear from the definitions what the implications are of using a particular semantics and what the expected behavior is of the combination.

When defining such a semantics of a combined knowledge base $\mathcal{KB}$, different representational issues arise which have to be dealt with. These issues stem from the different underlying assumptions in the formalisms such as open vs. closed-world assumption and unique vs. non-unique names assumption. Our main concerns are (i) the form of the domain of discourse for the quantification of the variables in the logic-program rules, (ii) implications of the unique-names assumption in the logic program, (iii) the notion of interaction from the theory to the logic program, and (iv) the notion of interaction from the rules to the theory. Each approach to combining rules and FO theory makes, either implicitly or explicitly, particular choices to deal with these issues in the definition of its semantics. In this section, we make these choices explicit by defining a number of formal principles which may underlie the semantics of a combined knowledge base.

4.1 Domain of Discourse

The semantics of logic programs is usually defined with respect to a fixed domain, viz. the Herbrand universe. An important property which holds for interpretations based on the Herbrand universe is *domain closure* [27], which means that the domain of each interpretation is limited to the Herbrand universe. In a combined knowledge base, one may want to take individuals outside of this fixed domain into account. This would require taking a larger domain of the models of P into account.

A straightforward approach is to simply use the Herbrand universe of $\mathcal{L}_P$. A drawback of this approach is that the only statements derived from Φ which are taken into account in P are the statements which involve names in the Herbrand universe. Consider the first-order theory $\Phi = \{p(a)\}$ and the logic program $P = \{r(b),\ q(x) \leftarrow p(x)\}$, where a is not in Σ_P. In case the variable in P quantifies only over the Herbrand universe U_H of $\mathcal{L}_P$, $q(a)$ cannot be concluded, since a is not in U_H.

An extension of this approach, which allows to consider also the names in Φ, is to consider an extended Herbrand universe, where the extended Herbrand universe consists of all names (i.e., ground terms) of the combined signature $\Sigma_{\mathcal{KB}}$. In this case, statements in Φ involving names which are not in the Herbrand universe of $\mathcal{L}_P$ are also taken into account. When considering an extended Herbrand universe as the domain of discourse, $q(a)$ could be concluded in the previous example. The potential drawback which remains with this approach is that unnamed individuals are not considered, as is demonstrated in the following example. The drawback can be overcome, however, by allowing *arbitrary domains* as the domain of discourse for P.

Example 3. Consider $P = \{q \leftarrow p(x)\}$ and $\Phi = \{\exists x p(x)\}$. If the domain of discourse of P is an extended Herbrand base, q can not be concluded, because there is no name t such that $p(t)$ can be concluded. □

We will now formally define a number of principles concerning the *domain of discourse* of the rules component of a combined knowledge base.

Principle 1.1 (Herbrand universe). *Given a combined knowledge base $\mathcal{KB} = \langle \Phi, P \rangle$, each interpretation M of $\mathcal{L}_P$, viewed as a pair $\langle U, \cdot^I \rangle$, has the same fixed universe*

$U = U_H$, where U_H is the Herbrand universe of $\mathcal{L}_P$. Furthermore, the interpretation function $\cdot^I$ is such that each ground term t over Σ_P is interpreted as itself, i.e., such that $t^I = t$.

Principle 1.2 (Combined signature). *Given a combined knowledge base $\mathcal{KB}=\langle\Phi, P\rangle$, each interpretation M of $\mathcal{L}_P$, viewed as a pair $\langle U, \cdot^I\rangle$, has the same fixed universe $U = U_{\mathcal{KB}}$, where $U_{\mathcal{KB}}$ is the set of ground terms of the combined signature $\Sigma_{\mathcal{KB}}$. Furthermore, the interpretation function $\cdot^I$ is such that each ground term t of $\Sigma_{\mathcal{KB}}$ is interpreted as itself, i.e., such that $t^I = t$.*

Principle 1.3 (Arbitrary domain). *Given a combined knowledge base $\mathcal{KB} = \langle\Phi, P\rangle$, each interpretation M of $\mathcal{L}_P$, viewed as a pair $\langle U, \cdot^I\rangle$, has an arbitrary first-order domain U and there are no restrictions on the interpretation function $\cdot^I$.*

Notice that Principles 1.1 and 1.2 coincide in case the names of the signatures Σ_P and $\Sigma_{\mathcal{KB}}$ coincide. The principles can be forced to coincide by extending Σ_P to include all ground terms of $\Sigma_{\mathcal{KB}}$ (see e.g. [12]); note that this may lead to an infinite logic program in case the signature is infinite.

Providing the standard-names assumption applies to the combined knowledge base, Principles 1.2 and 1.3 coincide, since then there is a one-to-one correspondence of names in the language and individuals in the domain.

4.2 Uniqueness of Names

Herbrand interpretations satisfy the unique-names assumption, i.e., for any two distinct ground terms in the Herbrand universe, their interpretations are distinct as well. There are, however, approaches which adopt a less restrictive view by axiomatizing a special equality predicate [27]. In such a case, there is a notion of default inequality: two ground terms are assumed to be unequal, unless equality between the terms can be derived.

The unique-names assumption does not hold in general for first-order interpretations. Several names in the language may be interpreted as the same individual in the domain (see, e.g., Example 1). Therefore, one may want to adopt a less restrictive view on uniqueness of names in the rules component of a combined knowledge base. We distinguish between maintaining the unique-names assumption, axiomatizing a special equality predicate, and discarding the unique-names assumption:

Principle 2.1 (Uniqueness of names). *Given a combined knowledge base $\langle\Phi, P\rangle$, for every interpretation $\langle U, \cdot^I\rangle$ of $\mathcal{L}_P$ and every pair of distinct names t_1, t_2 of $\mathcal{L}_P$, $t_1^I \neq t_2^I$ holds.*

Principle 2.2 (Special equality predicate). *Given a combined knowledge base $\langle\Phi,P\rangle$, a special binary equality predicate eq (cf. [27]) is axiomatized as part of P.*

Principle 2.3 (No uniqueness of names). *The unique-names assumption does not apply.*

Notice that Principles 1.1 and 1.2 enforce the unique-names assumption in the rules component; they cannot be combined with Principle 2.3. Notice further that in case a special equality predicate is axiomatized in P, it is generally desirable that if equality between two individuals is derived from Φ, this information is also available in P. As proposed in [28], the predicate eq may be defined in terms of equality $=$ in the classical component.

4.3 Interaction from First-Order Theories to Rules

Interaction between a first-order theory and a set of rules can take place in two directions: (a) from the FO theory to the rules and (b) from the rules to the FO theory. In this section, we consider the interaction from the FO theory to the rules; we discuss interaction from the rules to the FO theory in the next section.

We extend the notion of a logic program to distinguish between the uses of classical predicates and rules predicates. A logic program with classical atoms P consists of a set of rules of form

$$h_1 \mid ... \mid h_o \leftarrow a_1, ..., a_m, not\ b_1, ..., not\ b_n, c_1, ..., c_l, not\ d_1, ..., not\ d_k, \qquad (1)$$

where a_i, b_j are rules atoms and $c_{i'}$, $d_{j'}$ are classical atoms; $c_1, \ldots, d_k$ is called the *classical component* of the body of the rule, denoted $CB(r)$, and $a_1, \ldots, not\ b_n$ is called the *rules component*, denoted $RB(r)$. We moreover define the sets $CB^+(r) = \{c_1, \ldots, c_l\}$, $CB^-(r) = \{d_1, \ldots, d_k\}$, $RB^+(r) = \{a_1, \ldots, a_m\}$, and $RB^-(r) = \{b_1, \ldots, d_n\}$.

By interaction from the FO theory to the rules we mean the conditions under which the classical atoms in the body of a rule are true or false. We distinguish two basic principles a combined knowledge base may obey with respect to the interaction from FO theories to rules: interaction based on *single models* and interaction based on *entailment*. In the former case, the truth of $CB(r)$ corresponds to satisfaction in a single model $\mathcal{I}$ of the classical component Φ; in the latter case, the truth of $CB^+(r)$ and $CB^-(r)$ is determined by entailment or non-entailment from Φ, respectively. These notions of interaction are generalizations of the notions of interaction as defined in $\mathcal{DL}$+*log* [9] and dl-programs [10], respectively, as we shall see in the next section.

We now define the principles formally:

Principle 3.1 (Interaction based on single models). *Let $\mathcal{KB} = \langle \Phi, P \rangle$ be a combined knowledge base such that $\Phi \subseteq \mathcal{L}$, $\mathcal{I}$ an interpretation of $\mathcal{L}$, and B a variable assignment.*

The classical component of the body of a rule $r \in P$ is true *in $\mathcal{I}$ with respect to B, denoted $\mathcal{I}, B \models CB(r)$, iff $\mathcal{I}, B \models CB^+(r)$ and $\mathcal{I}, B \not\models CB^-(r)$.*

An interpretation M s-satisfies *a rule r with respect to $\mathcal{I}$ and B, denoted $M, B \models_{\mathcal{I}} r$, iff $M, B \models RB(r)$ and $\mathcal{I}, B \models CB(r)$ only if $M, B \models H(r)$.*

We call M an *s-model* of r with respect to $\mathcal{I}$ iff $M, B \models_{\mathcal{I}} r$, for every variable assignment B. Furthermore, M is an s-model of P with respect to $\mathcal{I}$ iff $M \models_{\mathcal{I}} r$, for every rule $r \in P$.

Principle 3.2 (Interaction based on entailment). *Let $\mathcal{KB} = \langle \Phi, P \rangle$ be a combined knowledge base such that $\Phi \subseteq \mathcal{L}$.*

The classical component of the body of a rule $r \in P$ *is* entailed by Φ *with respect to a variable substitution* β*, denoted* $\Phi \models CB(r)\beta$*, iff* $\Phi \models CB^{+}(r)\beta$ *and* $\Phi \not\models CB^{-}(r)\beta$*.*

An interpretation M e-satisfies *a rule* r *with respect to a variable assignment* B *and* Φ*, denoted* $M, B \models_{\Phi} r$*, iff, for some variable substitution* β *associated with* B*,* $M, B \models RB(r)$ *and* $\Phi \models CB(r)\beta$ *only if* $M, B \models H(r)$*.*

M is an *e-model* of r with respect to Φ iff $M, B \models_{\Phi} r$, for every variable assignment B. Furthermore, M is an e-model of P with respect to Φ iff $M \models_{\Phi} r$, for every rule $r \in P$.

Note that in case P is a ground program, the variable assignments and substitutions can be disregarded in the definitions of the principles.

Providing the combined knowledge base obeys Principle 1.1 or Principle 1.2, the variable assignment B is equivalent to its associated variable substitution β: $M, B \models \alpha$ iff $M \models \alpha\beta$, with $x/t \in \beta$ iff $x^{B} = t$, and the logic program P is actually equivalent to its ground instantiation with respect to U_H or the ground terms of $\Sigma_{\mathcal{KB}}$, respectively. Thus, the only case where the variable assignment is crucial in the definitions is when variables in the rule may quantify over arbitrary domains, i.e., when $\mathcal{KB}$ obeys Principle 1.3.

Stable Models for Logic Programs in Combined Knowledge Bases. In order to capture the nonmonotonic aspects of the rules components, we need to define which models are actually the intended models of P. We do this by extending the notion of stable models [7] to the case of logic programs in combined knowledge bases. For the definition of stable models, we assume the domain of discourse in an (extended) Herbrand universe (Principle 1.1 or 1.2). We first need to define the ground instantiation of P.

We augment the definition of $gr(P)$ to obtain $gr_{y}^{\mathcal{KB}}(P)$ as follows, where y is either H (in case of Principle 1.1) or $\mathcal{KB}$ (in case of Principle 1.2): $gr_{y}^{\mathcal{KB}}(P)$ is the union of all possible ground instantiations of r which are obtained by replacing each variable which occurs in a rules predicate by a term in U_y, for each rule $r \in P$.

We can now define the notion of a stable model for the logic program P in a combined knowledge base $\mathcal{KB} = \langle \Phi, P \rangle$ in view of Principle 3.1 (resp., Principle 3.2): Let M be an s-model (resp., e-model) of P with respect to $\mathcal{I}$ (resp., Φ), the *reduct* of P with respect to M, denoted $P_{\mathcal{I}}^{M}$ (resp., P_{Φ}^{M}) is obtained from $gr^{\mathcal{KB}}y(P)$ by removing

- every rule r such that $\mathcal{I} \not\models \exists CB(r)$ (resp., $\Phi \not\models \exists CB(r)$),
- the classical component from every remaining rule,
- every rule r such that $B^{-}(r) \cap M \neq \emptyset$, and
- the negative body literals from the remaining rules.

Then, M is a *stable s-model* (resp., *stable e-model*) *of* P *with respect to* $\mathcal{I}$ (resp., Φ) iff M restricted to rules predicates is a minimal Herbrand model of $P_{\mathcal{I}}^{M}$ (resp., P_{Φ}^{M}).

The following example shows that there is a difference between the two principles already in simple cases.

Example 4. Consider the combined knowledge base $\mathcal{KB} = \langle \Phi, P \rangle$ with $\Phi = \{p \vee q\}$ and $P = \{r \leftarrow p,\ r \leftarrow q\}$. Note that Φ entails neither p nor q. For the case of interaction based on single models of Φ, r is included in each of the (stable) models of

P with respect to every model of Φ, since we know that for each model of Φ, either p or q (or both) is true. In case the interaction is based on entailment, r is not included in the single stable e-model of P with respect to Φ, because neither p nor q is entailed by Φ. □

In the case of interaction based on single models, classical predicates are always interpreted classically,[4] and it is not possible to use "real" nonmonotonic negation over classical predicates or rules predicates which depend on them.

Example 5. Given the classical theory $\Phi = \{p(a)\}$ and the logic program $P = \{o(a), o(b), q(x) \leftarrow \mathit{not}\ p(x), o(x)\}$, where p is a classical predicate and o, q are rules predicates. Consider the interpretation $\mathcal{I}_1$ of $\mathcal{L}_\Phi$ such that $\mathcal{I}_1 \models p(a)$ and $\mathcal{I}_1 \models p(b)$. Now, $P^M_{\mathcal{I}_1} = \{o(a), o(b), q(a) \leftarrow \mathit{not}\ p(a), q(b) \leftarrow \mathit{not}\ p(b)\}$, which has one stable s-model, $M_1 = \{o(a), o(b)\}$.

Now consider the interpretation $\mathcal{I}_2$ of $\mathcal{L}_\Phi$ such that $\mathcal{I}_2 \models p(a)$ and $\mathcal{I}_2 \not\models p(b)$. Now, $P^M_{\mathcal{I}_2} = \{o(a), o(b), p(a), q(a) \leftarrow \mathit{not}\ p(a), q(b) \leftarrow \mathit{not}\ p(b)\}$, which has one stable s-model, $M_2 = \{o(a), o(b), q(b)\}$. □

The example shows that P has at least one stable model which does not include $q(b)$ (viz. M_1), whereas one might expect $q(b)$ to be included in every stable model, because $p(b)$ is never *known* to be true.

The following example shows that there might be a discrepancy when there is interaction based on entailment and there is no unique-names assumption in Φ, but it does hold in P.

Example 6. Consider the combined knowledge base $\mathcal{KB} = \langle \Phi, P \rangle$ with $\Phi = \{\forall x, y, z\ (p(x,y) \wedge p(x,z) \supset y = z);\ p(a,b); p(a,c)\}$[5] and $P = \{p'(x,y) \leftarrow p(x,y)\}$, with p a classical predicate and p' a rules predicate. In every model of Φ there is at most one role filler for p (viz. $b = c$), but the single stable e-model of P contains two role fillers for p'. However, one may also argue that this is actually the expected behavior, because the unique-names assumption holds for logic programs. □

Principles 3.1 and 3.2 can be seen as two extremes for the integration of rules and FO theories. One could imagine possibilities which lie between the two extremes. The two formulated principles are by no means the only ways of integrating rules and FO theories, but they neatly generalize current approaches in the literature.

4.4 Interaction from Rules to First-Order Theories

We now consider the interaction from the rules to the FO theory. We assume that the head $H(r)$ of a rule r may contain classical atoms.

Similar to the interaction from FO theories to rules, we distinguish between *interaction based on single models* and *interaction based on entailment*. In the case of interaction based on single models, a model M of $\mathcal{L}_P$ constrains the set of *allowed models* of

[4] This aspect is discussed in more detail in [28].

[5] Note that the first axiom in Φ corresponds to defining p as a functional role in description logics.

Φ; in the case of interaction based on entailment, we join the conclusions about classical predicates which can be drawn from the logic program with the FO theory. This allows to take conclusions from the logic program into account when determining entailments of the FO theory.

Principle 4.1 (Interaction based on single models). *Let $\mathcal{KB} = \langle \Phi, P \rangle$ be a combined knowledge base such that $\Phi \subseteq \mathcal{L}$, $\mathcal{I} = \langle U, \cdot^I \rangle$ an interpretation of $\mathcal{L}_\Phi$, and M an interpretation of $\mathcal{L}_P$, viewed as a pair $\langle V, \cdot^J \rangle$.*

We say that $\mathcal{I}$ respects *M iff, for every classical predicate p, $p^J \subseteq p^I$. Furthermore, $\mathcal{I}$ is an* s-model *of Φ with respect to M iff $\mathcal{I} \models \Phi$ and $\mathcal{I}$ respects M.*

For the principle of interaction based on entailment, we view the model M of a program P as a set of ground atoms that are known to be true; we do not consider the negative part of the model.

Principle 4.2 (Interaction based on entailment). *Let $\mathcal{KB} = \langle \Phi, P \rangle$ be a combined knowledge base such that $\Phi \subseteq \mathcal{L}$.*

Φ e-entails *a formula ϕ* with respect to *a model M of $\mathcal{L}_P$ iff $\Phi \cup M \models \phi$.*

Note that this principle views a model as a set of ground atoms and thus it can only be applied if there is a one-to-one correspondence between names in the language and elements of the domain. Thus, either Principle 1.1 or 1.2 must apply. The combination of the Principles 4.2 and 3.2 yields the following definition of the model of a program:

> An interpretation M is an *e-model* of a rule r with respect to a variable assignment B with associated variable substitution β and a FO theory Φ iff $M, B \models H(r)$ whenever $M, B \models RB(r)$ and Φ e-entails $CB(r)\beta$ with respect to M.

Stable Models for Logic Programs in Combined Knowledge Bases. We now extend the notion of a stable model introduced in the previous section. First, we need to slightly adapt the definition of a reduct of P, as before: Let x be either an s-model $\mathcal{I}$ of Φ with respect to M or Φ. Then, P^M_x is obtained from $gr^{\mathcal{KB}}_y(P)$, where y is either H (in case of Principle 1.1) or $\mathcal{KB}$ (in case of Principle 1.2), by removing

- every rule r such that $x \not\models \exists CB(r)$ if $x = \mathcal{I}$, or such that $x \not\models \exists CB(r)$ with respect to M if $x = \Phi$,
- the classical component from the body of every remaining rule,
- the classical component from the head of every rule r such that $x \not\models \forall CH(r)$ if $x = \mathcal{I}$, or such that $x \not\models \forall CH(r)$ with respect to M if $x = \Phi$,
- every rule r such that $x \models \forall CH(r)$ if $x = \mathcal{I}$, or such that $x \models \forall CH(r)$ with respect to M if $x = \Phi$, in case $CH(r) \neq \emptyset$,
- every rule r such that $B^-(r) \cap M \neq \emptyset$, and
- the negative body literals from the remaining rules.

Then, M is a *stable s-model* (resp., *stable e-model*) of P iff M restricted to the rules predicates is a minimal Herbrand model of $P^M_{\mathcal{I}}$ (resp., P^M_Φ).

The following example demonstrates the difference between the two kinds of interaction:

Table 1. Principles of Current Approaches

	SWRL	**dl-programs**	$\mathcal{DL}$+*log*
Domain of Discourse			
1.1 Herbrand Universe	-	-	+
1.2 Combined Signature	-	+	-
1.3 Arbitrary domains	+	-	-
Uniqueness of Names			
2.1 Names in U_H are unique	-	+	+/-[1]
2.2 Equality predicate	-	-[2]	-[2]
2.3 No uniqueness	+	-	+/-
Interaction from FO Theories to Rules			
3.1 Single models	+	-	+
3.2 Entailment	-	+	-
Interaction from Rules to FO Theories			
4.1 Single models	+	-	+
4.2 Entailment	-	+	-

[1] The combined knowledge base has the standard, and implied unique-names assumption.
[2] Both dl-programs and $\mathcal{DL}$+*log* may be extended with an equality predicate.

Example 7. Consider the combined knowledge base $\mathcal{KB} = \langle \Phi, P \rangle$ with $\Phi = \{p(a) \vee p(b)\}$ and $P = \{q \leftarrow p(a), \mathit{not}\ q; r \leftarrow p(b)\}$, where p is a classical predicate and q is a rules predicate. In case of interaction based on single models, r is included in every stable s-model, since for every model $\mathcal{I}$ in which $p(a)$ is true, there is no corresponding stable s-model for P.

In the case of interaction based on entailment, no such conclusion can be drawn: neither $p(a)$ nor $p(b)$ is e-entailed by Φ. In fact, the only stable e-model of P is the empty set. □

5 Representational Issues in Current Approaches

We can now compare current approaches to integrating description logics and logic programs with respect to the representational issues analyzed above. The three approaches we have selected for the comparison are SWRL [11,20], dl-programs [10], and $\mathcal{DL}$+*log* [9]. These approaches are generalizations of a number of other approaches as discussed in Section 3. The results of the classification are summarized in Table 1. In the remainder of this section, we describe the principles of the mentioned approaches in more detail. We conclude with a few remarks about stable models in these approaches.

5.1 Domain of Discourse

The domain of discourse for SWRL rules is simply the domain of the first-order interpretation of the SWRL FO theory (Principle 1.3). Thus, the variables in the SWRL rules quantify both over the named and the unnamed individuals in the DL component of the knowledge base. SWRL rules do not adhere to the unique-names assumption: several names may refer to the same individual, unless inequality between individuals is explicitly asserted. SWRL does explicitly distinguish between classical predicates and rules predicates. In fact, all predicates in a SWRL knowledge base are classical predicates.

In dl-programs, the domain of discourse corresponds one-to-one with a set of constants in some signature Σ. Typically, and most generally, this signature would be the combined signature $\Sigma_{\mathcal{KB}}$ and thus the variables in the rules may range over names in the combined signature (Principle 1.2).

$\mathcal{DL}$+log has the standard-names assumption for the entire combined knowledge base. Additionally, it is assumed that there is always an infinite number of constant identifiers available in the signature Σ_{Φ} and thus in $\Sigma_{\mathcal{KB}}$. According to the definition of combined knowledge bases in $\mathcal{DL}$+log, the domain of discourse of rules in P is the set of constants in the combined signature (Principle 1.2). However, there is a restriction on the use of variables in $\mathcal{DL}$+log, the *weak DL-safeness*: every variable which occurs in an atom in the head must occur in a positive rules atom in the body. This effectively ensures that each variable which occurs in a rules predicate quantifies only over the names of $\mathcal{L}_P$. Variables which only occur in classical predicates in the body of a rule may quantify over all names in $\Sigma_{\mathcal{KB}}$. Thus, depending on where a variable occurs in a rule, the domain of discourse is either the Herbrand universe U_H^P (Principle 1.1) or the set of names in the combined signature $\Sigma_{\mathcal{KB}}$ (Principle 1.2).

5.2 Uniqueness of Names

SWRL knowledge bases do not assume the unique-names assumption (Principle 2.3), although it can be axiomatized by asserting inequality between every set of distinct constant symbols in $\Sigma_{\mathcal{KB}}$. SWRL allows the use of the equality symbol in P. One could view this as a special equality predicate, although it does not require a special axiomatization, since it is a built into the semantics. All the usual equality axioms are obviously valid in SWRL. One could thus take the point of view that there is an equality predicate in the language and this is a classical predicate and thus SWRL combines the Principles 2.2 and 2.3.

The unique-names assumption holds for the rules in a dl-program (Principle 2.1). Combined with the fact that the domain simply consists of all names of the combined signature, uniqueness of names is assumed even if two names are equal in every model of the FO theory. We illustrated this discrepancy earlier in Example 6. A possible way to overcome this discrepancy is to axiomatize an equality predicate eq in the logic program (Principle 2.2) and to define it in terms of equality statements which are derived from the FO theory:

$$eq(X, Y) \leftarrow DL[=](X, Y).$$

The unique-names assumption holds in any $\mathcal{DL}$+log knowledge base and thus also in the rules component (Principle 2.1). One might allow arbitrary domains for Φ. As pointed out in [28], one may overcome the unique-names assumption by axiomatizing an equality predicate in P, and treating it as a classical predicate (Principle 2.2), similar to the axiomatization for dl-programs proposed above.

5.3 Interaction Between First-Order Theories and Rules

In SWRL, interaction from FO theories to rules, and from rules to FO theories, is based on single models (Principles 3.1, 4.1), since the rules and DL components in SWRL are simply part of one first-order theory. SWRL actually defines one model for both the FO theory and the rules. In terms of combined knowledge bases which we use in this paper, one could equivalently say that all predicates are classical predicates. The models for the FO theory and the rules share the same domain. Finally, an interpretation $\mathcal{I}$ is a model of $\mathcal{KB} = \langle \Phi, P \rangle$ iff $\mathcal{I}$ is an s-model of Φ with respect to every s-model M of P which shares the domain of $\mathcal{I}$.

Interaction between rules and FO theories in dl-program in both directions is based on entailment (Principles 3.2, 4.2). A (ground) dl-atom in the body of a rule in P is true if it is entailed by Φ. The interaction from rules to FO theories diverges somewhat from the description of Principle 3.2. Namely, classical predicates are not allowed to occur in the heads of rules in P. Instead, dl-atoms allow the possibility to select which part of a model M of P should be taken into account when determining truth of the dl-atom.[6] In other words, a ground dl-atom α is true in a model M with respect to FO theory Φ iff $\Phi \cup q(M) \models \alpha$, where $q(M)$ is either (a) a subset of M, (b) the negation of a subset of M, (c) the negation of a subset of the Herbrand base which is not in M, or (d) a composition of any of the above.

In $\mathcal{DL}$+*log*, interaction between FO theories and rules is based on single models (Principles 3.1 and 4.1), as is the case for SWRL. A model $\mathcal{I}$ is an s-model only if there is an s-model M of P which respects $\mathcal{I}$ and $\mathcal{I}$ respects M. The other direction also holds if M is additionally a stable s-model of P with respect to $\mathcal{I}$.

5.4 Stable Models in Current Approaches

SWRL does not have the notion of stable models. This is to be expected since the language does not allow default negation. A formula ϕ is entailed by a SWRL knowledge base $\mathcal{KB}$ if every model of $\mathcal{KB}$ is a model of ϕ.

In dl-programs, a model M is a stable e-model of P with respect to Φ if it is the minimal model of the reduct P_{Φ}^{M} with slightly more complicated conditions for the dl-atoms, since their form needs to be taken into account. Entailment is then defined as follows: P *bravely* entails a ground atom α if α is true in *some* stable model of P and P *skeptically* entails α if α is true in *all* stable models of P.

In $\mathcal{DL}$+*log*, a model M is a stable model of P if it is the minimal model of the reduct $P_{\mathcal{I}}^{M}$. A ground atom α is *entailed* by $\mathcal{KB}$ if (a) it is true in every s-model of Φ, in case α is a classical atom, or (b) it is true in every stable s-model of P, in case α is a rules atom.

6 Settings for Combining Classical Logic and Rules

Based on the analysis of the representational issues in Section 4 and as an abstraction of current approaches to combining rules and FO theories, we define three generic settings for the integration of rules and FO theories. These settings help to classify existing and future approaches to such combinations. Additionally, they help to clarify the space of possible solutions for the integration of FO theories and rules with respect to the way they resolve the representational issues we have pointed out in this paper.

[6] Actually, dl-atoms allow more sophisticated methods of controlling the flow of information. The negation of parts of M can be taken into account and negated information can be taken into account in the absence of information in M.

The three settings we have identified are:

1. In the *minimal interface* setting, the logic program and the FO theory are viewed as separate components and are only connected through a minimal interface which consists of the exchange of entailments. The dl-programs approach [10] falls in this setting.
2. Building an *integrated model*, where the rules and the logic program are integrated to a large extent, although there is a separation in the vocabulary between classical predicate and rules predicates. The integrated model is the union of two models, one for the FO theory and one for the rules, which share the same domain. $\mathcal{DL}$+*log* [9] and SWRL [20] fall in this setting, with the caveat that SWRL does not allow negation in the rules component.
3. A final possible setting is *full integration*, where there is no separation between classical predicates and rules predicates; this makes it possible, among other things, to express nonmonotonic negation over classical predicates. We are not aware of current approaches which fall in this setting, but we can imagine approaches along this line, possibly based on first-order nonmonotonic logics [29,17,30].

The main distinction between the first and second setting is interaction based on single models (Setting 2) versus interaction based on entailment (Setting 1). In the third setting, there is not so much interaction, but rather *full integration*: one can no longer really distinguish between the FO theory and the rules. While Settings 1 and 2 are abstractions of current approaches ([9] and [10], respectively), Setting 3 is not based on current approaches, but we see this setting as a possible development towards a tighter integration of FO theories and (nonmonotonic) logic programs.

Table 2 summarizes the settings and their representational principles.

Table 2. Principles of Settings

	Minimal interface	**Integrated models**	**Full integration**
Domain of Discourse			
1.1 Herbrand Universe	-	-	-
1.2 Combined Signature	+	-	-
1.3 Arbitrary domains	-	+	+
Uniqueness of Names			
2.1 Names in U_H are unique	+	-	-
2.2 Equality predicate	-[1]	-	-
2.3 No uniqueness	-	+	+
Interaction from FO Theories to Rules			
3.1 Single models	-	+	+/-[2]
3.2 Entailment	+	-	+/-[2]
Interaction from Rules to FO Theories			
4.1 Single models	-	+	+
4.2 Entailment	+	-	-
Distinction between classical and rule predicates	+	+	-

[1] An equality predicate can be axiomatized in P

[2] Full integration requires more complex interaction than single models or entailment alone

7 Related Work

Franconi and Tessaris [22] survey three approaches to combining (the DL subset of) classical logic with rules. The three approaches are (i) (subsets of) SWRL, (ii) dl-programs, and (iii) epistemic rules [31]. The latter are a formalization of procedural rules which can be found in practical knowledge-representation systems. Franconi and Tessaris show that all three approaches coincide in case the DL component is empty and the rules component is positive, but that they diverge quickly when adding trivial axioms to the DL component. While Franconi and Tessaris look at the problem of combining classical logic and rules from the point of view of several existing approaches, we surveyed the fundamental issues which may arise when combining classical logic with rules and classified existing approaches accordingly.

Variants of logic-programming semantics without the domain-closure assumption have been studied in the logic-programming literature. In [32], the stable-model semantics is extended to open domains by extending the language with an infinite sequence of new constants. Open logic programs (see, e.g., [33]) distinguish between defined and undefined predicates. The defined predicates are given a completion semantics, similar to Clark's completion [34], and equality is axiomatized in the language. The resulting theory is then given a first-order semantics. Open logic programs were adapted to open answer-set semantics in [35].

It is worthwhile to mention some approaches which propose to use rule-based formalisms (possibly with extended domains) to reason about classical logic, and especially about description-logic theories. [12] proposes to use disjunctive datalog to reason about the description logic $\mathcal{SHIQ}$, extended with DL-safe SWRL rules. [24] uses extended conceptual logic programs to reason with expressive description logics combined with DL-safe rules. [23] proposes a subset of a description logic which can be directly interpreted as a logic program. Open logic programs have been used in [33] to reason with expressive description logics. [24] uses the open answer-set semantics [35] to reason with expressive description logics extended with DL-safe rules. [36] and [37] reduce reasoning in the description logic $\mathcal{ALCQI}$ to query answering in logic programs based on the answer-set semantics.

8 Conclusions and Future Work

There exist several different approaches to the combination of first-order theories (such as description-logic ontologies) and (nonmonotonic) rules (e.g. [8,9,10,11,12]). Each of these approaches overcomes the differences between the first-order and rules paradigms (open vs. closed domain, non-unique vs. unique names, open vs. closed world) in different ways.

We have identified a number of fundamental representational issues which arise in combinations of FO theories and rules. For each of these issues, we have defined a number of formal principles which a combination of rules and ontologies may obey. These principles help to explicate the underlying assumptions of the semantics of such a combination. They show the consequences of the choices which were taken in the design of the combination and help to characterize approaches to combining rules and FO theories according to their expressive power and their underlying assumptions.

We have used the formal principles to characterize several leading approaches to combining rules with (description-logic) ontologies. These approaches are SWRL [20], dl-programs [10], and $\mathcal{DL}+log$ [9]. It turns out that SWRL and $\mathcal{DL}+log$ are quite similar concerning their representational principles, although the approaches might seem quite different on the surface; both approaches specify the interaction between ontologies and rules based on single models, but SWRL does not allow nonmonotonic negation in the rules. The dl-programs approach has quite different underlying assumptions: the interaction between the ontology and logic program is restricted to entailment of ground facts.

Based on the formal principles, the relations between the formal principles, and generalizing existing approaches, we have defined a number of general settings for the integration of rules and ontologies. An approach may define a *minimal interface* between the FO theory and the rule base, the semantics may be based on *integrated models*, or the approach enables *full integration*, eliminating the distinction between classical and rules predicates. These settings mainly differ in the notion of interaction between FO theories and rules. In the minimal interface setting, interaction is based on entailment, whereas in the integrated models setting, the models of the FO theory and the rule base are combined to define an integrated semantics. The full integration setting requires a unified formalism which can capture both classical first-order theories and nonmonotonic logic programs.

Besides the representational principles defined in this paper, an approach to combining rules and ontologies has of course other properties which are of potential interest. To wit, computational properties such as decidability and complexity, which are concerns in several existing approaches (e.g. [21,8,9,10]), are of particular interest. Another issue in such combinations is the ease of implementation and availability of reasoning techniques. For example, the approach in [8] allows to reduce reasoning with combined knowledge bases to standard reasoning services in answer-set programming (ASP) and description-logic engines, whereas the extension to $\mathcal{DL}+log$ [9] requires non-standard reasoning services for description logics (checking containment of conjunctive queries in unions of conjunctive queries). Finally, dl-programs [10] allow a simple extension of existing algorithms for answer-set programming, using standard reasoning services of description-logic reasoners.

Our future work consists of taking the above-mentioned types of principles into account for the classification of approaches to combining FO theories and rules. Furthermore, we will continue to classify upcoming approaches and consider the combination of nonmonotonic ontology languages (e.g. [38,31,39,40]), including ontology languages with transitive closure (e.g. $\mathcal{DLR}_{reg}$ [41]), with rules.

Nonmonotonic logics seem a promising vehicle for an even tighter integration of FO theories and (nonmonotonic) logic programs than dl-programs or $\mathcal{DL}+log$, in the setting of *full integration*. One could think of an extension of a nonmonotonic description logic. For example, [42] contains a proposal for extending the MKNF-DL [39], which is based on the propositional subset of the bimodal nonmonotonic logic MBNF [43], with nonmonotonic rules. Other nonmonotonic logics which one might consider are, for example, default logic [14,29], circumscription [16,17], and autoepistemic logic [15,30].

So far we have considered rules components with the stable-model semantics [7,13]. In future work we may consider the well-founded semantics [6] for arbitrary programs. Additionally, the combination of production rules with ontologies is recently receiving some attention in the context of the W3C Rule Interchange Format (RIF) Working Group[7]. One might consider characterizing combinations of production rules with ontologies, although there are semantic challenges for such a characterization.

References

1. Dean, M., Schreiber, G., eds.: OWL Web Ontology Language Reference. (2004) W3C Recommendation 10 February 2004.
2. Baader, F., Calvanese, D., McGuinness, D.L., Nardi, D., Patel-Schneider, P.F., eds.: The Description Logic Handbook. Cambridge University Press (2003)
3. Horrocks, I., Patel-Schneider, P.F.: Reducing OWL entailment to description logic satisfiability. In: Proc. of the 2003 International Semantic Web Conference (ISWC 2003), Sanibel Island, Florida (2003)
4. Genesereth, M.R., Fikes, R.E.: Knowledge interchange format, version 3.0 reference manual. Technical Report Logic-92-1, Computer Science Department, Stanford University (1992)
5. Delugach, H., ed.: ISO Common Logic. (2006) Available at `http://philebus.tamu.edu/cl/` .
6. Gelder, A.V., Ross, K., Schlipf, J.S.: The well-founded semantics for general logic programs. Journal of the ACM **38**(3) (1991) 620–650
7. Gelfond, M., Lifschitz, V.: The stable model semantics for logic programming. In Kowalski, R.A., Bowen, K., eds.: Proceedings of the Fifth International Conference on Logic Programming, Cambridge, Massachusetts, The MIT Press (1988) 1070–1080
8. Rosati, R.: On the decidability and complexity of integrating ontologies and rules. Journal of Web Semantics **3**(1) (2005) 61–73
9. Rosati, R.: $\mathcal{DL}$+log: Tight integration of description logics and disjunctive datalog. In: KR2006. (2006)
10. Eiter, T., Lukasiewicz, T., Schindlauer, R., Tompits, H.: Combining answer set programming with description logics for the semantic web. In: Proc. of the International Conference of Knowledge Representation and Reasoning (KR'04). (2004)
11. Horrocks, I., Patel-Schneider, P.F., Boley, H., Tabet, S., Grosof, B., Dean, M.: SWRL: A semantic web rule language combining OWL and RuleML. Member submission 21 May 2004, W3C (2004)
12. Motik, B., Sattler, U., Studer, R.: Query answering for OWL-DL with rules. In: Proceedings of 3rd International Semantic Web Conference (ISWC2004), Hiroshima, Japan (2004)
13. Gelfond, M., Lifschitz, V.: Classical negation in logic programs and disjunctive databases. New Generation Computing **9**(3/4) (1991) 365–386
14. Reiter, R.: A logic for default reasoning. In Ginsberg, M.L., ed.: Readings in nonmonotonic reasoning. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA (1987) 68–93
15. Moore, R.C.: Semantical considerations on nonmonotonic logic. Artificial Intelligence **25**(1) (1985) 75–94
16. McCarthy, J.: Applications of circumscription to formalizing common sense knowledge. Artificial Intelligence **28** (1986) 89–116
17. Lifschitz, V.: Circumscription. In: Handbook of Logic in AI and Logic Programming, Vol. 3, Oxford University Press (1994) 298–352

[7] `http://www.w3.org/2005/rules/wg`

18. Lloyd, J.W.: Foundations of Logic Programming (2nd edition). Springer-Verlag (1987)
19. Przymusinski, T.C.: On the declarative and procedural semantics of logic programs. Journal of Automated Reasoning **5**(2) (1989) 167–205
20. Horrocks, I., Patel-Schneider, P.F.: A proposal for an OWL rules language. In: Proc. of the Thirteenth International World Wide Web Conference (WWW 2004), ACM (2004) 723–731
21. Levy, A.Y., Rousset, M.C.: Combining Horn rules and description logics in CARIN. Artificial Intelligence **104** (1998) 165 – 209
22. Franconi, E., Tessaris, S.: Rules and queries with ontologies: a unified logical framework. In: Workshop on Principles and Practice of Semantic Web Reasoning (PPSWR'04), St. Malo, France (2004)
23. Grosof, B.N., Horrocks, I., Volz, R., Decker, S.: Description logic programs: Combining logic programs with description logic. In: Proc. Intl. Conf. on the World Wide Web (WWW-2003), Budapest, Hungary (2003)
24. Heymans, S., Nieuwenborgh, D.V., Vermeir, D.: Nonmonotonic ontological and rule-based reasoning with extended conceptual logic programs. In: ESWC 2005. (2005) 392–407
25. Donini, F.M., Lenzerini, M., Nardi, D., Schaerf, A.: AL-log: integrating datalog and description logics. Journal of Intelligent Information Systems **10** (1998) 227–252
26. Rosati, R.: Towards expressive KR systems integrating datalog and description logics: A preliminary report. In: Proc. of the 1999 International Description Logics workshop (DL99). (1999) 160–164
27. Reiter, R.: Equality and domain closure in first-order databases. Journal of the ACM **27**(2) (1980) 235–249
28. Rosati, R.: Semantic and computational advantages of the safe integration of ontologies and rules. In: Proceedings of PPSWR2005, Springer-Verlag (2005) 50–64
29. Lifschitz, V.: On open defaults. In Lloyd, J., ed.: Proceedings of the symposium on computational logic, Berlin: Springer-Verlag (1990) 80–95
30. Konolige, K.: Quantification in autoepistemic logic. Fundamenta Informaticae **15**(3–4) (1991) 275–300
31. Donini, F.M., Lenzerini, M., Nardi, D., Nutt, W., Schaerf, A.: An epistemic operator for description logics. Artificial Intelligence **100**(1–2) (1998) 225–274
32. Gelfond, M., Przymusinska, H.: Reasoning on open domains. In: LPNMR 1993. (1993) 397–413
33. Van Belleghem, K., Denecker, M., De Schreye, D.: A strong correspondence between description logics and open logic programming. In: Logic Programming, Proceedings of the Fourteenth International Conference on Logic Programming, MIT Press (1997) 346–360
34. Clark, K.L.: Negation as failure. In Gallaire, H., Minker, J., eds.: Logic and Data Bases. Plenum Press, New York, USA (1978) 293–322
35. Heymans, S., Van Nieuwenborgh, D., Vermeir, D.: Guarded Open Answer Set Programming. In: 8th International Conference on Logic Programming and Non Monotonic Reasoning (LPNMR 2005). Number 3662 in LNAI, Springer (2005) 92–104
36. Baral, C.: Knowledge Representation, Reasoning and Declarative Problem Solving. Cambridge University Press (2003)
37. Swift, T.: Deduction in ontologies via ASP. In: LPNMR2004. (2004) 275–288
38. Bonatti, P., Lutz, C., Wolter, F.: Expressive non-monotonic description logics based on circumscription. In: KR2006. (2006)
39. Donini, F.M., Nardi, D., Rosati, R.: Description logics of minimal knowledge and negation as failure. ACM Transactions on Computational Logic **3**(2) (2002) 177–225
40. Baader, F., Hollunder, B.: Embedding defaults into terminological knowledge representation formalisms. Journal of Automated Reasoning **14** (1995) 149–180

41. Calvanese, D., Giancomo, G.D., Lenzerini, M.: On the decidability of query containment under constraints. In: Proc. of the 17th ACM SIGACT SIGMOD SIGART Symp. on Principles of Database Systems (PODS'98). (1998) 149–158
42. Motik, B., Rosati, R.: Closing semantic web ontologies. Technical report, University of Manchester (2006) Available at `http://www.cs.man.ac.uk/ bmotik/publications/paper.pdf`.
43. Lifschitz, V.: Minimal belief and negation as failure. Artificial Intelligence **70**(1-2) (1994) 53–72